DE GRUYTER OPEN

STUDIES IN LOGIC, GRAMMAR AND RHETORIC 36 (49) 2014
DOI: 10.2478/slgr-2014-0005

Peter Simons
Trinity College Dublin

LINGUISTIC COMPLEXITY AND ARGUMENTATIVE UNITY: A LVOV-WARSAW SCHOOL SUPPLEMENT

Abstract. It is argued that the source of complexity in language is twofold: repetition, and syntactic embedding. The former enables us to return again and again to the same subject across many sentences, and to maintain the coherence of an argument. The latter is governed by two forms of complexification: the functor-argument structure of all languages and the operator-bound-variable mechanism of familiar formal languages. The former is most transparently represented by categorial grammar, and an extension of this can adequately describe the syntax of variable binders. Both developments have roots within the work of the Lvov-Warsaw School.

Keywords: categorial grammar, complexity, cohesion, definition, functor, operator, proof

This whole volume is one long argument
Charles Darwin, *The Origin of Species*

Revenons à ces moutons!
Anonymous, *La Farce de Maître Pathelin*

1. Introduction: Linguistic Complexity and Argumentation

Humans are the only animals to possess language. Other animals have communication systems but none can match the complexity of human language. It is the principal claim of this paper that the engine of this complexity is syntax, and that it is most transparently captured by categorial grammar, which of course was first formulated within the Lvov-Warsaw School of logic and philosophy. However a single sentence or a chain of semantically connected sentences needs devices for keeping track of subject matters as complexity increases. In formal languages the indispensable device to this end is the repetition of constants, symbols with fixed meaning used in different equiform tokens of one type, and of variables, as exemplified in the bound variables of expressions in logic and mathematics. The approach of categorial grammar can be extended to apply to such bound variables, and the paper briefly indicates how.

In natural languages, the devices for maintaining coherence within and across sentences are more varied, and include not just repetition but also pronouns, paraphrases, synonyms and the like. It is through such connecting devices that the processes of conversation, dialogue, discussion and argumentation may remain connected in such a way as to count as a single discourse. When in the course of an argument or other discourse we return to a previous subject, or are asked to "keep to the point", as in the medieval French comedy the judge enjoins the confused plaintiff to get back to talking about his stolen sheep, we have to be able to pick up from parts of the foregoing discourse, no matter how deeply embedded. That we are able to do so is determined by the limited and surveyable number of devices for building complexity and yet maintaining connection or cohesion of discourses. In logical argumentation, in proofs, a sequence of formulas "keep to the point" by similar forms of repetition, and also by linking diverse expressions through formal definitions. These facts, usually taken for granted, are taken seriously in the metalogical investigations of Stanisław Leśniewski, and exploited in the formalization of proofs from assumptions by Stanisław Jaśkowski, which are mentioned in the final section. So the Lvov-Warsaw tradition not only lies deep in the history of twentieth century logic, but also at the origins of modern argumentation theory.

2. Ontological versus Linguistic Complexity

Consider the following entry in Samuel Pepys' diary for Tuesday 25 September 1660:

> To the office, where Sir W. Batten, Collonell Slingsby, and I sat a while; and Sir R. Ford coming to us about some business, we talked together of the interest of this kingdom to have a peace with Spain and a war with France and Holland – where Sir R. Ford talked like a man of great reason and experience. And afterwards I did send for a cup of tee (a China drink) of which I never had drank before) and went away.
>
> Then came Collonell Birch and Sir R. Browne (by a former appointment) and with them from Towre-wharf in the barge belonging to our office we went to Deptford to pay off the ship *Successe*. Which (Sir G. Carteret and Sir W. Penn afterwards coming to us) we did, Collonell Birch being a mighty busy man and

> one that is the most indefatigable and forward to make himself work of any man that ever I knew in my life. At the globe we had a very good dinner, and after that to the pay again; which being finished we returned by water again. And I from our office with Collonell Slingsby by Coach to Westminster (I setting him down at his lodgings by the way) to enquire for my Lord's coming thither (the King and the Princesse coming up the River this afternoon as we were at our pay); and I found him gone to Mr. Crews, where I find him well; only, had got some brush upon his foot which was not well yet. My Lord told me how the ship that brought the Princesse and him (The *Tredagh*) did knock six times upon the Kentish Knock, which put them in great fear for the ship; but got off well. He told me also how the King had knighted Vice-admirall Lawson and Sir Rich. Stayner. From him late, and by Coach home – where the playsterers being at work in all the rooms in my house, my wife was fain to make a bed upon the ground for her and I; and so there we lay all night.[1]

Saving the fact that only a few knowledgeable historians would already know which persons are being referred to in this passage, and what the Kentish Knock is (a shallow sandbank in the North Sea east of the Thames estuary, dangerous to shipping), the general line of the story is easy enough to follow. Apart from Pepys himself, who as diarist is obviously involved in most of what is described, several of the people mentioned appear more than once in the narrative: Colonel John Birch, Colonel Robert Slingsby, King Charles II, the Princess Royal. Over the course of Pepys's whole diary, many people, places and events are mentioned many times. The narrative gives us a schematic but informative picture of part of what was going on to various participants in London on 25 September 1660, and the whole diary of course over a much longer period. Following a term much used by Whitehead, we might say the story is about the "adventures" of Pepys and his various friends and acquaintances, as well as the places they inhabit.

If we consider the story, it involves complexity: events follow one another and are linked, several people are involved in a single event, the same person is involved successively in many events. The people and places mentioned are themselves complex. The Palace of Whitehall, residence of the king, was a large and rambling group of buildings with many rooms. Pepys himself had all the complexity that any adult human being has, his bones, heart, liver, kidneys, brain and so on all functioning in their accustomed way across many years and encompassing untold millions of microevents, most of them too routine and unimportant to rate a mention, but existent nevertheless. Pepys moved around from place to place: from his home to his office, to other offices, to the theatre, to inspect His Majesty's ships, and so on. Every movement involved an untold number of displacements of objects and their parts, each describable by a mathematical function of considerable complexity. Of all this ontological complexity, only the minutest schematic portion is captured in the bald narrative of the diary, and the total detail outruns the linguistic and informational capacity of anyone to represent. Ontological complexity outruns linguistic complexity by many orders of magnitude, indeed possibly by at least one or two alephs, if motion is continuous, while on the other hand any portion of language is finite.

Despite this, we are able in language to capture part of this complexity in language. So the question I am raising is: what is it about language that enables us to correctly (truly) describe parts and aspects of the world? We take language for granted in everyday life, but it is on reflection amazing that a sequence of articulated sounds, or in the written case, a sequence of approximate geometric shapes, is capable of expressing such complication. To describe how this works is of course the job of linguistics, and I do not for a moment intend to enter into detail, but we may wonder where, in the phenomenon that is language, the potentiality for linguistic complexity arises. Of course without intelligent persons and their language-processing brains, none of it would be possible, but taking the instrument for granted, there still has to be a natural account of where the complexity arises.

Consider by way of analogy a piano, as played by a single pianist. Those strings, frame, 88 keys and the mechanism to which they attach, together with a capable performer of course, are the *conditio sine qua non* for performances of Beethoven's Opus 111 or Prokofiev's Sonata No. 8, but what gives rise to, what is the source of, the *musical* complexity? Part of it is that at any one time up to ten keys are depressable by a single pianist, but even a monotone instrument like an oboe can play complex pieces, so the chief source of the complexity must be succession in time, allowing notes to be of different lengths and to succeed one another. Succession allows a multiplicity of varied elements to be stacked up one after another, and in this regard language and music are very similar. But mere succession of varied elements does not in itself give rise to the kind of variety we find in the Pepys passage, allowing such successive complexes to represent very different kinds of thing and event and their manifold interrelations. Musical notes do not divide into different categories in the way that words do: there is nothing corresponding to the distinctions between nouns, verbs, adjectives, adverbs, prepositions, conjunctions and sentences, for example.

So while temporal succession or some analogous linear spatial succession (as in writing) is what serves as the locational container or space for linguistic complexity, there is more to it than that. The differences among the kind of words have been the subject of descriptive grammars for thousands of years, but the key general principles behind the power and variety of linguistic complexity have been understood much more recently, and the key breakthrough came in the work of the Lvov-Warsaw School, specifically in the work of Stanisław Leśniewski and Kazimierz Ajdukiewicz. That approach has come to be known as *categorial grammar*, and with minor reservations I consider it to embody the best general account of how linguistic complexity works. The bulk of this paper is an attempt to show how and why categorial grammar transparently represents linguistic complexity. In the final section I connect the complexity of language to the complexity of discourse more generally, and indicate briefly how the Lvov-Warsaw School opened up avenues in the ongoing discussion of argumentation.

3. Application

Categorial grammar had two founding fathers: one by practice, the other by precept. Gottlob Frege analysed linguistic utterances and their contents not via the traditional opposition between subject and predicate, which is at best relevant to a small (if important) class of sentences, but in terms of the mathematical idea of a function, in application to an argument or several arguments. In the true arithmetical sentence[2]

$$7 + 4 < 3 \cdot 4$$

there is no single subject and predicate, but there is a verb, '$<$', which applies to two complex terms, '$7+4$' and '$3\cdot4$', and these in turn are composed of two binary operators, the addition operator '$+$' and the multiplication operator '$\cdot$', each taking two unstructured numerals as arguments. Each of the complex terms names a number, on the left-hand side 11, on the right-hand side 12.

Frege's insight is that whereas simple or complex names such as '7' or '$7 + 4$' do their job of naming (denoting) something as they stand, the other signs, that is in our example '$+$', '$\cdot$', and '$<$', do not do work in isolation, but need to be supplemented by other signs in order to work. Frege says that such signs stand in need of supplementation, and indeed to emphasise this fact he always quotes them in a context with names, or expressions serving as dummy names, marking the places where supplementation or completion occurs. When suitably supplemented, the resulting wholes, involving the supplemented signs together with what supplements them, do a suitable semantic job, in the case of '$7 + 4$' and '$3 \cdot 4$' denoting numbers, and in the case of the whole sentence expressing a truth.

With few exceptions, language in use consists of producings and receivings of sentences. These make statements, ask questions, issue commands, and the like. How the sentences get to hook up semantically to the world is through two kinds of mechanism: indexical expressions (or indeed tacit aspects of the utterance) which exploit the context of utterance (its time, location, speaker, addressees, salient features of the environment etc.), and names, which denote given particular individuals or groups. The two mechanisms often function in tandem together in a sentence, even within a single phrase, as in

The woman over there talking to Dermot used to be my wife

where the first noun phrase 'the woman over there talking to Dermot' uses the present continuous tense, a contextually resolved reference to a woman perceptible to speaker and hearer, and the proper name 'Dermot', while the second noun phrase relates the woman to the speaker via the possessive pronoun, and the use of the compound verb 'used to be' indicates that the woman in question was the speaker's wife but at the time of utterance is so no longer.

Clearly the two key elements in this are the whole sentence and the parts that get the sentence to denote items in the world. Leaving non-declarative sentences aside, the point of sentences in standard speech is to put something forward that is intended to be taken as true. When things go well, what is said is true, when they don't, then not.

So the fundamental idea that made Frege appeal to the mathematical notion of a function rather than the logico-linguistic notion of a predicate is that, given one or more names, to get a sentence you need something else which, when suitably combined with the names, results in a sentence. The names can be varied and this remaining part be kept constant, just as the numbers input to the addition function $x + y$ can be varied and different sums result. We speak in such cases of the *application* of a function to its one or more arguments, resulting in a value. This is the key idea in categorial grammar, but with an important difference. Frege took there to be a fundamental isomorphism between language, what it means, and what it is about, or between sign, sense, and signification (reference). This led to his use of the metaphor of *unsaturatedness*, which he took to apply in the first instance to functions, then to sense, and finally to signs themselves. Without going into why in detail, we wish to dissociate the idea of application from any presumption of isomorphism, retaining only the idea of one expression's being the result of the *application* of one or more expressions (or expression patterns) to other expressions (or expression patterns), to yield a complex expression. To avoid this, and indeed to cohere with standard terminology, we avoid the term 'function' and instead employ Carnap's word 'functor'[3] and say that a complex expression typically (not always – see below) consists of a functor and one or more argument expressions. The complex expression results from the application of the functor to the arguments, or conversely from the saturation of the functor by the arguments.

4. Iterated and Recursive Application

The principle of linguistic complexification was again perhaps first fully understood by Frege, who pointed out two ways in which complexity may ramify. The first is that an argument to a functor may itself be complex, the result of saturation or application at a lower level. While there are practical limits to the depth to which linguistic structures can be nested and ramified, limits imposed by the contingent limitations of time, space and the ability of speakers to keep things in mind, there are no theoretical limits. A sentence in English can go on for as long as the author can manage or feel like, and there are indeed some long sentences, including the last of Joyce's *Ulysses*, which has over 4,000 words. Application can take place over and over, as when a string of what would otherwise be complete sentences are strung together coordinatively by repeated 'and's. More subtly, such repetitions are frequently cases of recursion, where a single form of application pertains to one or more of the arguments, as in nested relative clauses in the English nursery rhyme *The house that Jack built*, which in the final stanza are embedded to a depth of ten levels. Frege exploited just such nesting in his logic, building up elaborate formal sentences using repeated use of the conditional 'if ... then –––' in his graphical notation. For example in the proof of Theorem 655 of his *Grundgesetze der Arithmetik* there are formulas with sixteen clauses, so fifteen conditionals.[4] An adequate account of linguistic complexity must allow for this.

A different kind of ramification occurs when what works as a functor in some contexts is an argument in others. In the sentence

Sean plays golf

it is clear that the functor is the verb phrase 'plays golf'. We negate the whole sentence by negating this phrase:

Sean does not play golf

On the other hand the superficially similar sentence

Every Irishman plays golf

is negated not by negating the verb but by negating the initial quantifier phrase, as

Not every Irishman plays golf

which shows that in this sentence the verb phrase is an argument to the higher order functor 'every Irishman'.

Again Frege was the first to grasp this fact in all its theoretical ramifications. In principle any functor can serve as argument to a functor of order higher than its own. In practice we do not tend to climb very high in this hierarchy, but the option is there. For example in the sentence

It is easier to amuse every Irishman than every Scotsman

the quantifier phrase 'every Irishman' functions as an argument.

Following now standard terminology we call the functor which at the first level of analysis binds a complex expression together the complex expression's *main functor*. So whether a functor is a main functor or not depends on the context in which it occurs.

5. Syntactic Categories

While Frege worked very efficiently with expressions of different functorial levels, he did not pause to codify or notate the principles involved. The general idea of laws governing the combination of meanings was outlined in principle by Husserl in the fourth of his *Logical Investigations*, where meanings are divided into dependent and independent, the former corresponding to functor expressions, the latter to non-functor expressions. Husserl proposed that meanings fall into different categories (*Bedeutungskategorien*) according to the way in which they legally can combine.

Husserl's idea of categories of meaning and Frege's of categories of function were merged by one of the few logicians to be influenced by them both, Stanisław Leśniewski. In his logical languages Leśniewski assigned all expressions other than parentheses and quantifiers to what he called *semantic* categories. But again he worked with the idea rather than codifying and notating it. That achievement is due to his contemporary Kazimierz Ajdukiewicz, in his 1935 article 'Die syntaktische Konnexität',[5] in which a system of notation and rules of combination are explicitly formulated for the first time. After a slow start, this idea spread widely in the logical and linguistic community and has come to be called *categorial grammar*. We shall give briefly notation and principles of categorial grammar, which captures in the most transparent form the way in which linguistic complexity arises and ramifies.

In categorial grammar, as in the tradition leading up to it, expressions are divided into different syntactic categories. Some are non-functorial or basic. Which basic categories there are depends on the language. In propositional calculus the only basic category is that of SENTENCE (S). In Frege's logic, leaving the judgement stroke aside, there is again only one basic category, that of NAME (N). Leśniewski in his logical languages of ontology and mereology had both S and N as basic categories, and it is the same in predicate logic or in the simple theory of types. Natural languages may have further categories, as surmised by Ajdukiewicz, such as COMMON NOUN (C) and perhaps TENSE (T) and others; the details might vary from language to language while the principles remain the same. They are encapsulated in the following rules:

R1 There is a finite collection of basic categories $\kappa_1, \ldots, \kappa_m$, $m \geq 1$, all the κ_i different.

R2 Assuming all categories are ordered alphabetically (the details of how this is done may be left aside), if α is a category and $\beta_1, \ldots, \beta_n$ are $n \geq 1$ categories (not necessarily all different) in alphabetical order, then there is a functor category of expressions taking arguments of categories $\beta_1, \ldots, \beta_n$ and yielding a complex expression of category α. The category of the functor is written $\alpha\langle\beta_1 \ldots \beta_n\rangle$.

The simple recursivity of R2 ensures the ramification of potential complexity among expressions. Note that it is wholly syntactic in nature.

6. Cumulativity and Complexity

Complication in language has two clearly distinct sources. One is the simple cumulative possibility of saying more. This need not involve syntactic complexity beyond that required to form individual sentences. In the sense that complexity requires multiplicity *in unity* it is not complexity at all. A simple sentence repeated over and over (imagine some of piece of absurdist theatre or performance art) no more engenders linguistic complexity than a steadily dripping tap engenders a complex fluid flow. Call it instead *cumulativity*. The phenomenon of functorial nesting however is different and is clearly the source of any complexity beyond that of the simplest utterances. There is one linguistic phenomenon that shares features of both complexity and cumulativity and that is coordination. In sentences like Shakespeare's

> The master, the swabber, the boatswain, and I,
>
> The gunner, and his mate,
>
> Lov'd Mall, Meg, and Marian, and Margery,
>
> But none of us car'd for Kate

the listed names, with occasional 'and's in each list, are clearly cumulative, albeit constrained within the confines of a single sentential place. Likewise the ability to string individual sentences together with 'and', like daisies in a chain, is more akin to cumulation than genuine complexity. It can be subsumed under the latter, if we allow that the conjunction 'and' is of category S⟨SS⟩ and treat a cumulated conjunction as comprising the first sentence and the added sentence as the second. But while formally correct, the analysis only incidentally captures the repetitiveness which is characteristic of cumulation. While we can let coordination count as a form of complexity, it is then a limiting case. A more revealing analysis would be to say that 'and' is a multicategorial word of categories S⟨S…S⟩, for any finite number of S arguments greater than one. Here the cumulativity comes out in the analysis.

7. Combination

For linguistic complexity to occur, clearly expressions have to be stacked up together in a way which gives rise to unified linguistic complexes. Superficially this occurs by concatenation, which is itself a form of phonemic or graphemic cumulativity. What takes it beyond the mere stringing together of bits of language into a chain is the categorial diversity of the parts strung together, which enforce a structural hierarchy of parts according to the ideas familiar from grammar, but most cogently captured in categorial grammar. Since we are not concerned with the details of how combination takes place and how different arguments are fitted together with their functors, which details vary from language to language, we may be schematic in our representation of combination. Suppose A is an expression of functor category $\alpha\langle\beta_1 \ldots \beta_n\rangle$ and $B_1 \ldots B_n$ are expressions of categories $\beta_1 \ldots \beta_n$ respectively, with the β_i in alphabetical order as before, then we denote by $A(B_1 \dots B_n)$ the complex expression formed by saturating A by the B_i in the grammatically correct way. Its category is α. Note we are placing the functor before its arguments, as is standard in predicate logic. Strictly speaking the parentheses are superfluous providing the expressions are all well formed and of the correct respective categories, but we leave them in to help make structure explicit. Plausible grammatical analyses of the sentences

Sean plays golf

and

Every Irishman plays golf

have the forms

$$\text{plays}_{\ S\langle N\rangle\langle N\rangle}\ (\text{golf}_N)(\text{Sean}_N)$$

and

$$\text{every}_{\ S\langle S\langle N\rangle\rangle\langle C\rangle}\ (\text{Irishman}_C)(\text{plays}_{\ S\langle N\rangle\langle N\rangle}(\text{golf}_N))$$

respectively. If it is desired to make explicit the categories of resultant categories in the analysis (at intermediate and final level) we can easily do this, either by labeled tree structures, or more compactly if less perspicuously by labeled bracketing, as

$$[[\text{plays}_{\ S\langle N\rangle\langle N\rangle}\ (\text{golf}_N)]_{S\langle N\rangle}(\text{Sean}_N)]_S$$

and

$$[[\text{every}_{\ S\langle S\langle N\rangle\rangle\langle C\rangle}(\text{Irishman})_C]_{\ S\langle S\langle N\rangle\rangle}([\text{plays}_{\ S\langle N\rangle\langle N\rangle}(\text{golf}_N)]_{S\langle N\rangle})]_S$$

The latter type of notation obviously threatens to become rapidly unsurveyable, so for clarity labeled trees are better, though we omit them here out of space considerations. In the above two examples it is notable that the phrase ‘plays golf’ occurs in each case as a unit, an intransitive verb phrase, which corresponds to our linguistic feeling that it indeed is a unitary phrase in each case.

There are many ways in which categorial grammar can be carried well beyond what is merely indicated here: into matters of resolution of complex expressions and criteria of grammaticality, especially when considering partial saturation; into considerations of linear structure, its limits and the nature of remote connection; and even into aspects of morphology, such as the difference between the untensed ‘play’, as in

Can Sean play golf?

and the tensed 'plays' or 'played'. For present purposes however we may leave things here, and simply reiterate our point that categorial grammar offers the most perspicuous and elegant representation of the nature of linguistic complexity.

8. Binding

The relatively simple functorial structures outlined above work remarkably well for natural languages, but it is a stark fact that they break down for nearly all logical and mathematical languages, for reasons to be explained. There are also features of natural language such as pronoun binding which the mere categories of expression and their functorial combination fail to capture or explain. For example in the sentences

Dale insulted Stevie and she left him

and

Dale insulted Stevie and he left her

there are in each case two possible readings, depending on the gender of the persons named 'Dale' and 'Stevie'. This goes beyond grammar. A very straightfoward illustration in logic comes from the difference between two predicate logical sentences

$\forall x \exists y Rxy$

and

$\forall z \exists y Rxy.$

The first sentence is unexceptionable, whereas the second is not well-formed according to some formation rules, but even if well-formed, differs from the first in being not a closed formula or sentence but an open formula. In other words, whereas under an interpretation the first sentence will have a truth-value, the second will not, since it contains a free or unbound variable 'x'. Yet from the point of view of grammar there is no difference between them if both are well-formed, and no grammatical explanation why the second might not be well-formed in some languages. It was recognized from the very beginning by Ajdukiewicz that variable-binding operators like the quantifiers are not adequately dealt with as functors in the sort of categorial grammar we have considered hitherto. Ajdukiewicz and others have attempted expedients to get around this, but there is a reason why no such attempt can succeed.

Before we see what this is, let us recall the now familiar fact that a language using variable binding, such as those of logic and mathematics, does not need more than one binding operator, or more precisely, one typed family of typically analogous operators. Russell already had an inkling of this, but it was first made fully explicitly by Alonzo Church.[6] Church's λ operator can be used as the sole binder, all other operators being definable as the combination of a functor with the λ operator. If λ can be replaced or eliminated, then the problem of representing variable binding would not arise. And so it does not, since it is well known that λ can be replaced by a suite of combinators, which are functors, not operators. For a purely combinatory language, functors alone suffice and the categorial grammar we have outlined is adequate. There are some details about the differences between typed and untyped languages but the general principle is the same.

However this does not get us off the hook of explaining how languages with binding work. To suppose it did would be like supposing that we can give a grammar for English by using the grammar of French and showing that any English sentence has a French translation. It is simply avoiding the issue. So we need to show how languages with binding can have their own grammar. Here the principal difference between functor (only) languages and operator languages comes into play. For a functor simply adds one layer of grammatical complexity to its arguments, and is grammatically indifferent to any embedded complexity of its arguments. That in a sense is its virtue and simplicity. A variable binder however not only adds a level of complexity, but it can bind variables nested at any finite depth of structure whatsoever within its scope. This can be seen even in simple logical examples such as

$$\forall x \ulcorner Fx \to \exists y \ulcorner Rxy \lor \forall z \ulcorner (Fy \land Rzx) \to Rzy \urcorner\urcorner\urcorner$$

where the bound variable 'x' occurs at the top, intermediate and bottom levels of the matrix wherein it is bound, and also more dramatically elsewhere in logic and mathematics. It is this arbitrary depth of binding that leads to the long and unsurveyable swathes of combinators that typically grace combinatory equivalents to λ formulas, since it takes many applications of structure-shifting and repetition-reducing combinators to get variables out to where they can be "explained away".

For this reason, an adequate grammar for variable binders must go beyond functorial categorial grammar in two linked ways. Firstly, it must represent not only the category but also the internal structure of any matrix, i.e. expression into which is bound. And secondly it must mark the places within this structure into which a given binding variable binds. There is a subtle but relatively unknown reason why it is not necessarily a variable of the right shape that is to be found filling such places. It is possible to have a logical language – that of Frege's *Begriffsschrift* is the most salient example, indeed the only one I know of – in which places are *marked* by a bound variable but filled by some other expression.[7] So for example Frege expresses that 12 is a multiple of 4 by a formula[8] which we can notate, expanding a notation of Leśniewski and using lower corners to group a sequence of binding variables and upper corners as above to mark operator scope, as

$$\text{Anc}\llcorner\gamma\beta\lrcorner\ulcorner 0_\gamma + 4 = 12_\beta\urcorner$$

meaning that 12 is got from 0 by adding 4 some finite number of times. The Anc operator is the proper ancestral, in this case of the relation of being 4 greater than, and its variables mark but do not fill places in its matrix. Taking a leaf out of Frege's book, we can use such marking variables to show which places a variable operator reaches into within its scope. This is why the full syntactic structure of the matrix is required, and not just its resultant category. I have outlined elsewhere[9] the principles of and given a notation for such an extended categorial grammar, which involves adding a third rule R3 to our grammar,

R3 If A is expression of category α containing occurrences of expressions $e_1, \ldots, e_k$ where the e_i are here listed (but do not necessarily occur in the expression A) in alphabetical order, and $\gamma_1, \ldots, \gamma_k$ are the categories of $e_1, \ldots, e_k$ respectively, and Z is an expression of category $\beta\llcorner\gamma_1 \ldots \gamma_k\lrcorner\ulcorner\alpha\urcorner$ and $v_1, \ldots, v_k$ are variables of categories $\gamma_1, \ldots, \gamma_k$ respectively (which may but need not be identical with their respective e_i), then the expression $Z\llcorner v_1 \ldots v_k\lrcorner\ulcorner A^*\urcorner$ is of category β, where A^* is either the same as A if the v_i are the e_i, or is derived from A by marking places where (tokens of) the e_i occur with the variables v_i (as in Frege's notation). Here the variables v_i accompanying Z in lower corners bind either variables or places within A, and in the latter case their loci of binding are shown in A^*.

The only possibility that this rule leaves out is where the matrix, here a single expression, is replaced by a sequence of expressions of possibly different categories, like a many-placed functor. It is safe to say such a theoretical possibility has been rarely if ever used even in advanced logico-mathematical languages. It might have a use, but we can leave that for another time.

It is the combination of functorial complexification and variable binding deep into the resulting complexes that give mathematical languages their spectacular expressive power. As we mentioned, the variable-binding option is in principle dispensable, and natural languages exploit it hardly if at all, but the combinatorial alternative is so unwieldy in complex cases that variable binding is nearly always practically preferable for human use. For computers lacking the frailty of human memory it may well be different. However without functors to build up the complex structures into which variables can be inserted so as to be bound, there is nowehere for variables to bind into. So it remains true that the primary, if not always the sole, source of linguistic complexity of the genuinely structural kind is functorial application.

9. Repetition

A story can link together events and episodes regarding particular participants by the device of repetition. In the case of Pepys's diary with which we began, the first-person nature of the narrative keeps the diarist constantly in play, but as we mentioned, other players have their parts and they too may return again and again. In this way the complex tapestry of events and their participants is woven. Apart from pronouns, which work only over relatively short distances, the main work of keeping the same individuals under discussion in the narrative is carried by repetitions of identifying names and other phrases: 'Sir R. Ford', 'Spain', 'my Lord', 'the King', 'my wife' and so on. It is no accident that bound variables, while in context they function semantically and syntactically like pronouns, are in superficial form like repeated names, as indeed the frequent grouping together of bound and free variables indicates. For it is the repetition of a distinctive identifying form like a name but not itself a name that enables bound variables to function across arbitrary syntactical distances, across arbitrarily long and cumulatively told stories, and to plumb arbitrary syntactic depths. So repetition, while not the source of *linguistic* complexity, is the carrier for language's ability to elaborate accounts of arbitrary *semantic* complexity about a subject. To take one fairly extreme example, one may consider the eight volumes and (to date) ten companion volumes of the biography of Winston Churchill by Randolph Churchill and Martin Gilbert. Theoretically, that could be grammatically compressed into a single sentence, no doubt to its considerable literary detriment. The names 'Winston' and 'Churchill' occur therein many thousands of times, and while not the only source of return to the same topic, are its principal carrier. If we are repeatedly to "revenir à nos moutons” we need names for the sheep. And to employ an insight of Wittgenstein, ultimately repetition, even of what is said, is shown, not talked about.

Repetition is not the same as cumulation: it is the pinpointing of repeated elements within a structure, whether complex, cumulative, or (most probably) both. It is, bound variables aside, not a syntactic feature, but a semantic one, and is indispensable. In the confines of special logical languages, such as combinatory logics, repetition of bound variables may be eliminated, but repetition is not itself thereby eliminated, since the combinators involved are typically repeated in application many times. Such logico-structural elements are not the way in which language hooks into reality, any more than a forest of repeated ‘and’s or ‘if’s gives a logical formula a subject matter. It is through us and our understandings of elements not compounded that language engages with reality. That it can do so in ways approaching adequacy is however down to the four sources of complexity in language and its use: functorial complexity, cumulation, repetition, and the latecomer, variable binding.

10. Repetition and Argumentative Unity

In 1926, following a challenge from Jan Łukasiewicz, Stanisław Jaśkowski devised the first modern system of natural deduction, a formalization of the practice of mathematicians of proving things not by using the axioms of propositional and predicate calculus together with substitution, but by making assumptions and seeing what followed from them. The archetype of such practices is Conditional Proof: if from the assumption p one may infer that q, then one is entitled to assert the conditional $p \rightarrow q$, discharging or dropping reliance on the assumption. Jaśkowski went on to publish his ideas in 1934,[10] and the method has since become the standard way of doing proofs in elementary logic, to the benefit (and relief) of generations of undergraduates and their teachers.

Jaśkowski’s method, and likewise all subsequent forms of proof from assumptions, all rely essentially on the repetition of variables. You are only entitled to derive $p \rightarrow q$ after a subproof from the assumption that p when what you infer from p is q, and not something else. Similar remarks apply to more complex rules as well, including those for predicate logics. In modern relevance logics, it is required not only to keep track of assumptions but to keep track of *uses* of assumptions in proofs.[11] In this way, relevance logics outlaw derivations of such notorious paradoxes of implication as $p \rightarrow (q \rightarrow p)$ and $p \rightarrow (q \vee \sim q)$, which can only be proved by "straying off the point" or becoming "irrelevant".

A sequence of formulas or formulas and subproofs in a proof is not a single formula, and its complexity is not syntactic. It is kept together as a unified discourse of its kind by repetition. Nevertheless, as we know, proofs in mathematics can become extremely complex. They work, and their architects stay on the point, in good part by repetition, of variables used within and across sentences, and of constants, be they names or predicates, that are assumed understood at the beginning of the proof or are introduced in the course of the proof. All, or nearly all, proofs of any complexity make use of definitions in order to reduce repetitions and cognitive load. To take a simple but important example, in topology it is useful and important to know when two topological spaces are homotopy equivalent. It takes several lines and a lots of preparation going back to first principles to say what homotopy equivalence is. If we were not allowed to say spaces were homotopy equivalent we would very quickly lose track. Defined terms serve as mental counters, allowing us to move on more quickly to more advanced and more important results.

No one spent more time and effort in formulating the principles of correct definition than Stanisław Leśniewski. The expression of Leśniewski's requirement on what it is to be an acceptable definition of one of his logical systems of protothetic is encapsulated as Terminological Explanation XLIV of his metalogical description of that system: in abbreviated form it extends over two sides and comprises eighteen independent clauses.[12] To get to that point Leśniewski relies on over forty prior metalogical definitions, all of which rely themselves on repetitions of constants and variables, and further are expected to conform at the metalevel to equally stringent normative requirements, so the definition of 'definition' is informally expected to conform to standards as rigorous as those it formulates. Further, this definition is just one of several that Leśniewski relies on to be able to specify what counts as an acceptable extension of a logical system which starts from an axiom and has reached a certain point. All his definitions are relative to the stage of the system and are self-adjusting in their import as a result. Logical systems for Leśniewski are expandable logical stories, and as such are required to hang together in the same way as Churchill's biography, and indeed in practice more so, since, like Darwin's *Origin*, a logical system is "one long argument", the last thesis of which is where the argument has reached to date.

The interlocking sequence of definitions that make up Leśniewski's Terminological Explanations lead into his stipulations as to what kinds of expression can be correctly added to a system at a given stage in its development in order to extend it by another thesis. As such they comprise perhaps the most fully worked out set of conditions on argumentative coherence that have ever been formulated, and as anyone who has worked with them can attest, they are very intricate.

The point of this final section has been twofold. Firstly it has been to stress that what holds an extended argument together as a unity is a complex and insufficiently understood matter. We are familiar with arguments that drift off the point, and are ready with such admonitions as "That's got nothing to do with the matter under discussion!" Saying what such digressions amount is is much harder. There is in natural language no hard and fast border between staying on the point and drifting from it, even though we can tell the difference in clear cases. In logic and in more formal argumentation theory it should be possible to be more explicit about the requirements that are used without often hitherto having been formulated. In their different ways the two Stanisławs have made a good start, so that in argumentation theory as in logic, we can continue to look to the Lvov-Warsaw School for inspiration.

NOTES

[1] Pepys (1971), 253–4.

[2] I am ignoring here Frege's mature view that this expression is not a sentence with a truth value but a name of a truth value.

[3] Carnap (1934, 1937), § 3.

[4] Frege (1903), 221.

[5] Ajdukiewicz (1935, 1967).

[6] See in particular Church (1940).

[7] Simons (1988).

[8] Frege (1983) p. 24, formula 4 (1979, p. 22).

[9] Simons (2006).

[10] Jaśkowski (1934).

[11] See e.g. Anderson & Belnap (1975), 17 ff.

[12] Leśniewski (1992), 479–481.

REFERENCES

Ajdukiewicz, K. (1935). Die syntaktische Konnexität. *Studia Philosophica*, **1**, 1–27. Transl. Syntactic Connection, in S. McCall (1967, Ed.), *Polish Logic, 1920–1939* (pp. 207–230), Oxford: Clarendon.

Anderson, A. R. & Belnap, N. D. (1975). *Entailment. The Logic of Relevance and Necessity*. Princeton: Princeton University Press.

Carnap, R. (1934). *Logische Syntax der Sprache*. Vienna: Springer. Transl. *Logical Syntax of Language*. London: Kegan Paul, 1937.

Church, A. (1940). A Formulation of the Simple Theory of Types. *Journal of Symbolic Logic*, **5**, 56–68.

Frege, G. (1903). *Grundgesetze der Arithmetik, begriffsschriftlich abgeleitet*. Vol. 2. Jena: Pohle.

Frege, G. (1983). *Nachgelassene Schriften*. 2nd ed. Hamburg: Meiner. Transl. *Posthumous Writings*. Oxford: Blackwell, 1979.

Jaśkowski, S. (1934). On the Rule of Suppositions in Formal Logic. *Studia Logica*, **1**, 5–32. Reprinted in S. McCall (Ed.), *Polish Logic, 1920–1939* (pp. 232–258), Oxford: Clarendon, 1967.

Leśniewski, S. (1992). *Collected Papers*, Vol. II. Dordrecht: Kluwer.

Pepys, S. (1971). *The Diary of Samuel Pepys. Vol. I, 1660*. Ed. R. Latham and W. Matthews. London: Bell & Hyman.

Simons, P. M. (1988). Functional Operations in Frege's *Begriffsschrift*. *History and Philosophy of Logic*, **9**, 35–42.

Simons, P. M. (2006). Languages with Variable-Binding Operators: Categorial Syntax and Combinatorial Semantics. In J. Jadacki & J. Paśniczek (Eds.), *The Lvov-Warsaw School. The New Generation* (pp. 239–268), Amsterdam: Rodopi.