

Playful speculations on how language might be and why there is functional material

Abstract *Hierarchical constituent structures are a hallmark property of natural language. Traditional generative syntax either generates these structures solely in the module of syntax, as in main-stream generativism (Chomsky 1995), or different modules generate them in parallel (Kaplan and. Bresnan 1982, Jackendoff 1983). I argue that the source of hierarchical structure in language is a specific algorithm of knowledge retrieval and update. The procedure of information retrieval and update is domain-general, as it builds, maintains and retrieves general-purpose web-shaped knowledge representations. The way of operation of this component is such that its output consists in the type of hierarchical structures observed among linguistic constituents. These structures, representing segments from the knowledge representation, are mapped to and recovered from the phonological structure by a language specific mapping component. This view dispenses with the module of syntax, reducing the space between phonology and semantics to an algorithm mapping between hierarchical and linear structures. Illustrations are provided of the proposed structure of knowledge representation, of the mechanisms and outputs of the procedure of information retrieval and update, and of main operations involved in the mapping algorithm. It is shown how at a certain stage in the mapping, exactly structures attributed to linguistic expressions in generative grammar, in particular those argued for in Kayne (1994) and in cartographic approaches to syntax (Rizzi 1997, Cinque 1999, Cinque and Rizzi 2008), emerge, thus indicating that the proposed modification of the big picture does not significantly affect the empirical research into the meaning - constituency - word order interactions, or models provided to account for them. It is argued that the material in grammar identified as functional corresponds to those elements of knowledge representation which have priority in retrieval and a special role in the mapping algorithm, due to their universality and/or high frequency of occurrence. A discussion is provided of how this shift affects the architecture of grammar, as well as of some of the other general questions opened by the proposal.*

1. Introduction

The paper offers a conceptual discussion of the architecture of grammar, in particular of the nature and origin of the linguistic hierarchical constituency structures and of the central goal of the research of grammar. It is argued that rather than a recursive application of the operation merge to lexical items and to own output, a domain-general algorithm of information retrieval and update is that brings about the hierarchical constituent structure observed in language. This domain-general algorithm is responsible for creating, maintaining and retrieving a particular type of knowledge representation available to humans, and it operates over simplex concepts – using them to distinctively represent and retrieve entities via their descriptions. The core mechanisms of the algorithm effect a web-like organization of the knowledge representation.

The generative enterprise in the research of grammar aims to formulate a set of rules or constraints able to generate all and only the grammatical sentences of a language from a set of simplex syntactic items selected from a stored list of tuples termed the lexicon. Such a model enables us to discriminate 1) between selections of lexical items, termed lexical arrays, which can and those which cannot be combined into a grammatical sentence of a language, 2) between

strings of words which do and those which do not present grammatical sentences of a language, and 3) between constituency configurations (grammatical structures) which are and those which are not grammatical in a given language. Most generative approaches do this by formulating a set of rules whose recursive application to a lexicon enumerates an infinite set of grammatical sentences of a language by generating the corresponding hierarchical constituency structures in a way which is sensitive to the relevant properties of the lexical items. These relevant properties are represented as functional features.

The target phenomenon of generative grammar is grammaticality. The capacity to perform operations through which this phenomenon emerges is taken as the core of the language faculty. A modular architecture is postulated for grammar, in which the building of hierarchical structure takes place in the central module, labeled syntax. In main-stream generativism, this module is the only one which interfaces other modules. It draws lexical numerations from the lexicon, organizes them in hierarchical structures, and delivers these structures to phonology and semantics. The former takes care of the linear physical realization of these structures and the latter of their semantic interpretation. These two domains figure as a mere by-product of the narrow grammar, i.e. as a specific interpretation of the structures that the narrow grammar generates.

Let us look at an example. Consider a somewhat simplified lexical array, i.e. numeration, from the English language in (5).

- (1) {this, a, guest, teacher, is}

A viable generative grammar of English can generate from this numeration the sentence in (2a), with the constituent structure in (2b).

- (2) a. This guest is a teacher.
b. [[this guest] [is [a teacher]]]

However, also the grammatical sentence in (3a) with the structure in (3b) can be generated from the same numeration.

- (3) a. This teacher is a guest.
b. [[this teacher] [is [a guest]]]

One consequence is that syntax defined in this way is indeterministic: the same numeration may give ground to multiple different sentences. This indeterminism is not innocent, as it also means that what output is sent to the semantic interface, i.e. to what meaning syntax will map a numeration, is beyond systematic control. Another consequence is that in order to exhaustively enumerate the grammatical sentences of a language from their respective numerations, syntax needs not just to attempt to generate a hierarchical structure from each possible numeration, but rather all the possible hierarchical structures from each of the numerations. Grammatical sentences correspond to the well-formed among them (in derivational approaches termed converging derivations).

The elimination of this issue comes as a byproduct of Chomsky's (2001) introduction of subnumerations,. Each numeration, being a set of lexical and functional items, comes with a hierarchical organization: it is specified for the relevant subsets, corresponding to cycles of

combinatory computations. The numeration in (1) would then be additionally specified either as (4a) or as (4b), corresponding to (2) and (3), respectively.

- (4) a. {{this, guest}, {{a, teacher}, is}}
- b. {{this, teacher}, {{a, guest}, is}}

This step, however, escalates another foundational issue: the question how the numeration is determined. Having been assigned internal hierarchical organization, numerations include exactly the type of structure that is supposed to be generated by syntax. Especially in light of the view that exactly the capacity to generate unbounded hierarchical structures is the narrow language faculty (Hauser, Chomsky, Fitch 2002) – that hierarchical structures of the same type precede the implementation of the operation that builds hierarchical structures makes the theory inconsistent.

Importantly, Chomsky takes subnumerations to be semantically determined: they correspond to structural units which, after the spell-out to LF, yield propositions (cf. Arsenijević and Hinzen 2012, who argue that they rather match individual acts of reference). Since different subnumeration structures correspond to different meanings, semantic information is somehow available at the point of selection and pre-organization of the lexical material. How is this compatible with the strict modularity of the language architecture, where the lexical and the semantic module are fully disjoint and only communicate via syntax?

I argue that the problem is not purely technical, but rather essential: the standard generative view postulating a separate module between semantics and phonology whose task is the generation of hierarchical structures from lexical entries is wrong. Rather than being generated from lexical material, and then spelled-out for phonological and semantic interpretation, hierarchical structures observed in language are semantic in nature, and emerge in the semantic module, i.e. in the storage of conceptual information, from the procedure of its retrieval and update. The traditional notion of syntax, both as a discipline of linguistics, and as its corresponding module of language, have been taken for granted and imported without questioning. While language indeed manifests the observed hierarchical structures – these structures are essentially semantic.

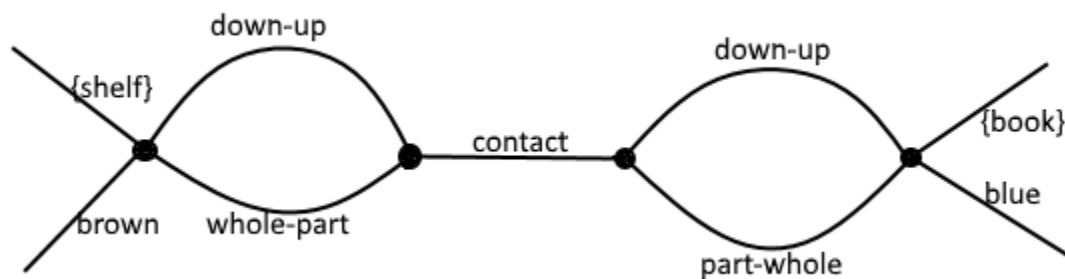
I argue that there is no syntactic module whatsoever: there are the three remaining modules: the conceptual system maintaining information states, i.e. knowledge representations, the phonological system which interacts with the articulation and perception of language, and the lexicon. And there are algorithms within these modules, as well as those mapping between them. Crucially, this is not to reject the generative research of grammar – a large portion of its analyses of the empirical linguistic material preserve their validity in the present perspective. But the answers to some of the core questions of research of grammar, from its architecture, the origin of hierarchical structures, to what makes the narrow language faculty and what are the innate aspects of language – do fundamentally change. And so does the theoretical relevance and place of the hierarchical models of constituent structure, too.

I argue for an architecture in which an algorithm which uses the lexical information transforms the linear sequence in (2a) into a hierarchical structure which is semantic in nature and presents a segment of a web-like conceptual knowledge representation (close to (2b), but with conceptual structures in place of the lexical items). The other way around too – the algorithm transforms a selection from a conceptual knowledge representation into a linear structure like the one in (2a).

This bidirectional transformational algorithm is all that stands between semantics, phonology and the lexicon – there is no module that generates structures between them. The hierarchical structure is immanent to the semantic representation, i.e. it emerges in another algorithm, one that builds, retrieves and updates the conceptual knowledge representation. In effect, the present approach demotes the intuition of grammaticality, and establishes the mapping between meaning and sound, as the central empirical object of study of grammar.

Consider a toy example of conceptual knowledge representation in (5), where the curly brackets around the concepts *shelf* and *book* mark a complex underlying conceptual structure of these items, which has been collapsed into a simple property in the interest of simplicity (i.e. in the actual knowledge representation, these nodes are replaced by complex structures). The four entities (represented by the four knots) from left to right are: the shelf (here specified as the entity with the properties of being a shelf, being brown and including a particular upper part), its upper surface (represented as the entity which is an upper part of the shelf and in contact with the lower part of the book), the lower surface of the book (represented as the entity that is a lower part of the book and in contact with the upper surface of the shelf) and the book itself (represented as the entity which is a book, blue and includes a particular lower part).

(5) The concept of *the blue book being on the brown shelf*



Entities, i.e. properties as their substance, are retrieved from and added to the representation within specific restricted segments of the aggregate knowledge representation (Barwise and Perry's 1983 resource situations). Thus, to be retrieved or updated, the segment in (5) must be a sub-segment of a bigger 'topical' segment.

As will be discussed, a consequence of the nature of the algorithm is that the information operated by it is universally a segment of a web. This means that it is a set of entities, such that each entity is represented as a set of properties, and connected to at least one other entity within the set by at least one of the properties it consists of. Only continuous web-segments can be retrieved or updated at once, i.e. the retrieval and update procedure forms and traces segments from a web. There is always one entity which is the target of retrieval, and the retrieved segment presents its optimal unique specification for a given resource situation. The target entity needs to be uniquely determined from a set of topical entities and a set of properties, which are included in the retrieved segment.

Update can be analyzed into retrieval (identifying the entity, i.e. knot, that is enriched by new properties), plus a writing-in operation which adds new knots and new properties to the representation. Therefore, in the rest of the paper, for simplicity, I sometimes only refer to retrieval even though I mean both retrieval and update.

I consider this algorithm domain-general: it is plausibly used also by our capacity of spatial navigation, by our arithmetic capacity, or by our logical reasoning capacity. It is shared by all the cognitive capacities involving logical knowledge-manipulation.

Taking this view narrows the room for a number of conceptual and methodological choices. For instance, due to its ‘generative-semantic’ orientation, the present approach implies a radically compositional view of grammar: even monomorphemic lexical items typically stand for complex hierarchical structures – as the hierarchical structure is formed by concepts rather than lexical items. Another example: in the present view, the hierarchical structures originate as ‘n-ary’ branching with one node, i.e. one relational property, having a special status – that leading in the direction of the target entity, and the others being flat (even though ordered, as discussed in section 6). The tendency of grammar to operate binary asymmetric hierarchies (Kayne 1994, Moro 2000) emerges within the mapping algorithm as a step towards linearization.

Theoretical choices and tentative generalizations in the taken approach engender a range of new questions, and outline the contours of an entire research program. For obvious reasons of space and focus, I do not discuss all the exciting issues emerging from it – but rather concentrate on the rough presentation of the general view of grammar and briefly examine the role and nature of the functional material that it implies.

Considering that the proposal introduces a very general new view of grammar and cognition, the paper cannot afford minutiose expositions or empirical analyses implementing and testing it, or even to lay out a complete theory and formalization. The goal is to sketch the main directions for modelling and research, and present potentially fruitful observations and tentative generalizations.

The paper is organized as follows. Section 2 problematizes the issue of origin of the hierarchical structures observed in grammar, and outlines some conceptual issues with the standard view in the generative grammar. In section 3, I introduce the main idea, and in section 4 I provide a somewhat more concrete elaboration. Section 5 outlines another algorithm: one that maps the hierarchical structures retrieved from the knowledge representation with the linear structures of phonology. In section 7, I tackle the issue of functional material: I argue that functional features stand for specific concepts (hence semantic material) which play a special role both in retrieval and update of the knowledge representation and in the mapping between the hierarchically structured traces of retrieval and update and phonological structures. Section 8 revisits the mapping algorithm in light of the presented view of the functional material and offers enriched structures. In section 9, I briefly consider some very general questions raised by the proposal. Section 10 concludes.

2. The origin of hierarchical structure in language

Generative grammar follows the Saussurean tradition in viewing grammar as a mapping between the physical carrier of language (interfaced via the phonological form, PF) and the conceptual content associated with it (interfaced via the logical form, LF). A third side in the mapping is the lexicon, which is roughly modelled as a set of triples of a phonological representation, a semantic content and syntactic features. The mapping itself is considered to be the task of syntax: syntax combines lexical items thus generating hierarchical structures interpretable both at LF and (but only after linearization) at PF. This architecture is known as the inverted Y model.

In the Minimalist Program (Chomsky 1995), syntax generates structures by a binary set-forming operation *merge* applying to syntactic items, and syntactic items are either units from the lexicon, or previous outputs of the operation. These structures are interpreted by the semantic and by the phonological component. At the phonological component, they are ultimately mapped onto a linear structure, which on most approaches corresponds to an additional hierarchical structure with units such as segments, syllables, phonological words and phonological phrases (the mapping between the two hierarchical structures is non-trivial, see Buring 2013). At the semantic component, the structures built by syntax are directly interpreted in terms of relations such as scope, coreference or function-argument relations. Semantics either has no hierarchical structures of its own, remaining entirely parasitic on the syntactic structure, or its structure trivially maps to that of syntax.

This situation poses a foundational question:

Why is semantic structure trivial in relation to the syntactic structure, while the phonological structure is not? I.e. why phonological structure – at least the noncontroversial linear one – emerges in phonology, but semantic structure is imported from syntax?¹

The reason why this kind of asymmetric view developed was rather pragmatic. At the time when generative syntax emerged, it was as clear as ever that language is externalized linearly, but close to nothing was known about the conceptual representations that the semantic component works with. In order to close as many open ends as possible and develop a theory of language in the narrowest sense, generative grammarians chose to focus on those aspect which can be directly empirically tested on the linguistic material, and ignore the rest, i.e. most prominently the semantic component. Only those aspects of meaning that could be shown to exhibit systematic interaction with the structural properties of linguistic strings were considered, mostly used as a tool in the exploration and empirical testing of the theory. The rest of the meaning was concealed in a black box which later was named the Conceptual-Intentional system, and treated as a forbidden territory.

One consequence of this move was that the inverted Y architecture was unbalanced in two ways, and that two roles were conflated in one module. The disbalance was in the pointed-out fact that the PF had its own characteristic structure – the linear one (with potentially some specific hierarchical organization too), while the LF had none: it simply interpreted the structure derived by syntax. The conflation was in the fact that syntax was responsible both for generating the hierarchical structures and for mapping between sound and meaning.

The conflation is aggravated if syntax is seen as the main, and perhaps the only generator of structures in language, or in an even more radical variant: that syntax is the generator of all the structures involved in the human rational thinking (Reinhart 2006, Hinzen 2006, Arsenijević and Hinzen 2012, Hinzen and Sheehan 2013).

The obvious step is to divorce the two syntaxes: the one that maps between PF and LF and the one that generates potentially domain-universal hierarchical structures, i.e. to postulate a mapping algorithm separate from the one that produces hierarchical structures. This is the line that I explore in this paper: to treat the traditional syntactic hierarchical structure as generated

¹ Another type of structure with murky origin in the Minimalist Program is that of Chomsky's (2000) sub-numerations – subparts of the set of lexical items in the input to the generating system.

by the semantic module (while outlining a possible build-up of that module) and to reduce the component between the semantic and the phonological interface to the mapping task.

Observe that the two types of structure, traditionally referred to as the phonological and the syntactic structure, can be described, respectively, as the structure among the meaningless atoms of language, and the structure among the meaningful atoms of language (cf. Martinet's 1949 notion of the double articulation). The adopted view, where the hierarchical structure is essentially semantic, allows us to divide them in exactly this way: to the structure of phonological realization (the linear structure in Figure 1) and the structure of semantic interpretation (the hierarchical structure in Figure 1).²

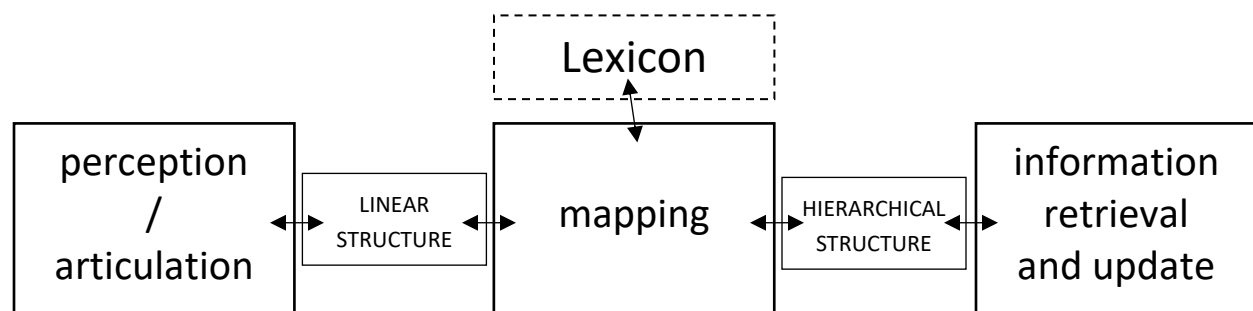


Figure 1: The proposed architecture of language

The task becomes to propose a cognitively plausible procedure which stores and retrieves information (and thus manipulates declarative knowledge), such that it generates the hierarchical structures displayed by sentences of natural language, or structures sufficiently similar to them. This task is tackled in the next section.³

² The semantic nature of the hierarchical structure is in fact implicit in the generative grammar, when non-lexicalist approaches are considered, such as Distributed Morphology (Halle and Marantz 1993), or Nanosyntax (Starke 2011a, Caha 2009). In these approaches, the hierarchical structure is built from functional features and bearers of conceptual content (roots). In such a case, where the verb *shelve* for instance underlyingly stands for *place on a shelf* (Hale and Kayser 1993) – what is the nature of the hierarchical structures themselves? They are interpretable at LF. If they only involve syntactic features, then they are some kind of semantic skeletons (perhaps somewhat enriched, as it is possible that functional features too receive semantic interpretation). If they also involve roots, then they even more clearly stand for meanings. In both cases, the hierarchical structure can be considered semantic.

³ It is clear by now that the present view is a partial revival of the generative semantics (e.g. Lakoff 1970a), which also considered that the constituent structures uncovered by syntactic analysis are deeply semantic. However, the context has significantly changed in the meantime, and many issues which seemed to be fatal for generative semantics have disappeared. For instance, many approaches to syntax nowadays explore the structure at the sub-lexical level (Distributed Morphology, Halle and Marantz 1993, Nanosyntax, Starke 2011a, Caha 2009), and arguments have been put forth from the semantic and pragmatic perspective too that words, and even morphemes involve internal structure (Asher 2011). One of the crucial arguments against generative semantics, illustrated in (i), is trivially dismissed in contemporary syntax. Lakoff (1970b) treats the verb *kill* as underlyingly equivalent to the construction *cause to die*. Fodor (1970) points out an asymmetry, illustrated in (i): the verb *kill* allows for only one modification for time, while *cause to die* allows for two. As generative semantics did not have a developed theory of the lexical interface, it was inferior to generative syntax in resolving this issue.

(i) a. *On Friday, John killed Mary on Saturday. b. On Friday, John caused Mary to die on Saturday.

3. Hierarchical structure between knowledge, logic and language

In sections 1 and 2, I argued that an architecture in which the LF structure is motivated by the properties of the conceptual domain in the way the PF structure is motivated by the properties of the sensory and motoric capacities, and the mediating component is solely responsible for the mapping is advantageous to the traditional inverted Y model. I propose an alternative architecture, on which the hierarchical structures traditionally recognized as syntactic organize simplex concepts into complex ones, rather than organizing lexical items into constituents (although translation to the latter might occur at some stage of the mapping).

In order to achieve this, I commit to a radical decomposition of meaning to a uniform bottom: to the smallest set of meanings (i.e. concepts), such that each possible meaning which is not contained in it can be represented in terms of a subset of those contained organized in a certain structure.

Let us assume the following definition of the rational thinking characteristic of humans: it stands for obtaining required bits of information by retrieving available data and implementing a logic on it. Information states on which our rational thinking operates are at any given time informationally and logically incomplete. Informationally in the sense that we not only do not have the aggregate information about the universe – but in rational thinking we do not even operate over the aggregate information we have stored in our memory. We focus on a small segment of it. Logically in the sense that only selective logical inferences are drawn, such that their results are also explicitly represented as information. Other logically inferable information is latently present, but not represented as information.

I here refer to these activated and logically enriched information states as topical segments of one's aggregate knowledge representation. Every pair of a logic and a knowledge representation define exactly one complete knowledge representation which can be derived by exhaustive implementation of the logic onto the incomplete one. In the most economical scenario, one's aggregate knowledge representation specifies the minimal information sufficient to derive the corresponding complete knowledge representation, i.e. its part that is logically inferred is maximized. In reality, however, the division is not entirely optimal, and it is also affected by how frequently certain logically inferred information occurs as topical (frequent logically derived

Today, however, on many approaches, it is standard to consider that *kill* decomposes into *cause* and *die* – in the spirit of Dowty (1979), Parsons (1990) for the semantic and Hale and Kayser (1993), Harley and Noyer (1999) for the syntactic structure. The two components may sit under one tense head, yielding the verb *kill*. Alternatively, each verbal head project its own TP, yielding the realization with two tensed verbs, each of which can be modified for time. The two structures, clearly, correspond to two slightly different meanings, but the meaning of the root *√kill* still does correspond to a combination of the meanings of the roots *√cause* and *√die*, in a particular structural configuration.

The question emerges what the generative semantic enterprise would look today. In other words, whether and how the technical developments like the ones above enable a theory which would treat the traditional syntactic structures as semantic by origin, at the same time providing a coherent account of how these structures interact with the phonology and with the lexicon. Such a view would face certain problems – most importantly the lack of a direct enough empirical access to the conceptual objects independent of the linguistic carriers. We have a relatively clear picture what stands at the phonological interface of grammar, but not at the semantic interface. In the next section, I speculate about one possible design of the purely conceptual domain: the knowledge representation and its retrieval and update algorithm, as well as the nature of its interface with grammar.

information can be stored for reasons of time and processing economy). Rational thinking is then any change from one knowledge-representation to another, such that they project the same complete representation, and the former is a subset of the latter.

In this paper I argue that the knowledge-representation component above is built, maintained and retrieved by a specialized algorithm, and that this same algorithm is responsible for the generation of the observed by syntacticians to underly constituency in linguistic expressions. While this is where its effects are most clearly observed, the same algorithm operates the knowledge representation in its interaction with other capacities such as perception, rational thinking, music.

Note that by virtue of its update capacity, this algorithm not only retrieves and updates, but actually builds the knowledge representation (potential inborne content aside, the first updates begin the process of building the knowledge representation), thereby imposing on it a particular structure determined by its characteristic retrieval and update procedure. In other words, and in line with much of the thinking in the generative tradition – the capacity behind the specific type of hierarchical structure found in language is not just one of the products of the human cognitive apparatus – it is one of its defining properties, as it determines its way of storing, retrieving and updating information (Reinhart 2006, Hinzen 2006, Arsenijević and Hinzen 2012, Hinzen and Sheehan 2013).

The narrow linguistic capacity then does not include this algorithm, and hence neither the hierarchical structures characteristic of language. Its main component is another algorithm – one that maps between the data structures of the retrieval and update algorithm and the linear phonological structures, with the help of the information from the lexicon.

4. Knowledge representation and its retrieval and update

In this section, I outline a view of the information retrieval and update algorithm generating hierarchical structures of the type traditionally ascribed to syntax, and of the knowledge representation that it builds. The algorithm is characteristic for being able to ‘move’ between the representations of individual entities in the knowledge representation in order to reach the designated one (typically from information-old to newer entities), resulting in directed web-like segments of knowledge representation. A bidirectional mapping algorithm between these segments and phonological representations enables information transfer between different knowledge representations – typically those of different individuals.

Let us assume a Leibnizian view where (at least in our cognitive representation) an entity is equivalent to the set of properties known of that entity, with one further specification: the properties can be monadic (i.e. sets of entities)⁴, or binary (sets of sets of entities). The universe then consists of entities, which have no essence to be distinguished from each other: the only way to distinguish them is via their unique sets of properties (two entities sharing all the same properties cannot be thought of as two entities).

Let the knowledge representation be modelled as a web built by the aggregate set of the available simplex properties (in the sense that they are not decomposable into a structured set of simpler properties), all of which are either monadic properties or relations. This notion of a

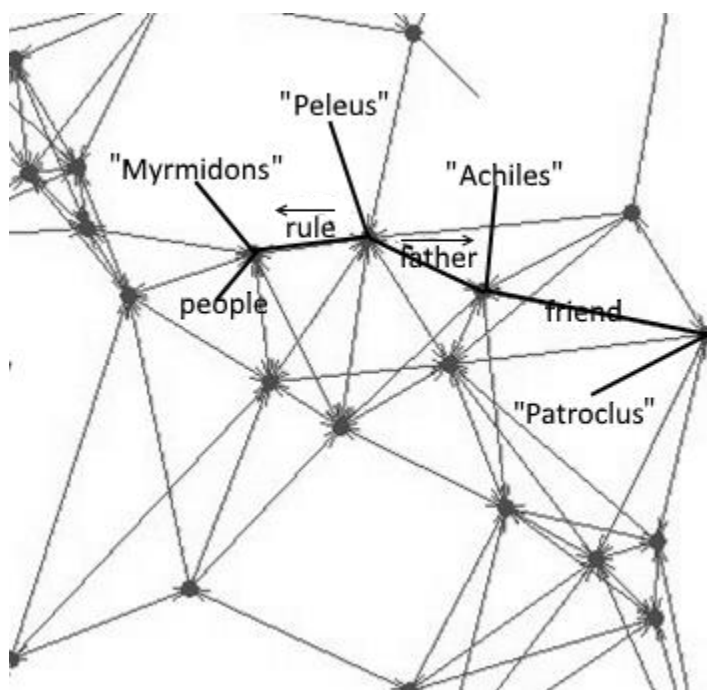
⁴ In fact, monadic properties are possibly modelled as relations, one argument of which is a percept, emotion, or another generic concept.

property matches the notion of a simple concept as used in sections 1-3, and will be used instead for reasons of precision in the rest of the paper.

Relational properties play a crucial role, because enable the algorithm to move from one entity to another, and thus provide the knowledge representation with properties of a web. An entity is any distinctive set of properties specified in the knowledge representation, i.e. any distinct knot in the web.

In (10), for illustration, a simplified segment from such a web is given. Entities are represented by knots and properties by labeled lines (asymmetric properties are joined by an arrow for the direction). Interpretations are straightforward: every line is a predicate, and each predicate applies to one or two arguments represented by the knot(s) at its end(s). The example includes the specification of an entity named Myrmidons, another entity named Peleus which rules the former, an entity named Achilles which is his son, and an entity named Patroclus which is friends with Achilles.

(6) The web-like knowledge-representation built and retrieved by the algorithm (a segment)⁵



At this point, many important aspects of knowledge representation are unclear or simplified – some for brevity, others because they yet need to be explored. Some are supplemented and discussed in the rest of this paper, while others are left for another venue.⁶

⁵ I treat names as one place predicates, and consider their definite behavior to be derived (see e.g. Boër 1974, Fara Graff 2015). This is why they are given in quotes.

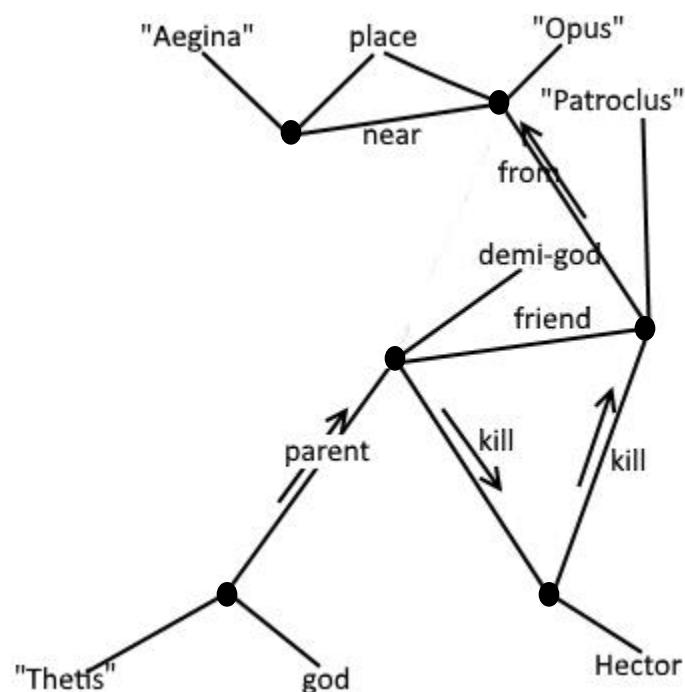
⁶ It is important to note, however, that the knowledge representation is not necessarily represented as a web. In fact, it can be represented in a declarative way, as a set of minimal propositions (each consisting of one monadic or binary predicate), as long as the retrieval and update algorithm is specified so that the entities involved in these propositions act as addresses, each proposition being addressed by the two entities occurring in it. The retrieval then at any point only has access to the addresses contained in the currently processed proposition. It can switch to another entity on any proposition based on a binary predicate – by switching from the address where it retrieved

Assume that communication always proceeds between two knowledge representations. The speaker compares her own knowledge representation and the one in which she represents her shared knowledge with the hearer. She identifies a segment in her own knowledge representation which contains the intended update and through a web-segment uniquely specifies the way of incorporation in the shared knowledge representation. She then updates the shared knowledge representation.

Communication hence goes in parallel within each interlocutor: each of them maintains two representations, and the success of communication depends on their ability to keep the shared knowledge representations sufficiently similar to each other (a failure triggers explicit negotiation and synchronization). In the ideal case, identical update of the respective shared knowledge representations takes place in parallel in the speaker's and in the hearer's mind.

Imagine now a conversation between two shepherds from the surrounding of Troy during the siege. One of them, shepherd A, is well informed, the other, shepherd B, ignorant. They are looking at a unit of soldiers camping on the beach, and shepherd B asks who they are. Shepherd A knows that they are Myrmidons, and has an accurate shared knowledge representation with shepherd B, given in (7). They do not share the knowledge of who Myrmidons are, which lends ground to shepherd A to assume that shepherd B does not know who Myrmidons are. Their shared knowledge includes, however, Patroclus and Hector, known by their names as well as for Hector having killed Patroclus; additionally, it includes a friend of Patroclus known for having killed Hector and being the son of the goddess Thetis and therefore a demi-god (but shepherd B does not know his name) and the two locations Aegina and Opus, including that they are near each other for some given standard and that Patroclus comes from the latter.

(7) Relevant knowledge possessed by shepherd B (according to shepherd A)

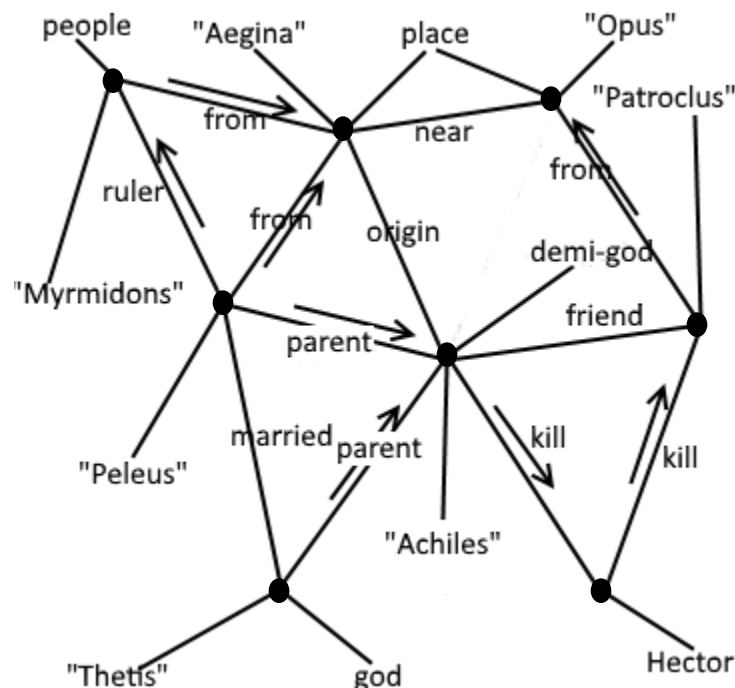


the current proposition to the address of its other argument. The web-like representation is just a more direct and compressed way of representing the procedure and the output of the algorithm.

Shepherd A also maintains a rank of relevance of the different entities figuring in the conversation, where for instance Patroclus and Hector are equally prominent, more so than Thetis, and she is more prominent than the two locations, Aegina and Opus.

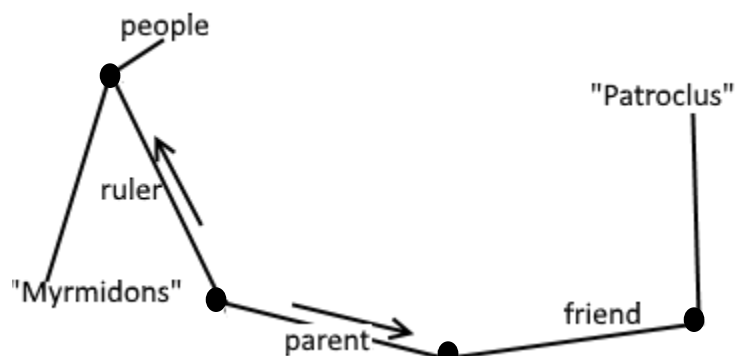
In order to answer shepherd B's question who the people are, shepherd A retrieves his own knowledge representation, the relevant segment of which is given in (8), and selects the subsegment which is optimal regarding minimality of connections from Myrmidons to the shared knowledge and the relevance of the involved entities, given in (9). In addition to shared knowledge, shepherd A knows Achilles's name, that his father was Peleus, the ruler of the Myrmidons, that both Peleus and the Myrmidons come from Aegina – which is therefore a place from where Achilles draws his origins.

(8) Relevant segment from shepherd A's knowledge connecting Myrmidons to shared knowledge



The optimal path from the shared knowledge content to the knot in the upper left corner standing for Myrmidons needs thus to be established based on minimality and relevance. In the given case, this is the path in (9). Within the given segment of knowledge representation, the identified subsegment (to which I refer as the trace of retrieval) uniquely determines each of the remaining entities involved, successfully incorporating Myrmidons into the shared knowledge representation.

(9) The retrieved optimal subsegment from Myrmidons to shared knowledge



This is where the domain-general phase of processing ends: the retrieval and update algorithm has selected an optimal segment with which to update the shared knowledge representation. From here, the mapping algorithm, a narrow-linguistic capacity, takes over. Its task is to map the retrieved subsegment to PF, i.e. to a linguistic expression – or possibly to an entire set of linguistic expressions which can be ranked on the basis of different criteria to select the best candidate considering pragmatic aspects such as new vs. old information, presuppositions, implicatures or communicative intentions. In the described communicative situation, that sentence, or one of those sentences, would be: *(Those are) the people ruled by Patroclus's friend's father.*

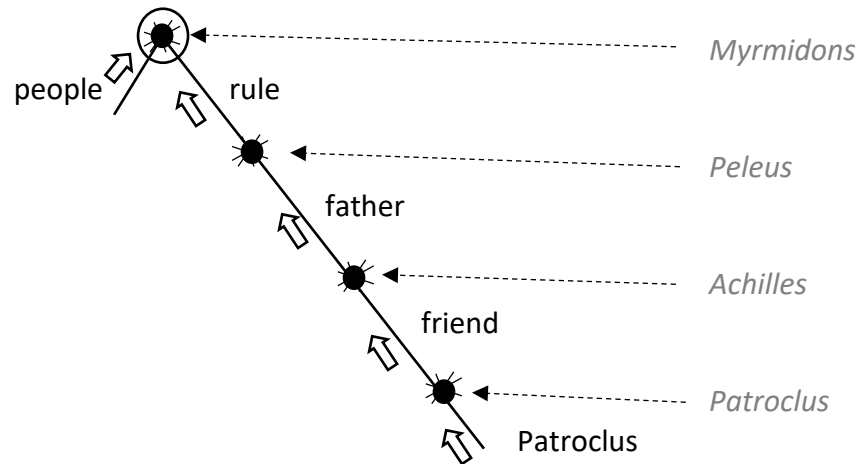
If, for instance, alternatively, shepherd B knows that Myrmidons are a people, but does not know who rules them, shepherd A can update him by retrieving exactly the same retrieval and update trace in exactly the same way, but with a somewhat different division into old and new information, determining also a somewhat different PF: *Myrmidons are ruled by Patroclus's friend's father.*

Assume now that someone knows enough about Myrmidons to distinguish them from other entities (for instance, that they are a people, and that they are called Myrmidons, unlike any other people), as well as about Peleus (for instance that he was the father of Achilles – but not that his name was Peleus), Achilles (that he was Patroclus's friend – but not that his name was Achilles) and about Patroclus (that he was called Patroclus, among other possible known properties). Then one can connect two subsegments, one departing from the Myrmidons and reaching Peleus via the relation *rule*, and another departing from Patroclus and reaching Achilles by the relation *friend*, by specifying that their ends stand in the *father* relation. The establishing of the connection updates the knowledge representation, and the simplest linguistic realization of such an update operation is *The ruler of the Myrmidons was the father of Patroclus's friend* (i.e. *The father of Patroclus's friend was the ruler of the Myrmidons*). This is another case of update.

In (10), I graphically represent some subsegments of the type above, following a simple convention. While entities are still represented by knots, and properties by nodes, I coded retrieval as directed bottom-up (to additionally specify this, I use arrows to mark the direction of retrieval towards the target entity). The process of retrieval (as in (10a)), or update (as in (10b)) is represented in the bottom-up direction in order to point out the similarities with the generative hierarchical syntactic structures. A segment of the knowledge representation always ultimately specifies exactly one entity, through a subset of its properties. In update, part of the segment is introduced into the knowledge representation – under the condition that the rest of the segment

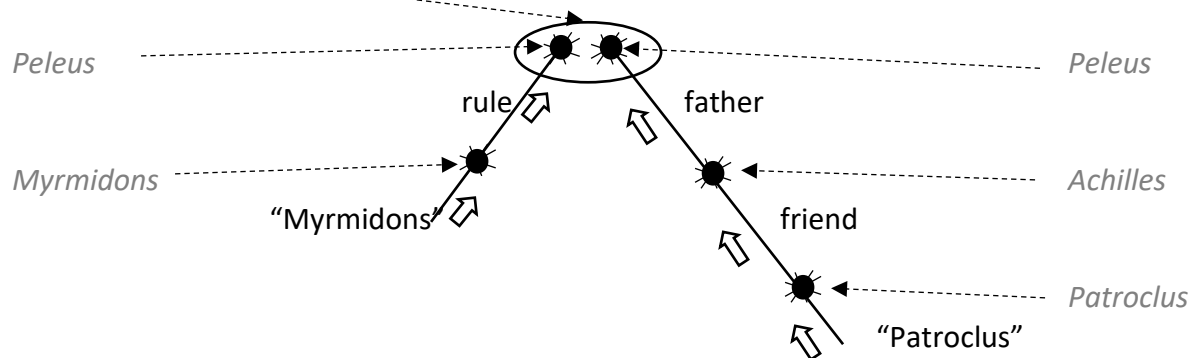
– its old content – suffices to identify the targeted entity. Each knot stands for a larger set of properties and relations, graphically represented by the short lines stemming from their centers. As these stand for properties which are not used for retrieval or update, their other ends are left out, and they are not marked by arrows. Only those that have been retrieved for distinctive representations are represented by full lines with property-labels.

(10) a. *the people ruled by Patroclus's friend's father*



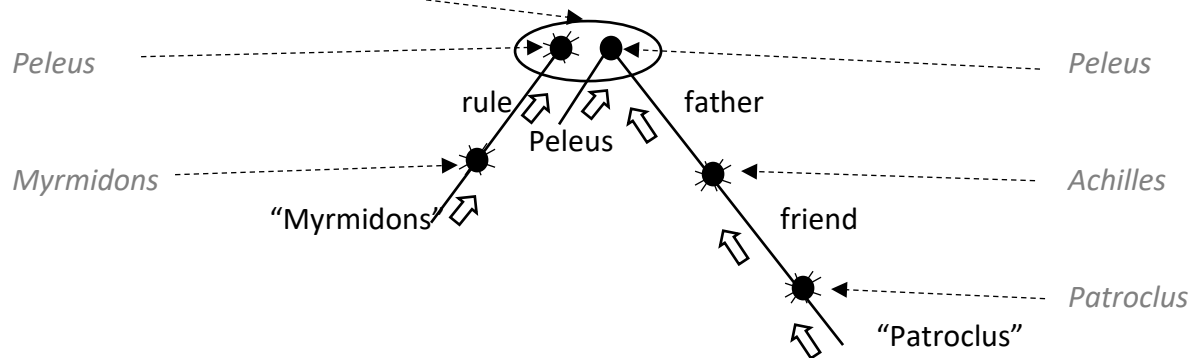
b. *The ruler of the Myrmidons was the father of Patroclus's friend.*

Update (the unification of the two nodes)



- c. The ruler of the Myrmidons was Peleus, the father of Patroclus's friend.

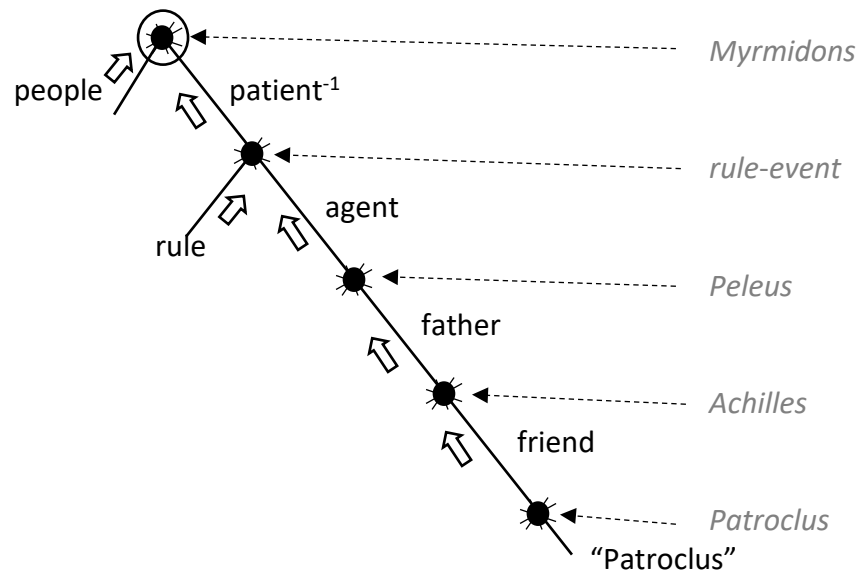
Update (the unification of the two nodes)



The fact that the retrieval and update procedure is oriented, as represented by the arrows, is responsible for the linguistic tendency to express relations in asymmetric ways, such as figure-ground asymmetries (consider the symmetry of the relations denoted by asymmetric prepositions *next to*, *by*, *near*, i.e. pairs of expressions for the same constellation, such as *above* and *below*, or *ahead* and *behind*), or by subjecthood (in particular passives, middles, anticausatives).

As already pointed out, the representations discussed are incomplete in many ways. For instance, in order to handle adverbials, this system needs a modal Davidsonian extension along the dimension of types (introduction of eventualities and worlds). Consider the enrichment in (11), where a knot standing for the eventuality is added to the representations from (10).

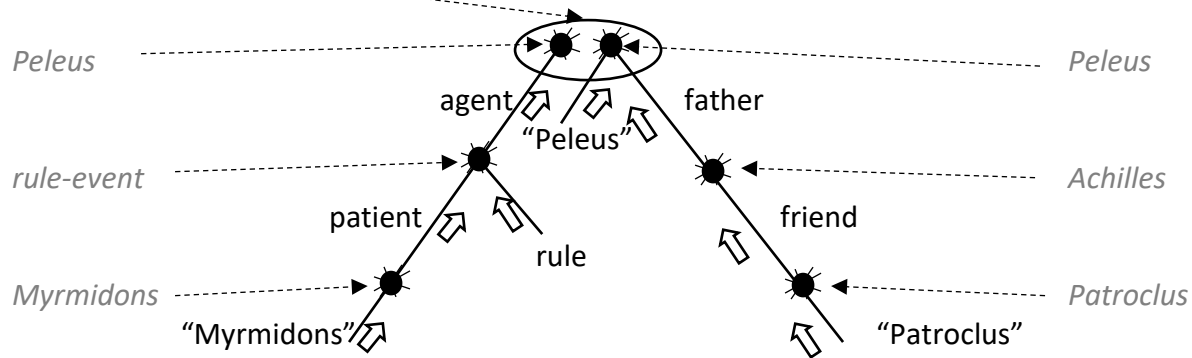
- (11) a. the people ruled by Patroclus's friend's father⁷



⁷ The marking *patient*⁻¹ specifies that the retrieval goes from the eventuality to the participant rather than the other way around: it stands to *patient* like *parent* stands to *child* – they denote the same relation, just with the opposite orientations.

b. *The ruler of the Myrmidons was Peleus, the father of Patroclus's friend.*

Update (the unification of the two nodes)



As previously noted, relations can be asymmetric in the sense that the two nodes linked by the property *father*, or by the *patient* relation have different roles in this relation. This means that the relational properties should be represented as oriented links, whose orientation is vacuous for symmetrical properties. In such cases, language often develops pairs of lexical items realizing distinguished only by the orientation of retrieval and update relative to the orientation of the property-link (*parent-child*, *possessor-possessum*, *patient-affect*).

The emerging structures are hierarchical and potentially infinite, and they resemble the structures used in the generative syntactic modelling – except that they display unrestricted multiple branching (it is for simplicity that I offer examples with a maximally ternary branching). Every knot is located by a set of properties and relations, and each relation offers the opportunity to move to another knot, where the procedure recursively reapplies.

Let us finally consider the linearization of the structure in (11b). The structure involves three binary (Patroclus, Achilles, Myrmidons) and two ternary branching nodes (Peleus and the ruling event). All branches are directed, and all except for one represent asymmetric relations. Considering that the mapping relies on the lexicon, i.e. that it replaces units of information by lexical items, it is reasonable to assume that the characteristic of each relation be lexically realized. Let us specify these lexical realizations, using the English lexicon, see the underlined words in (12). Now the mapping to the linear structure additionally needs to specify the mapping between the lexical items and nodes.

as prepositions and suffixes restricted to items with the same variable₁ – we even get even those in (15b', c').

(15) a.

<"Myrmidons", a>,	<i>Myrmidons</i>
<patient, a, b>,	<i>of/-ed</i>
<rule, a>,	<i>rule</i>
<agent, a, b>,	<i>by/-er</i>
<"Peleus", a>,	<i>Peleus</i>
<"Patroclus", a>,	<i>Patroclus</i>
<friend, a, b>,	<i>friend</i>
<father, a, b>	<i>father</i>

- b. Myrmidons -ed rule by Peleus Patroclus friend father
- b'. Myrmidons ruled by Peleus Patroclus friend father
- c. Myrmidons of rule -er Peleus Patroclus friend father
- c'. ruler of Myrmidons Peleus Patroclus friend father

While a parrot that produces (15b' or c') from (12) would quickly make it to the news and top journals in comparative cognitive science, these strings are still far from being expressions of the English language. Not just they do not fully match the intended English sentences, but they also remain at a level of significant underspecification. The last four words in both cases are basically a word salad and allow for at least a dozen of pragmatically reconstructed meanings.

Note that the claim here is not necessarily that the retrieval and update algorithm discussed is only available to humans. Weaker versions are possible too, such as that it is available to other animals, but only in limited domains and with limited mappings (see Arsenijević 2008 for a proposal that in other animals it is limited to the domain of spatial cognition). What is special for humans in that case is the domain-general nature of the algorithm, as well as the availability of a mapping algorithm which bidirectionally maps the hierarchically structured retrieved subsegments with linear representations.

5. Linear compression

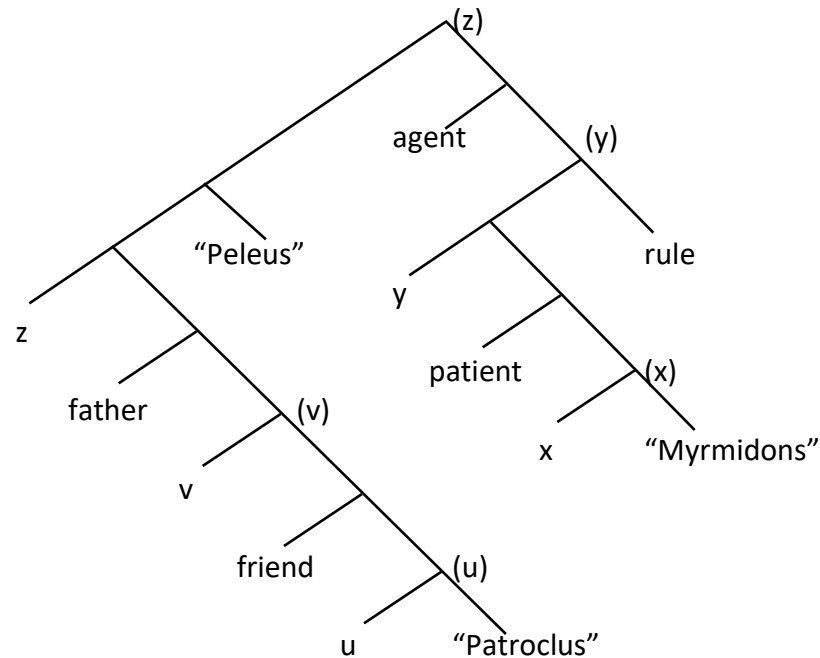
The structure we arrived at in (14), repeated here for convenience as (16) contains redundant material (each variable occurs at least twice), and fails to exploit the resources of the linear order (i.e. its linear orientation) to reduce the amount of phonologically visible material.

(16) {<"Myrmidons", x>, <patient, y, x>, <rule, y>, <agent, z, y>, <"Peleus", z>, <"Patroclus", u>, <friend, v, u>, <father, z, v>}

Introduction of general procedures for the linearization of tuples sharing an argument may lead to an overall simplification of the phonological representation. Consider the following possibility: the original hierarchical structure is transformed into a hierarchy of triples (for monadic properties – pairs), such that a triple may replace an entity that is identical to the triple's first argument. Thus, <patient, y, x> and <agent, z, y> collapse into <agent, z, <patient, y, x>> because <patient, y, x> in the argument position stands for its first argument y. When more than one such transformation are possible – the most economical one wins. For the representation in (12), this derives the structure in (17), provided both in the set-representation and schematically. Variables in brackets in the tree in (17) mark the positions where a variable has been replaced by a triple,

and the ordering between the property and its two arguments in the schematic representation iconically represents that from retrieval: <target-argument, property, departure-argument>, with reference projecting along the left node (highest projections of each referent are marked in brackets).

(17) {<agent, <"Peleus", <father, z, <friend, v, <"Patroclus", u>>>>, <rule, <patient, y, <"Myrmidons", x>>>>}



Observe now that the basic unit of structure used to transform the subsegment delivered by retrieval and update, the triple of a property and two arguments, matches the traditional generative-syntactic phrase, encompassing a specifier (the target-argument in the direction of retrieval), a head (the property) and a complement (the departure-argument). Rather than stipulating this structural unit as the smallest non-atomic constituent based on empirical insights suggesting its existence, the present view derives it from the central role of relational properties in the knowledge representation and the requirement to map the hierarchical semantic structure with a linear one. The ordering of the triple is entirely arbitrary from the perspective of retrieval and update, but has consequences for linearization. It is hence expected to find variation across languages or across finer parameters (different families of properties may be specified for different orders). This is exactly what is linguistically observed as the head-directionality parameter (Greenberg 1963, Chomsky 1981, Baker 2001, a.o).

Note, moreover, that the structure in (17) is strictly binary branching, just like the cartographic generative-syntactic structure (Kayne 1994, Rizzi 1997, Cinque 1999). While Kayne motivates the strict binary branching by the linearization requirements and the methodological minimality, the present view additionally provides a concrete operation that derives it: the replacement of a variable by a triple pointing to the knot it stands for. This operation is straightforwardly motivated: it exploits a formal property of the source structure (sets of triples) to translate it into

a more economical one (fewer elements – decreased by the number of replaced variables) while at the same time making it systematically linearizable.

Assume now that each property in the structure in (17) is assigned one lexical item, that variables are not lexicalized (after the transformation to the binary-branching form they are obsolete), and that the structure is linearized along the lines of Kayne (1994). Knowing which properties are monadic (if any) and which are binary suffices to properly reconstruct the hierarchical structure from its linearization. This makes the third gain of the transformation: bidirectional mapping, i.e. strict bidirectional recoverability.

For an easier comparison with Kaynean structures – in the one in (17), each node taking a ‘word’ (i.e. a non-variable) corresponds to a head, and each node taking a variable, as well as each structure standing for a triple (and marked with a variable in brackets at its top) stands for an argument – a specifier or a complement. Any other ordering between the property and its two arguments would do the job equally well (yielding strong head-initial and head-final languages and other related parameters).

This stage in the mapping is thus more or less the farthest point that the traditional generative syntactic analyses reach coming from the direction of the surface phonological string as its departure point in the reconstruction of linguistic structures. Syntacticians observe ordering and semantic properties of realized linguistic expression, and describe hierarchical structures which underlie them. They then consider these structures to stem from the operations combining words (and word-combinations), and strive to model this combinatorics, within a dedicated discipline (and level / module) of grammar labeled syntax. I argue that it is not words, but rather concepts that the combinatory module operates on.

My proposal is far from being the only such attempt – similar enterprises of looking for the hierarchical structures in the domain of meaning can be found not just in generative semantics (Lakof 1970), but also in Jackendoff (1983), Pietroski (2018) or in non-generative formal approaches to syntax, such as the Categorical Grammar – see in particular Cremers et al. (2014). Arguments for such a continuation towards the conceptual domain are not just conceptual (as in section 2). The proposed web-like knowledge representation is plausible in light of the network-organization of the brain, of the abundance of lexical items in natural language denoting binary predicates (transitive verbs, relational nouns, gradable adjectives, prepositions) – and in particular of the simple and logical way from the web to binary branching hierarchical representations just illustrated.

6. Three relevant dimensions: categories, universality and frequency

The algorithm of knowledge retrieval and update is sensitive to a classification of properties along three dimensions.

One includes distributional regularities among simple properties. There are sets of properties which either do not cooccur, or if they do – they are resolved into a single property before being interpreted by the algorithm for retrieval and update (for instance *blue*, *yellow*, *red*, or *round*, *rectangular*, *triangular*, or similarly properties standing for different degrees along a scale such as height, weight, spatial distance). There are also pairs of sets of properties such that particular members of one set are conditioned by particular members of the other (for instance the temporal ordering relational property *before-after* is conditioned by the ontological category *eventuality*, while the spatial relational property *proximal* is conditioned by the ontological

property *material*, indicating that the two properties acting as conditions, with other such properties, form a category). Categories of simple properties directly correspond to the traditional functional categories. Some of them, further, have special characteristics directly relevant for both algorithms under discussion. These categories also correspond to the traditional functional features (in cartographic syntax, see Rizzi 1997, Cinque 1999, Cinque and Rizzi 2008, all categories correspond to functional features). Member properties of categories which act as functional features also show some relevance for structural relations, which is why most approaches to grammar treat them too as functional features, i.e. as values thereof.

The second dimension is universality: properties can be universal, domain-universal or non-universal. This dimension applies to categories of simple properties. Universal properties are properties from categories which are obligatorily specified for every entity. Among them are the ontological class (material individual, abstract individual, eventuality, state, degree) and properties of quantity (atomicity, unit-of-counting specification or absence thereof). Properties from universal categories are those properties mentioned above, which are traditionally also identified as functional features, occurring as values of the feature corresponding to their category.

Domain-universal property-categories are property-categories which are not universal themselves, but their specification is implied by particular members of some universal property-categories. Temporal location is a property implied by the specification of an entity as an eventuality, relation to at-issue worlds and/or the actual world by the specification of an entity as a situation.

A prominent characteristic of universal categories is that they are relevant for the computation of logical inferences. This characteristic is shared by some of the domain-universal categories (size, shape, dynamicity), which makes them more prominent than those which lack it.

The remaining properties are non-universal. Universal properties are computed before domain-universal properties, and domain-universal properties before the non-universal ones (sibling, quick, possessed).

The third significant dimension of properties is their frequency of occurrence. Both global and local frequency (the latter relating to recent instances of retrieval and update), of both individual properties and property-categories, play a role. It is also relevant how frequently a property or a property-category is subject to update (i.e. how often a property occurs as the one added to the shared knowledge representation). Universal properties display maximal frequency, and the order of computation of others, within their class (domain-universal, i.e. non-universal), is determined by their frequency characteristics.

7. Functional material

The point of departure of the present proposal is that language maps between the structure of our representation of information and the phonological structure, where the former has only one uniform kind of building blocks: simple properties. It is, however, a common observation of all approaches to linguistics that certain items in language play a special role, one which is grammatical rather than content-bearing. These items, which are referred to as functional, and typically modelled in terms of functional categories and/or functional features, thus also correspond to simple properties or property-structures, just like the non-functional, lexical items. The immediate question to answer is how these asymmetries emerge.

As indicated in section 6, in the present approach, these functional items correspond to property-categories and some of the member-properties: those that are universal, and those that are relevant for logical computations. Their special status emerges from special roles that they play in the two algorithms, and in logical inferencing over the retrieved information. Hence, technically, due to their status along the three dimensions outlined in section 6.

It is reasonable to assume that the retrieval of a property is faster if its category is universally present in the description of every entity. This is then a safe bet for the retrieval algorithm, a deterministic aspect, which makes the procedure computationally cheaper. For this reason, universal properties are the first to be retrieved, and in the ideal case they are sufficient to identify the referent (yielding pronominal reference). Once the universal properties are retrieved, another domain becomes deterministic – that of domain-universal properties, so the same reasoning applies to them. If the information assembled is still insufficient to distinctively describe the entity, non-universal properties are retrieved. The corresponding functional items thus correspond to deterministic, more quickly retrieved properties.

Frequency vacuously applies to universal property-categories, but is relevant for the domain-universal and non-universal ones. When more than one property-category are implied by a combination of universal properties, those of higher frequency are more probable to be found. This makes the procedure where they are retrieved first more likely to succeed, and therefore they are retrieved first. The next most frequent property is retrieved next, unless a distinctive description has been reached. The same holds among non-universal properties: non-universal property categories of higher frequency are retrieved before the less frequent ones, until a sufficiently specific description is retrieved.

As the process of information retrieval and update involves logical computations (such as deriving kind-entities and quantifier expressions from sets of entities sharing one or more properties, or inferences from transitive identity- or temporal relations), domain-universal properties which are more relevant for logical operations (shape, size, temporal ordering, causal, part-whole or quantity relations) are retrieved and updated before those that are not (color, material, emotion). This is why universal properties, such as the ontological class or properties of quantity, are additionally relevant: they fundamentally affect not just the linguistic, but also the logical computation of information. Domain-universal properties, and possibly even the non-universal ones (though on this approach it is more likely that all logically relevant properties establish relations with universal properties and therefore are at least domain-universal), can be logically relevant too, which promotes them in the order of retrieval relative to those without logical relevance. To sum up: core functional items (those corresponding to universal properties) are all logically ‘active’, and logically relevant non-universal properties are more likely to receive the status of functional items than the logically irrelevant ones.

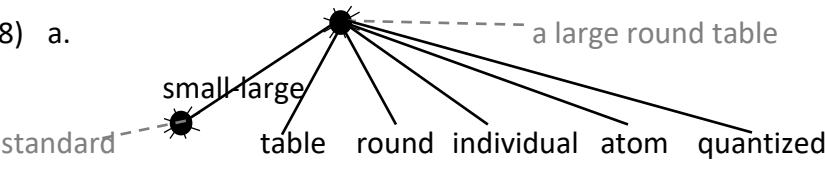
A further way in which universal property-categories are useful for the algorithm of retrieval and update is that the number of occurrences of any universal property-category in a subsegment specified by retrieval and update reveals the number of entities included in it. Technically, this means that a universal property-category can fulfill the role that the representation in (17) uses variables for: to represent the identity of entities in a linearized conjoined representation of the retrieved segment.

The mapping algorithm too benefits from a strict order of retrieval. Without such a restriction, it would face optionality in the order in which it subjects the properties to the procedure of

embedding illustrated in (17). Considering that two triples sharing the higher argument have two possible embeddings, three have six, and n triples have $n!$ possible embeddings, the determinism that ordering the embedding after the order of retrieval brings makes this process much cheaper and faster.

Properties which serve such special purposes plausibly also have a special treatment in the lexicon: they are among the few individual simple properties that have own lexical items (others are only lexicalized as parts of complex lexical meanings), and due to high frequency – the lexical items realizing them are likely to be phonologically simple. These are all aspects of being functional.

As indicated in section 6, cartographic syntactic generalizations (Rizzi 1997, Cinque 1999, Cinque and Rizzi 2008) plausibly emerge from the ordering of property-categories in retrieval and update. Consider a knowledge representation segment as in (18a).⁸

- (18) a. 
- b. <<<x, class, <a, atom-quantized>, round>, large>, table>
b'. *<<<x, class, <a, atom-quantized>, large>, round>, table>

Due to a presumably higher frequency of retrieval of the property-category *size* than of the property-category *shape*, the retrieval of the segment in (18a) will always give the structure in (18b), never that in (18c). The resulting word order will depend on language-specific parameters of the mapping algorithm, yielding surface realization obeying the regularities observed and investigated within the cartographic theory of syntax. This view makes an empirical prediction, namely that the cartographic hierarchies reflect frequencies of retrieval. While it is hard to measure frequency of retrieval of a property-category in the human memory directly, various indirect measures are possible as a way to test it.

In section 8, I show how linearization and recoverability of hierarchical structure too rely on the ordering of computation of property-categories in the algorithm of retrieval and update.

The present view predicts two types of systems when it comes to the way non-functional properties are mapped to lexical items. One is that there is no item in the lexicon without any functional content.⁹ In this type of languages all words are functional, in the sense that they are all specified for at least one functional property, typically the ontological category (an individual, an eventuality, a degree) and the properties of quantity (homogeneous, quantized). Words

⁸ The concept *table* is a simplification in the interest of exposition: the actual knowledge representation would have a complex structure in its place, with properties like prototypical function, shape, size, material conspiring to represent the concept of a table. In such a case, the given properties *large* and *round* would also occur within or closely related to the properties of prototypical size and shape. Moreover, a proper representation of the property *large* would specify These aspects of the representation belong to the tasks for future research.

⁹ Arsenijević (2020b) argues that this is the case in the Indo-European type of languages, where even the classes of words carrying non-functional properties are obligatorily specified for the ontological class and the domain-universal properties they imply. Nouns are specified for describing any ontological class of entities, verbs for eventualities including all their subsets (thus eventualities, states, degrees) and one macro-class consisting of the traditional prepositions, adverbs and adjectives being unspecified for an ontological category (hence unable to refer without other expressions).

realize those combinations of universal and non-universal properties which are pragmatically licensed and frequent in use (yielding count nouns, telic verbs etc.). This is what we find in languages of the Indo-European type, for instance, with words clearly bearing lexical categories and aspects of properties of quantity such as count vs. mass nouns or telic vs. atelic verbs. The other scenario is that functional and non-functional properties are strictly separated – there are functional and lexical items, with a broad space of combinatorics which is only restricted by pragmatic considerations (yielding uncategorized words which combine with functional items to be interpreted, such as ‘nouns’ with classifiers, ‘adjectives’ with adnominal markers). This option is instantiated in Papuan languages, where the same word often may combine both with a noun-class marker and with a verb-class marker, or be adnominally used (cf. Klammer 1998, Kratochvíl 2007). Languages may show hybrids, i.e. different combinations of both options.

A coarse-grained look at the empirical picture complies with the present view. Traditional functional features mostly relate to the universal property-categories like ontological classes (situations, individuals, eventualities, degrees – closely related to lexical categories) or properties of quantity (gender, classifiers, number, telicity), as well as to the plausibly frequent domain-universal ones (proximity – implied by the unrestricted ontological class, grammatical aspect and tense – implied by the ontological class of eventualities). Cartographic observations match the view of the relevance of frequency – to the extent that the order of categories is empirically confirmed to match the order of frequency of the corresponding properties in retrieval and update generally (i.e. also in non-linguistic use of the retrieval and update algorithm).

8. The mapping algorithm

Once I have introduced the special status of universal and domain-universal properties, let me illustrate it on an example. The representation in (19) includes universal properties: ontological classes and properties of quantity. It is a more realistic representation of a retrieved segment than the earlier ones – although still a simplification.

the book

patient⁻¹

quantized atom individual

the give-event

eventual. homogen. dynamic

recipient

agent

John

quantized atom human "John"

rule

quantized atom human

father

quantized atom human

friend

quantized atom human

Bill

quantized atom human "Bill"

Bill's friend's father

Bill's friend

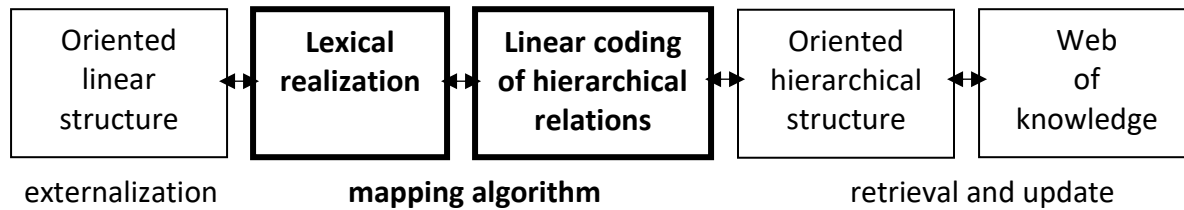
(20) a. {<quantized, x>, <atom, x>, <individual>, <book, x>, <patient⁻¹, x, y>, <event., y>, <homog., y>, <dynamic, y>, <rule, y>, <recipient, y, w>, <quant., w>, <atom, w>, <human, w>, <"John", w>, <agent, y, z>, <quant., z>, <atom, z>, <human, z>, <father, z, v>, <quant., v>, <atom, v>, <human, v>, <friend, v, u>, <quant., u>, <atom, u>, <human, u>, <"Bill", u>}

The representation in (20b) allows for a further simplification. As each variable occurs only once, and each series of nesting stands for a unique description, identifying in the given reference domain a unique entity – the variables are redundant. Rather than dispensing with them, I replace them with a signal that the nesting has reached its end, for which I use the symbol \$. This is illustrated in (21).

The task of the mapping algorithm is to map between the subsegment identified by retrieval and update (an oriented hierarchical structure) and the phonological realization (an oriented linear structure), in an architecture more or less as in (22), without significant information-loss. The

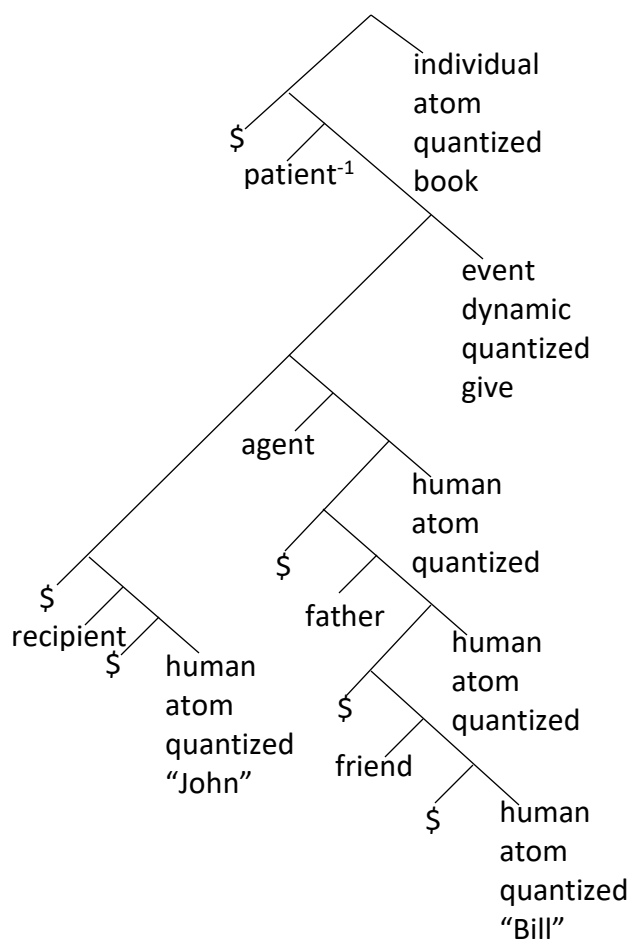
hierarchical constituency relations contained in the trace of the retrieval and update must be linearly coded.

(22) The scope of the mapping algorithm



Assuming that every ontological property in (21) (i.e. *individual*, *human*, *event*) signals the beginning of the description of an entity, every stop sign (\$) its end, and every relational property – i.e. every point where the ordering of properties mismatches their order of retrieval – a point of embedding of another entity, this string fully recoverably encodes a hierarchical structure. In this way, the linear structure, which has a smaller expressive power than the hierarchical structure, by ‘multiplying’ its expressive capacity of the linear order of elements with the independently specified linear order of properties regarding retrieval, lifts its power to the square and manages to encode hierarchical information. In (23), I provide a tree-representation of the hierarchical structure encoded in (21), where again for simplicity multiple monadic sister-properties taking the same first argument (i.e. with the same orientation) are collapsed into chunks.

(23)



An additional space of structural transformations is probably available at the interface of the structures of this type with the lexicon, in order to enable efficient lexicalization of structural chunks from a limited lexicon. Similar operations have independently been proposed as spell-out driven movement in nanosyntactic approaches to lexical realization (e.g. Pantcheva 2011).

While the picture is now somewhat clearer, a lot remains unclear, of course. The structure in (23), for instance, still corresponds to about a dozen of well-formed expressions (*The book was given to John by Bill's friend's father, the book (that was) given to John by Bill's friend's father*). To complete the picture, the model needs to include the maintenance of the topic and resource situations, the communicative intensions, the (mis)matches between the speaker's and the shared knowledge representation, the lexical inventory, aspects of economy in the interaction between all these components. These are topics which are independently investigated in linguistics irrespective of this proposal – and their interaction with the proposed view is a natural topic for future research.

9. The outlook

While it is illusory to try to even list, let alone answer, all the questions about and emerging from the proposed view, in this section I try to ask and comment on a few of the most prominent ones.

What drives the retrieval and update?

Obviously, the retrieval must be sensitive to our intensions and to the differences between the two relevant speaker's knowledge-representation: that of the shared knowledge and that of the speaker's own knowledge. Further, it needs to interact with the logical reasoning. For instance, when retrieval finds that a set of entities shares one or more properties, logical reasoning may trigger the introduction of an additional kind-entity standing for all the existing and potential entities sharing the respective properties – a move licensed by considerations of logic and economy. By assessing kind and individual entities, it may enrich traces of retrieval and update with additional logical properties, such as quantification or disjunction.¹⁰ To achieve this, combinations with theories like bi-directional Optimality Theory (Blutner 2000) or Rational Speech Act models (Frank and Goodman 2012) are a natural strategy. Its combinability with such theories is a significant advantage of the present approach compared to the traditional generative architecture where the only input to grammar comes from the lexicon, insensitive to any cognitive embedding (such as intentions or logical reasoning).

How is the syntactic theory affected by this proposal?

On the one hand, syntactic theory is fundamentally affected, as the module, or level of representation of syntax as traditionally thought of is effectively eliminated. Rather than combining lexical items based on a set of rules and/or constraints, thereby producing hierarchical structures, the traditional matter of syntax is divided into two components: one which is relegated to semantics (the generation of hierarchical structures by information retrieval and update) and another responsible for the mapping of that hierarchical structure with the linear phonological representations. Being semantic in nature, the hierarchical structure does not bottom out at the level of lexical items (words or morphemes), but goes deeper, to the level of semantic primitives.

On the other hand, the actual representations of constituency of linguistic expressions remain pretty much the same (i.e. based on maximally triples of a primitive predicate and two arguments with a strictly binary branching, aimed to capture the same empirical observations about constituency). The set of functional features, or of cartographic regularities, also remain in the present framework as prominent targets at the empirical level, and theoretically also tools, in the modelling of grammatical regularities.

Yet, the nature and place of these structures and features in the system is essentially different. The hierarchical structures recovered by empirical analysis of linguistic strings are seen as stages in the mapping procedure, at one end of which there are the hierarchical structures of the retrieved and/or updated information, i.e. meanings. Functional features are taken to represent regular semantic content or categories of this content, which is special in being exploited by the process of retrieval and update and by the mapping procedure, due to some properties that it has (in particular universality). In the present view, hence, the structures indicated by the constituency of linguistic expressions are not the ultimate goal of investigation, but one of the stages, and one of the cues for the reconstruction of the original hierarchical structures retrieved from the knowledge representation and the path to it coming from the mapping algorithm.

Moreover, in the present approach there are no syntactic operations which affect the interpretation or the phonological realization. Operations in the space between the meaning –

¹⁰ It is also worth considering a non-classical representation of such properties, i.e. maximality instead of universal quantification, addition instead of conjunction etc.

i.e. the subsegment resulting from the retrieval and update of the knowledge representation – and the sound, may only serve the mapping. Under the identical pragmatic conditions – the mapping is always deterministic: it cannot add or change anything. For a given phonological representation in a given pragmatic environment – the entire sequence of mapping operations is uniquely determined in the grammar, and so is the semantic outcome. For a given outcome of retrieval and update in a given pragmatic environment – the entire sequence of mapping operations is uniquely determined in the grammar, and so is the phonological outcome. One quantifier or operator sits higher in the structure than another because that is the hierarchical structure corresponding to the expression interpreted (or that among several possible ones which is favored by pragmatic considerations). The word order in the expression is the way it is not because syntax capriciously decided to move an expression that contains an operator, but because that is simply the word order that under the respective pragmatic environment maps with the respective semantic hierarchical structure.

How is the semantic theory affected by this proposal?

Although this issue has not been given much consideration in the paper, the proposal in fact makes a bigger departure from the main stream formal semantic theory than from the traditional formal theory of syntax. Two aspects are crucial: 1) it is built on a conceptual, rather than truth-conditional base: the meaning of natural language sentences corresponds not to how the world is or should be for them to hold, but only to cognitive representations, i.e. information states¹¹ and 2) that communication does not proceed by updating the knowledge representation of the hearer, but by updating the speaker's representation of the shared knowledge (although this latter aspect is not essential for the proposed view – it is in principle compatible with other views too). The proposed view is similar to Pietroski (2018), and in spirit also to Jackendoff (1983), i.e. to Zwarts and Verkuyl (1994). It abandons truth conditions as the semantic substance, and replaces them with conceptual representations – with a strong role (though undiscussed in this paper) of the pragmatic component of comparison between the speaker's representations of her own and of the shared information state and to the communicative intentions.

How is the focus of research in grammar affected by this proposal?

The redesign of the model of grammar architecture proposed shifts the focus of linguistic interest to the precise properties and a formal model of the retrieval and update procedure and of the mapping between meaning and sound, and to the potential explanations that their properties offer for the empirically observed phenomena in language. Plausibility of the proposed view is also strongly conditioned by its ability to handle the logical semantic phenomena that appear to target the meta-level of knowledge representation such as quantified expressions or predicate-modification. This further requires detailed investigation of the different components and interfaces of the model, such as the lexicon and lexical realization, or the interaction between the logical reasoning and information retrieval and update.

Is there such thing as a narrow language faculty? How much of language is innate?

In the mainstream generative grammar, the core of the narrow language faculty has been argued to lie in the binary set-building operation merge, applying to lexical elements and to the sets it

¹¹ Cognitive representations themselves may be truth-conditionally modeled, but their web-like structure emerging from the algorithm that operates them is crucial for the generation of hierarchical structures empirically attested in language.

produces (Hauser, Chomsky and Fitch 2002). Effectively, this makes the hierarchical constituency structures of linguistic expressions a specific property of language. In the present view, hierarchical structures are not produced by a set-forming operation, but rather by one that forms and retrieves knowledge representations. Since knowledge representation is neither linguistic in nature, nor only accessible by language, this makes the hierarchical structure a general cognitive phenomenon. Specific for language is the algorithm that maps between the hierarchical retrieval and update and the linear structures of phonology. Part of this specific component is the lexicon – which plays a crucial role in the mapping. Even the lexicon is not necessarily specific for language – as it probably uses general cognitive resources (plausibly again a web-like knowledge representation) and only operates a specific type of entities (conventional memorized mappings between segments of knowledge representation and phonological strings). Such a niche in the overall knowledge-representation could have developed under the pressure of the mapping algorithm and language use – just like many other niches of the knowledge representation plausibly emerged under particular functional pressures (e.g. cognitive spatial maps, see Arsenijević 2008). The strong hypothesis is thus that the only language-specific innate capacity is the capacity to map between hierarchical and linear structures. In this paper, I examined the possibility that it crucially involves an embedding procedure which recursively nests properties in one another following their order of retrieval. It is even conceivable that other components, such as the lexicon, develop ontogenetically from general cognitive resources in the process of language acquisition, under the pressure of the two algorithms and without any innate cognitive bias. It is more plausible, though, that such an innate basis has switched over time from the ontogenetic to the phylogenetic level.

Does the proposed view introduce a functionalist turn?

I would rather consider it a turn room for formally plausible combinations of formal and functional explanations. It is still the case both that handles for formal explanations are available (from the formal properties of the two algorithms), and that the formal explanations come before the functional. Nevertheless, certain phenomena are more likely to have a functional explanation (see Haspelmath 2008 for a discussion) – or sometimes the line between them is not clear. Such is the case regarding the universal properties and the role they play in retrieval and mapping procedures.

10. Conclusion

I proposed a fundamental change in the language architecture within the generative methodological enterprise. Instead of a syntactic module which maps between sound and meaning by generating hierarchical structures which are interpreted into these two types of information, I proposed two distinct components. One of them is the procedure of information retrieval and update, which builds, maintains and retrieves a web-shaped knowledge representation and delivers hierarchically structured packages of information to the narrow linguistic faculty. This component is domain-general, and it is responsible for the generation of the hierarchical structures of constituents in linguistic expressions, but also for the generation of the structures encoding our knowledge in other domains of cognition. The hierarchical structure in language is thus semantic in nature. The other component maps between the semantic hierarchical structures produced by the former and the linear structures of the sound-component of language. The mapping component is specific for the linguistic capacity. I provided relatively

rough illustrations of the structure and properties of a model of knowledge representation, the outcomes of the retrieval and update procedure and of the mapping algorithm. I showed how a particular embedding of minimal units of meaning in one another, within the domain of representation of a single entity, economizes with the amount of information and enables fully recoverable linear transcoding of the hierarchical structure. It results in the type of structure recognized in linguistic expressions in generative syntax (especially in the cartographic approaches). I provided a brief and superficial discussion of some of the general questions opened by the proposal.

References

- Arsenijević, B. 2008. From spatial cognition to language. *Biolinguistics* 2/2008, 3-23.
- Arsenijević, B. 2017. Gender, like classifiers, specifies the type of partition: Evidence from Serbo-Croatian. *CLS* 52, 21-37.
- Arsenijević, B. 2020a. Situation-relatives. To appear in U. Junghanns, A. Blümel, J. Gajić and H. Pitsch (eds.), *Advances in formal Slavic linguistics 2019*. Berlin: Language Science press.
- Arsenijević, B. 2020b. Adjectives as a lexical category – a story of striving for extension. To appear in M. Mitrović and P. Panagiotidis (Eds.), *A°: The Lexical Status of Adjectives*. Amsterdam: Jon Benjamins.
- Arsenijević, B. and W. Hinzen. 2012. On the absence of X-within-X-recursion in human grammar. *Linguistic Inquiry* 43/3, 423-440.
- Asher, N. 2011. *Lexical meaning in context: a web of words*. Cambridge; New York: Cambridge University Press.
- Baker, M. C. 2001. *The atoms of language: The mind's hidden rules of grammar*. New York: Basic Books
- Barwise, J. and J. Perry. 1983. *Situations and Attitudes*. Cambridge, MA: MIT Press.
- Boër, S. E. 1974. Proper Names as Predicates. *Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition* 27 (6): 389-400.
- Büring, D. 2013. Syntax, information structure, and prosody. In *Cambridge Handbooks in Language and Linguistics*, 860–896. Cambridge: Cambridge University Press, 2013.
- Caha, P. 2009. *The Nanosyntax of Case*. Ph.D. Dissertation, University of Tromsø.
- Chomsky, Noam 1957. *Syntactic Structures*. The Hague: Mouton.
- Chomsky, N. 1981. *Lectures on Government and Binding*. Dordrecht: Foris.
- Chomsky, N. 1995. *The Minimalist Program*. Cambridge, MA: The MIT Press.
- Chomsky, N. 2000. Minimalist inquiries: The framework. In R. Martin, D. Michaels, and J. Uriagereka (Eds), *Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik*, 89-155. Cambridge, MA: MIT Press.
- Chomsky, N. 2001. Derivation by phase. In Michael Kenstowicz (ed.), *Ken Hale: A Life in Language*, 1-52. Cambridge, MA: MIT Press.
- Chomsky, N. 2005. Three factors in language design. *Linguistic Inquiry* 36. 1–22.
- Cinque, G. 1999. *Adverbs and Functional Heads*. New York: Oxford University Press.

- Cinque, G. and Rizzi, L., 2008. The cartography of syntactic structures. In V. Moscati (ed.) *CISCL Working Papers on Language and Cognition*, 2, 43-59.
- Cremers, C., M. Hijzelendoorn and H. Reckman. 2014. *Meaning versus Grammar: an inquiry into the computation of meaning and the incompleteness of grammar*. Leiden: Leiden University Press.
- Dowty, D. 1979. *Word Meaning and Montague Grammar: The Semantics of Verbs and Times in Generative Semantics and in Montague's PTQ*. Dordrecht: Reidel.
- Fara Graff, D. 2015. Names Are Predicates. *Philosophical Review* 124 (1): 59-117.
- Fodor, J. 1970. Three Reasons for not Deriving "KILL" from "Cause to Die". *Linguistic Inquiry* 1.4
- Greenberg, J. H. 1963. Some universals of grammar with particular reference to the order of meaningful elements. in: *Universals of language*. Joseph H. Greenberg (ed.), 73– 113. Cambridge: MIT Press.
- Hale, K. and S. J. Keyser. 1993. On argument structure and the lexical expression of syntactic relations. In Ken Hale and Jay Keyser (eds.) *The View from Building 20: A Festschrift for Sylvain Bromberger*, 53-108. MIT Press.
- Halle, M. and A. Marantz. 1993. Distributed Morphology and the pieces of inflection. In Ken Hale and Samuel Jay Keyser (eds.), *The view from Building 20: Essays in Linguistics in honour of Sylvain Bromberger*, 111-176. Cambridge, MA: MIT Press.
- Harley, H. and R. Noyer. 1999. Distributed morphology. In *GLOT International State-of-the-Article book*, 3-9. Berlin: Mouton de Gruyter.
- Haspelmath, Martin. 2008. Parametric versus functional explanations of syntactic universals. In Theresa Biberauer (ed.), *The limits of syntactic variation*. Amsterdam: Benjamins.
- Hauser, M D.; N. Chomsky; and W. T. Fitch. 2002. The faculty of language: What is it, who has it, and how did it evolve? *Science* 298.1569–79.
- Hinzen, W. 2006. *Mind design and minimal syntax*. Oxford: Oxford University Press.
- Hinzen, W. and M. Sheehan 2013. *The Philosophy of Universal Grammar*. Oxford: Oxford University Press.
- Jackendoff, R. 1973. The base rules for prepositional phrases. In by Stephen R. Anderson, and Paul Kiparsky (eds.) *A festschrift for Morris Hale*, 345-356. New York, NY: Holt, Rinehart and Winston.
- Jackendoff, R. 1983. *Semantics and Cognition*. Cambridge, MA: MIT Press.
- Kaplan, R. and J. Bresnan. 1982 *Lexical-Functional Grammar: A Formal System for Grammatical Representation*. In J. Bresnan (Ed.), *The Mental Representation of Grammatical Relations*. Cambridge MA: MIT Press.
- Katz, J. J. 1970. Interpretive semantics vs. generative semantics. *Foundations of Language* 6, 220-259.
- Kayne R. S. 1994. *The Antisymmetry of Syntax*. Cambridge MA: MIT Press.
- Klamer, M. 1998. *A grammar of Kambara*. Berlin: Mouton de Gruyter.
- Kratochvíl, F. 2007. *A grammar of Abui: a Papuan language of Alor*. Utrecht: LOT.
- Lakoff, G. 1970a. On generative semantics. In D.D. Steinberg and L.A. Jakobovits (eds), *Semantics*, 232-96. Cambridge: Cambridge University Press.
- Lakoff, G. 1970b. *Irregularity in Syntax*. New York: Holt, Rinehart & Winston.
- Martinet, André. 1949. *La double articulation linguistique*. *Travaux du Cercle linguistique de Copenhague* 5, 30-37.

- Moro A. 2000. *Dynamic Asymmetry*. Cambridge MA: MIT Press.
- Pantcheva, M. 2011. *Decomposing Path: The Nanosyntax of Directional Expressions*. Ph.D. Dissertation, University of Tromø.
- Parsons, T. 1990. *Events in the Semantics of English: A Study in Subatomic Semantics*. Current Studies in Linguistics Series 19. Cambridge, MA: MIT Press.
- Pietroski, P. M. 2018. *Conjoining Meanings. Semantics Without Truth Values*. Oxford: Oxford University Press.
- Reinhart, T. 2006. *Interface strategies: Optimal and costly computations*. Cambridge, MA: MIT Press.
- Rizzi, L. 1997. The fine structure of the left periphery. In: L. Haegeman (ed.), *Elements of Grammar*, 281-338. Dordrecht: Kluwer Academic Publishers.
- Roberts, C. 2002. Demonstratives as definites. In: van Deemter, K, Kibble, R (eds), *Information sharing*. Stanford, CA: CSLI, 89–136.
- Starke, M. 2011a. Nanosyntax: A short primer to a new approach to language. *Nordlyd*, 1(36), 1-6.
- Starke, M. 2011b. Towards elegant parameters: Language variation reduces to the size of lexically stored trees. Transcript from a talk at Barcelona Workshop on Linguistic Variation in the Minimalist Framework. Available at <http://ling.auf.net/lingBuzz/001183>.
- Wolter, L. 2006. *That's that: The semantics and pragmatics of demonstrative noun phrases*. Unpublished doctoral thesis, University of California at Santa Cruz, CA, USA.
- Zwarts, J. and H. Verkuyl. 1994. An Algebra of Conceptual Structure; An Investigation into Jackendoff's Conceptual Semantics. *Linguistics and Philosophy* 17 (1), 1-28.