1 Introduction

A remarkable property of human knowledge of language is that it is infinite. You are not a simple recording device, capable only of repeating members of the finite set of sentences you have logged from experience. On the contrary, you can understand and produce novel utterances; in fact, the vast majority of all sentences that are understood and produced by humans across the globe are new. There are so many sentences, indeed an infinite number of new ones, that we all can, and should, use only our own. Although we all share the set of words we use, and it’s perfectly ethical for you to use a thesaurus to find just the right word for the poem you’re writing, the same is not true with sentences. There is no sentence-thesaurus; and taking someone else’s sentence is very often not ethical – it’s plagiarism. The infinity of sentences is the basis for this ethical principle; there are enough sentences to go around for each of us to use only our own. It’s also the starting point of Generative Grammar, and of our discussion of Merge: a driving goal of the modern study of language is to determine and explain this property of discrete infinity.¹

Your knowledge of language is infinite, but your memory is finite. Your knowledge of language therefore can’t be just a list of memorized sentences. A central component of any theory of language, then, involves generating an infinity of sentences with finite resources. From a finite set of atomic elements, lexical items (roughly but not exactly words²) composed of irreducible linguistic features, the syntax must build an infinite array of hierarchically structured expressions interpretable at the ‘meaning’ interface and available for externalization at the ‘form’ (sound/sign³) interface, the so-called basic property of language (Berwick and Chomsky 2016).

¹ There’s a one-word expression, a two-word expression, and so forth indefinitely, but no one-and-a-half word expression; that’s “discrete” or “digital infinity.” Among others, see Huybregts (2019).
² There is a significant difference between the abstract elements in the Lexicon, the “lexical items” that we refer to in the text, and a common sense notion of a word that may actually be spoken or written. We can leave this aside for present purposes.
³ Despite historical prejudice, the sound modality is not critical to what makes language in the sense intended. The sign modality is equally relevant. Core, mind-internal aspects of language are shared across modalities (Lillo-Martin 1991; Emmorey 2002; Sandler and Lillo-Martin 2006, among others), and different language modalities seem to share modality-independent neural hardware; see Petitto (1987, 2005). For example, when sign-language users have damage in Broca’s area (which is responsible for language production), they will show production errors, just as Broca’s aphasias patients who use a spoken language could have a problem with language production. See Hickok and colleagues (1998) and Klima and colleagues (2002).
Generative Syntax

In current work the operation Merge is the primary structure-building device of the syntax. At the most general level, the picture looks like this. There is a lexicon consisting of a finite set of lexical items available for computation:

(1) The Lexicon (storage bin for lexical material)

The Lexicon provides the raw material out of which Merge, the structure-building device, constructs larger objects. The Lexicon and Merge together constitute language in the narrow sense of the term. We assume that lexical items are drawn as needed from the Lexicon and inscriptions of them are available for computation. Thus, a lexical item such as the noun child can be selected, and as many inscriptions of it as might be needed can be available, allowing such sentences as One child slept while a second child played with another child. Similarly, in mathematics there are multiple inscriptions of, say, the numeral three in an equation like 3x + 3y = 3.

The computational process of structure building takes place within a Workspace (WS), which is updated in the course of the derivation of some expression. The WS is the set consisting of the material available for computation at a given derivational stage. Thus, the WS contains inscriptions of lexical items that have been entered into it and any objects constructed by Merge at earlier points:

---

4 Merge was first introduced in Chomsky (1994); see also Chomsky (2004a, 2013). For further discussion see Collins and Stabler (2016), Epstein and colleagues (2014, 2015), Epstein (2022), Collins (2017), among others. Section 7 provides a history of the development of Merge in the generative tradition.

5 An even stronger view, to be reviewed in Section 8, is that Merge is the only operation of the syntax.

6 See Hauser and colleagues (2002).

7 We return in Section 3 to the technical specification of the WS, differentiating the set that is the WS from the sets that are syntactic objects within the WS; the WS is a set (that is a simple way to represent it) but the WS itself is not a syntactic object that is joined by Merge with any other object. See Collins and Stabler (2016); see also Kitahara and Seely (in press) and Marcolli and colleagues (in press) for a more complete formalization closer to the version of Merge described here.
Preparing for an application of Merge

Merge takes as input the WS, which contains computationally accessible material, and it gives a modified WS as output. Informally speaking, Merge operates as follows:

(i) ‘looks inside’ the WS that it is applying to,
(ii) targets material within the WS,
(iii) builds from that targeted-material an object (i.e., it builds a nonatomic structure), which is now
(iv) a new object within the WS, thereby modifying the WS.

To illustrate, suppose the WS consists of the lexical items shown in (4).

(4) WS = [the, see, I, child]

Merge can take this WS as input and target the inscriptions of the lexical items the and child to create the set {the, child}, adding that set to the WS, and yielding (5).

(5) WS’ = [see, I, {the, child}]

---

8 As we’ll see in more detail as we develop the framework, material that enters the WS is generally accessible for computation. However, in certain circumstances, material can be present in the WS but not accessible to Merge – this material is in effect ‘hidden’ from Merge. Generally speaking, an element is accessible unless rendered inaccessible in some way – we’ll trace some of those ways in Section 6, and will consider various complexities associated with the notion ‘accessibility.’

9 The sets constructed by Merge correlate with the traditional construct ‘phrase.’ For ease of exposition, we will sometimes use both terms, set and phrase, designating the same object. We use standard curly brackets to indicate the sets/phrases constructed by Merge; to avoid confusion, we’ll use square brackets, [], to designate the WS, which as pointed out earlier in this section is also a set, but not a syntactic object constructed by Merge.

10 As we will see as we get into additional details, the inscriptions of the and child that were members of the WS do not themselves remain as members of the WS. The objects the and child are, in effect, replaced by the new set {the, child}, a result that will follow from independent third-factor principles, as will be clarified in Section 4.
Merge can then take this modified WS as input, target within it the object it just created, namely \{the, child\}, and join that object with see to produce the predicate phrase see the child,

\begin{equation}
WS'' = [I, \{\text{see, \{the, child\}}\}]
\end{equation}

and so on leading to the final abstract representation of the sentence I see the child.

As we’ll see as we proceed, building our exposition through successive stages of complexity, Merge does not in practice freely target any element in the WS. In fact, there are very general principles, external and internal to language, that constrain just how Merge applies, with far-reaching empirical consequences. Note further that Merge is recursive in that it can target the objects that it creates in subsequent applications. In other words, in principle, the output of one application of Merge can serve as input to another application of Merge.\(^\text{11}\)

Putting aside technical details, to be introduced in Section 6, this general picture provides a solution to the problem of discrete infinity. With a finite number of atoms and a finite number of computational mechanisms (so far, just one, Merge), the system has an infinite output; Merge can build a new syntactic object out of what it has already created. Merge, then, is the central component of language, where language is understood as a computational device generating linguistic objects receiving an interpretation at the meaning interface and a potential externalization at one of the SM interfaces (of sound or sign).

The goal of this contribution to the Elements series is to closely examine Merge, its form, its function, and its central role in current linguistic theory. We explore what it does (and does not do), why it has the form it has, and its development over time. The basic idea behind Merge is quite simple. However, Merge interacts, in intricate ways, with other components including the language’s interfaces, laws of nature, and certain language-specific conditions. Because of this, and because of its fundamental place in the human faculty of language, this Element’s focus on Merge provides insights into the goals and development of Generative Grammar more generally, and its prospects for the future.

To provide an outline of this Element: In Section 2, we review important background information, tracing the biolinguistic perspective on language assumed by Generative Grammar, that is, that language (in the narrow sense focused on here) involves a computational device embedded within an array of

\(^{11}\) In a more general sense, what Merge builds within the WS at a given stage remains accessible for computation at later stages. Thus, if Merge puts X and Y together to form Z, then the new object Z is available for further computation.
human cognitive systems, and interacting with them. We also stress that a central goal from the generative perspective is to explain the properties of language, not merely to describe them. We then consider a number of key modes of explanation that have been pursued in the generative tradition, including recent work in minimalism, that seek to deduce seemingly complex properties of language from a few simple computational operations that interact with general laws of nature.

Having set the stage, we turn in Section 3 to Merge. The language faculty, by virtual conceptual necessity, involves a structure-building device. The goal is to stipulate as little as possible about it, deriving its seemingly complex properties from more general principles. We start with the simplest conception of Merge, that is, Merge as it would be in a ‘vacuum’ removed from other properties of language and from laws of nature – we start with Merge as a simple computational device. We then add in, step by step, different principles which affect Merge, and which thereby shape the form and function of this operation. Being a computational device, Merge conforms to general efficiency principles that all computation is subject to. But Merge is also a component of the human language capacity. Thus, Merge is subject to – and its operation is constrained by – general properties of human cognition, as well as language-specific principles. We trace such principles in Sections 4 and 5. Merge itself is maximally simple. But it interacts with general and language-specific principles in intricate ways, constraining its application; and these interactions conspire together to produce, ideally, just the empirical effects that we find.

Through Sections 2 to 5, we keep the discussion nontechnical: our goal is to present cutting-edge research on the nature of Merge in a fairly accessible way, minimizing formalism where possible. More formal details are presented in Section 6, which gives technical illustrations of the workings of Merge in key empirical domains; Section 6 is chiefly designed for those readers with a formal background in syntax. Many of the principles associated with Merge that are presented here represent very recent developments in the field; thus, we provide the historical context in Section 7, which reviews the development of Merge over time, tracing key historical antecedents, in an effort to provide the broader context for recent developments. We summarize and take up prospects for the future in Section 8. Overall, we hope that this Element’s contribution will offer an introduction to Merge accessible to anyone generally interested in the study of language; but we also hope to provide some of the latest thinking on Merge that will be valuable to those with an extensive background in the field.
2 Background: Goals and Orientation of the Generative Enterprise

Merge is the central structure-building operation of language. To fully appreciate this, we need to understand how ‘language’ is understood, and trace a number of essential background assumptions, methodological considerations, and research goals.12

2.1 The Object of Inquiry: Language as Biology

Humans have an extraordinary capacity for language. As introduced in the mid-twentieth century, Generative Grammar is focused on this human capacity, seeking to determine its nature, to establish its core properties and, crucially, to explain those properties.

Since its inception,13 the generative enterprise14 has adopted the biolinguistic perspective on language, understanding language as a property of human biology, a cognitive faculty of the human mind. Language is a component of the brain in the same way that components of human vision, emotion, and other cognitive faculties are. Also crucial is the distinction between possession of knowledge of language and the use of that knowledge.15 Consider the old Groucho Marx joke: One day I shot an elephant in my pajamas. How he got into my pajamas I’ll never know. One’s knowledge that there is an ambiguity here is quite different from one’s use of that knowledge to amuse family members at the dinner table. Generative Grammar is concerned with the knowledge state, asking such questions as: What does knowledge of language consist of? What is the best theory of this knowledge? Where did it come from in the species? And, perhaps the most fundamental question of all might be put this way:

How can there be just one human language and multiple languages at the same time?16

12 The discussion in this section is meant to outline background assumptions essential to understanding the nature of Merge in current syntactic theory. For further detail, see, among others, Chomsky (1986, 2000, 2004b, 2017a, 2021b).
13 Early works include Chomsky (1955, 1959, 1965, 1966a, 1968), and Lenneberg (1967), among others.
14 As a historical note, the term ‘generative enterprise’ was first used in The Generative Enterprise (Chomsky 1982a; Chomsky 2004b).
15 For a full account of the origins and break with structural linguistics, see Chomsky (1964). For further discussion, see Chomsky (1966b).
16 For discussion, see Huybregts (2017). Given current genomic evidence, the evolution of Merge (or one Faculty of Language, FL) apparently antedates early human dispersals with subsequent distinct means of externalization (multiple languages). The idea that there is one human Faculty of Language, but multiple means of externalizing the products of that faculty resolves the paradox.
Take any typically developing human baby, place that baby in any linguistic environment, and the baby will, effortlessly for the most part, grow the ambient language – there’s a massive naturalistic experiment on the planet right now that shows this. At a certain level of abstraction, then, there is just one human language faculty and thus one human capacity for language, which is, by definition, part of the innate biological endowment of *Homo sapiens* – your baby is born with it, your puppy is not. Yet, when we look around the world, we see literally thousands of mutually unintelligible languages, which are often characterized as radically different. How is this possible? What must the human language capacity be like for this to occur?

To understand how such questions are addressed, it’s important to note that, from the biolinguistic perspective, language is understood as a computational system, one that builds structured objects from lexical material, where, in current theory, Merge is the structure-building device. This means, among other things, that Merge has the general properties of computational devices, and that language is subject to general principles of computational systems, something taken up in detail in later discussion.

It is important to note too that, in recent work in the generative enterprise, it is assumed that language is closely related to thought; a conception captured simply in William Dwight Whitney’s phrase that language is “audible thought,” a notion that revives a long tradition dating back thousands of years (Chomsky 2022a). Perhaps language is/constitutes thought (see Hinzen 2017, and Chomsky 2022a and Chomsky, in press). On this view, syntactic computation primarily serves the conceptual-intentional (CI) system. Externalization of language, through speech or sign, is secondary; in short, convergent syntactic computations necessarily receive an interpretation at CI but needn’t be externalized (most language use is internal).

2.2 The Quest for Explanation

Humans are born not knowing any particular language, and grow to know one or more ‘individual language(s),’ like French, Ibibio, Russian, and thousands of others, in the course of typical development. Generative Grammar seeks to construct a theory of the cognitive system, the faculty of language, that underlies this process.

At one end of the process, we have the diversity of ‘individual languages.’ An individual language is taken to be a computational system that is a property of and internal to an individual (a biological property of humans). It is ‘intensional’ as well; a function in intension, the actual *grammar*, not its production nor just any grammar generating the same expressions. This notion of language
is labeled I-language, essentially a lexicon and a computational system in the mind of a speaker.

Importantly, I-language is in sharp contrast with E-language, that is, externalization. The central concern of the computational system is the mapping from the Lexicon to the CI interface. I-language relies on structure, and structure only, ignoring linear order; indeed, syntactic rules do not invoke linear order. Externalization linearizes structures in speech/sign. As a matter of principle, the objects constructed by Merge, namely sets, bars linear order from I-language. In contrast, sensorimotor mechanisms cannot see structure and are sensitive to linear order. The two are thus neatly separated and structure dependence is a necessary consequence, explained not stipulated. Structure dependence is nicely illustrated in acquisition. The child learning the language pays attention to what it never hears, namely structure, and not what is right in front of it, namely linear order. As shown by experiment, a thirty-month-old child determines agreement in cases like

(7) the boy and the girl are/*is in the room,

not by appeal to the simplest computational rule, adjacency. Rather, the child reflexively relies on something it never hears: the structure its mind creates. The child then assigns plurality by virtue of the nature of this abstract structure.

This crucial distinction between I-language and externalization is also highlighted by homesign or emerging sign languages. As Huybregts and colleagues (2016) discuss, homesign in deaf isolates or newly emerging sign languages (e.g., ABSL, Negev) in communities with a high incidence of congenital hearing loss amply demonstrate the profound distinction between possession of a language capacity and the use of that capacity. What’s “invented” is not I-language itself (which develops naturally in the individual as determined by human genomics) but rather different ways of externalizing these in different communities.

As further illustration of this important distinction, Chomsky (2012, emphasis added) states:

As discussed in Marr (1982), complex biological systems must be understood at different levels of analysis (computational, algorithmic, implementational). Here we discuss internal language, a system of knowledge, which we understand at a computational level. Since such a system is intensional, therefore not a process, there’s no algorithm. In contrast, externalization, a process of using the internal system, may find an algorithmic characterization.

17 For recent discussion see, among others, Chomsky (2021b).
18 According to some experimental work, down to eighteen months (Shi et al. 2020).
Overall, the evidence is overwhelming that linear order is irrelevant to propositional structure and its interpretation.

I-languages represent the faculty of language in its mature state, attained after interaction with the environment (e.g., taking in linguistic input). Babies are born with the faculty of language in its initial state. A central goal is to understand how the innate computational system gives rise to I-languages. Crucially, what is the balance between the contribution of the language faculty (what the baby brings to the language-acquisition task), the contribution of the input, and the contribution of relevant general laws of nature? These conspire to explain how a child goes from having a general human language capacity to having a particular language, like Japanese.

It’s clear from these central goals that the generative enterprise is concerned with explanation, not mere description. Though the resulting empirical findings are crucial to the enterprise, it is not enough to describe the properties of some language, using whatever unconstrained mechanisms might be available to get the job done. The goal is not to use any type of mechanism that can cover the data. Generative Grammar, from the outset, been concerned with the explanation of the properties of human language – not with just what the properties are, but why those properties take the shape that they do, why they are this way and not another, and why they might exist in the first place. Ideally, analysis of the data contributes to a (conceptually plausible and empirically motivated) account of the central question traced above: How can there be one language and multiple languages at the same time?

Explanation is difficult. How do we know when we’ve explained something? One aspect of explanation involves simplicity. Historically, what we find throughout the development of the generative enterprise is a reduction of the inventory of theoretical postulates within the syntax. What was language-particular (e.g., the rules of French), construction-specific (e.g., the rules of relative clause formation), and syntax-specific (as opposed to a more general rule or principle, not unique to syntax) in earlier stages of the framework was reduced, or factored down, or eliminated, distilling out more general principles – standard practice in science generally. The effects of these postulates were (in large part) derived from the interaction of the syntax (which will be characterized in more detail as we proceed) with the systems that it necessarily interfaces

19 For important discussion, dealing with certain confusions regarding the notion ‘linguistic input,’ see Epstein (2016).
20 The theory of this initial state is often labeled Universal Grammar, UG, adapting a traditional notion to a new context.
21 This fundamental point is missed in the state-of-the-art language models (e.g., GPT3, see, for example https://garrymarcus.substack.com/p/noam-chomsky-and-gpt-3). For important discussion see Chomsky (2022b). See also Chomsky and Moro (2022).
with, systems of sound/sign (sensorimotor) and thought (conceptual intentional). And the effects of these language- and construction-specific postulates were derived from general principles of computational systems; notions of computational efficiency, such as ‘least effort,’ ideally laws of nature.  

A good starting point for explanation, then, is the recognition that three factors enter into the growth of language in the individual (as briefly alluded to earlier in this section; see Chomsky 2005). First is the innate biological endowment, the human faculty of language. Second is experience, interaction with the environment. Third are laws of nature not specific to language, including considerations of computational efficiency, something natural for a computational system like language:

(8) First factor: genetic endowment  
Second factor: experience  
Third factor: laws of nature

From the outset, the goals regarding the empirical content of the innate computational system, the first factor, were in conflict. On the one hand, descriptive adequacy (i.e., getting the facts right) seemed to require that the innate endowment be rich in available mechanisms, initially including multiple and rather complex subcomponents for structure building and structure manipulation (see Section 7 for further details in a historical context). It seemed that the innate system had to be quite complex if the facts of language were to be accounted for – even a superficial look shows that language is complex, diverse, mutable. On the other hand, given the apparently recent evolution of language in the species, the innate computational system must be simple – a complex, multifaceted system could not have evolved in so short a time. Explanatory adequacy (i.e., accounting for the acquisition of language) required uniformity, simplicity, and an account of the ease and rapidity of language acquisition. Furthermore, the quest for simplicity is a defining feature of theory building. As Einstein notes: “The grand aim of all science is to cover the greatest possible number of empirical facts by logical deduction from the smallest possible number of hypotheses or axioms” (Einstein 1954, p. 282).

The attempt to deal with these conflicting demands, that the innate computational system account for complex facts and yet be simple, characterizes much of the history of the generative enterprise. How can the system be made as simple as possible while at the same time maintaining descriptive adequacy?

---

22 See Section 7 for more detailed discussion of the history of the development of Merge and the changes that have occurred in the components of the syntax.

23 As noted earlier, language is a cognitive faculty, understood as a computational device.