

The biolinguistic search for labels

A review and formalisation of labelling within
Minimalist syntax

Louis Edward Chander Van Steene

Magdalene College

University of Cambridge

1st June 2022



This thesis is submitted for the degree of Master of Philosophy.

Declarations

Authorship. This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text.

Statement of length. This thesis is 29,833 words in length. Excluded from this count are the contents of the title page and preface, the epigraph, mathematical and logical formulae, glosses and translations, and the list of references.

Copyright. This work is Copyright © Louis Van Steene 2022. It may be distributed under the Creative Commons Attribution 4.0 International Licence (<https://creativecommons.org/licenses/by/4.0/>).

Software. X_YL^AT_EX was used to typeset this thesis. The T_EX source code can be found at <https://github.com/louis-vs/mphil-thesis-cam-2022>.

Abstract

In any theoretical discipline, it is important to pay careful attention to the precise nature of the formal primitives that are employed in explanations of natural phenomena. The case is no different for language, on the particular conception that a language is a state of knowledge of an individual (I-language), and that the faculty of language is a biological organ that must be understood as a part of the natural world. This biological connection crucially forms the basis of the field of biolinguistics. Under particular investigation here is the hierarchical structure-generating engine of I-language, what is termed (generative) syntax.

This thesis has three broad objectives. The first is to clarify and justify the approach to language outlined above, and to summarise and evaluate theoretical devices that are standardly accepted within the Minimalist Programme, a particular framework of I-linguistic investigation. The second and third objectives concern ‘labelling’, the hypothesised process by which the objects constructed by syntax are given names. It is established that almost every aspect of the purpose and operation of labelling is up for debate. The second object is thus to clarify the central issues that emerge in the labelling literature, first grounding the discussion in terms of the historical development of syntactic theory, then proceeding to evaluate more recent proposals. The final task is to (partially) formalise a particular theory of I-language, modifying and extending the framework constructed by Collins and Stabler (2016). This process reveals the fundamental fragility of many of the critical concepts underlying Minimalist syntax. Suggestions towards improving this situation and extending the empirical coverage of the theory are also presented.

Acknowledgements

For supervising this thesis I would like to thank Ian Roberts and Theresa Biberauer, both of whom have been great inspirations during my time at Cambridge. I would particularly like to thank Ian for our helpful discussions in supervisions and seminars which led me down the rabbit hole of labelling, and to thank Theresa for encouraging me to apply for the MPhil course in the first instance, and for her tireless work as my Director of Studies during my undergraduate years. On that note, I would also like to extend thanks to all of my undergraduate supervisors, in particular Bert Vaux for his encouragement to delve into theoretical issues. Chris Collins gave some very useful advice and references near the start of this project, for which I am also grateful. Any remaining errors or omissions in this work are of course my own.

This work was made possible by an ESRC DTP studentship co-funded by the Harding Distinguished Postgraduate Scholars Programme Leverage Scheme, which I was incredibly lucky to receive. I would like to extend my gratitude to the Cambridge ESRC and to Theresa who aided me in my application.

I would like to thank my fellow Cambridge linguists for their stimulating discussions over the years, in particular Liam and Adam. The Undergraduate Linguistics Association of Britain has made up a not insignificant part of my life and I especially extend my thanks to those committee members with whom I have worked for the past two years for their unending support and dedication, and to those who presented at the conference in Edinburgh this April for encouraging me to stay engaged with the diversity of the field of linguistics. A warm thank you to my friends who have made my time at Cambridge so memorable, and to my family for their continued support. Finally, thank you Lena: your boundless love and energy has made the final few months of writing this thesis all the more enjoyable.

List of abbreviations

C&S	Collins and Stabler (2016)
BCC	Borer-Chomsky Conjecture
BFS	Breadth-first search
BPS	Bare Phrase Structure
CFC	Core Functional Category
C_{HL}	Computational procedure for human language
C-I	Conceptual-Intentional interface
CO	Computational Operation
DFS	Depth-first search
DM	Distributed Morphology
EM	External Merge
EST	Extended Standard Theory
FC	Form Copy
FI	Full Interpretation
FL	Faculty of Language
GB	Government-Binding
GDA	Generalised Dynamic Antisymmetry
GP	The Goldfish Property
IC	Interface Condition
IL	Interface Level
IM	Internal Merge
LA	(in Section 1.4) Lexical Array; (elsewhere) Labelling Algorithm
LF	Logical Form
LI	Lexical Item
LI_k	Lexical Item Token
MaxTP	Maximise Throughput

MG	Minimalist Grammar
MinCCD	Minimise Caching of Completed Derivation
MinCID	Minimise Caching of Incomplete Derivations
MinCUI	Minimise Caching of Unintroduced Items
MinRedup	Minimise Reduplication
MinRO	Minimise Redundant Operations
MinSC	Minimise Space Complexity
MinSearch	Minimise Search
MinTC	Minimise Time Complexity
MinVO	Minimise Vacuous Operations
MM	Methodological minimalism
MP	The Minimalist Programme
MS	Minimal Search
MY	Minimal Yield
NS	Narrow Syntax
NTC	No Tampering Condition
OM	Ontological minimalism
PF	Phonetic Form
Φ	Phonological component
φ -features	Person, number and gender features
PIC	Phase Impenetrability Condition
PLD	Primary Linguistic Data
PS-rule	Phrase structure rule
SA	Search Algorithm
SCM	Successive-Cyclic Movement
SD	Search Domain
Σ	Semantic component; (in Section 4) Search
SM	Sensorimotor interface
SMT	Strong Minimalist Thesis
SO	Syntactic Object

ST	(in Section 2) Standard Theory, i.e. Chomsky (1965); (in Section 4) Search Target
θ -role	Thematic role
TLTB	The Less The Better (substantive optimality)
UG	Universal Grammar
WS	Workspace

Table of contents

Declarations	i
Abstract	ii
Acknowledgements	iii
List of abbreviations	iv
Table of contents	vii
1 Introduction	1
1.1 Biolinguistics and the Galilean challenge	2
1.2 Formalisation and mathematical linguistics	11
1.3 Computational and substantive optimality	13
1.4 Informal theoretical summary	16
1.4.1 C_{HL}	17
1.4.2 The interfaces	18
1.4.3 Features and the lexicon	19
1.4.4 <i>AGREE</i>	20
1.4.5 Phases and cyclicity	20
1.5 Interfaces, variation and variability	22
1.6 Summary and outline	24
2 History of labelling	25
2.1 Early generative grammar	25
2.2 EST and levels of representation	26
2.3 X-bar theory and ‘projection’	29
2.4 Bare Phrase Structure	33
2.5 Merge and labels	36
2.6 ‘Label-free’ (narrow) syntax	37
2.7 Summary	39
3 Review of recent approaches	41
3.1 Key questions	41

3.2	Finding the label	42
3.3	Form of labels	46
3.4	Timing of labelling	47
3.5	Labelling interactions	48
3.6	Summary	50
4	Formalisation	51
4.1	Preliminaries	52
4.2	Copies and repetitions	54
4.2.1	Distinguishing copies and repetitions	54
4.2.2	Copies and multidominance	56
4.2.3	Defining occurrences	59
4.3	Merge, workspaces and derivations	59
4.4	Phases and cyclic transfer	63
4.5	Minimal search	65
4.5.1	Accessibility and trace-invisibility	66
4.5.2	DFS or BFS?	66
4.5.3	Domain and target	68
4.5.4	Defining minimal search	69
4.6	Features	70
4.7	Defining <i>Label</i>	72
4.7.1	Top-down labelling	73
4.7.2	Bottom-up labelling	74
4.7.3	Comparison	75
4.8	<i>Agree</i>	75
4.9	Key conclusions	75
4.9.1	Computational and substantive optimality	76
4.9.2	<i>Agree</i> and <i>Label</i>	76
4.9.3	$Label \in UG?$	76
5	Conclusion and future prospects	77
	References	78

1 Introduction

[Sagredo:] It always seems to me extreme rashness on the part of some when they want to make human abilities the measure of what nature can do. On the contrary, there is not a single effect in nature, even the least that exists, such that the most ingenious theorists can arrive at a complete understanding of it. This vain presumption of understanding everything can have no other basis than never understanding anything. For anyone who had experienced just once the perfect understanding of one single thing, and had truly tasted how knowledge is accomplished, would recognize that of the infinity of other truths he understands nothing.

— Galileo, *Dialogue concerning the two chief world systems*
(1632/1967, p. 101)

The quotidian use of language often concerns itself with the correct naming of things, be these actual objects in the world or, perhaps more accurately, objects constructed by our minds through (and in spite of) our interactions with the world. On a meta level, this same nominal curiosity extends to those abstract objects which we make use of in formulating these descriptions, the invisible structure which allows sound and sign to be associated with meaning in infinitely complex ways. *A priori*, there is no reason to associate particular configurations within a structure with particular *labels*, adopting the typical terminology. Rather, the conceptual necessity of labels, to the extent they are required, must be derived from observations of the natural world to which language belongs, which is no trivial task.

This thesis concerns the nature of labels within the formalised realm of natural language syntax. It will present one possible set of answers to the crucial questions: what is the nature of a syntactic label, how is it assigned, and how is it then made use of in derivations? This project is manifestly ambitious, and hence these introductory sections serve to restrict the scope of the questions being pondered in ways that are extensive, yet I believe principled. Firstly, the notion of ‘language’ itself needs to be established and scrutinised, along with a clarification of the precise object of study: language as a part of the natural world. With this in place, guiding principles for the construction of a formal model of this natural phenomenon can be assembled, manifesting in the form of constraints and heuristics that suggest themselves from general observations and truisms. These can then be made specific in the presentation of an outline of the theory that will be adopted and adapted in the subsequent sections. This introduction therefore sets the stage for a precise formalisation of syntactic labels and the processes they are involved

in, subsumed within the wider metatheoretical landscape. The focus is thus placed narrowly on theoretical and metatheoretical concerns, in a specific sense to be defined—empirical discussion of the consequences of these proposals is left for future work.

1.1 **Biolinguistics and the Galilean challenge**

A major motivation of linguistics as construed in the present context is to rise to what Chomsky (2017, i.a.) formulates as the *Galilean challenge*, citing a passage from Galileo’s infamous *Dialogo*:

“[Sagredo:] But surpassing all stupendous inventions, what sublimity of mind was his who dreamed of finding means to communicate his deepest thoughts to any other person, though distant by mighty intervals of place and time! Of talking with those who are in India; of speaking to those who are not yet born and will not be born for a thousand or ten thousand years; and with what facility, by the different arrangements of twenty characters upon a page!” (Galilei, 1632/1967, p. 105)

Construed narrowly, this passage refers to the alphabet, truly a human “invention”, as opposed to endowment. Nevertheless, it would not be unfair to extrapolate from this passage, as Chomsky (2017) does, a general wonder at the generative property of language, how finite knowledge can have infinite range. In the terms of the later thinker, Willhelm von Humboldt, language is characterised by the “infinite use of finite means”. This oft-quoted aphorism is insightful within the philosophical context of modern linguistics, as explored in great detail by Chomsky (1966), but it is also notable for its conflation of language knowledge and use. This Aristotelian distinction was revived in the modern generative tradition by Chomsky (1965), after its occlusion within the tenets of behaviourism, in which the concept of ‘knowledge of language’, indeed symbolic knowledge of any kind, is effectively unformulable (cf. Chomsky, 1959; Gallistel & King, 2010). As outlined by Chomsky (1986b, p. 3), this “shift of focus [] from behaviour or the products of behaviour to states of the mind/brain that enter into behaviour” provides the grounds for a rich research programme—that of generative grammar. Consequently, language *use* will not be considered much further in this thesis, beyond the necessary empirical selection of instances of language use required to give insight into the knowledge that underlies such use. This follows the general assumptions of the generative programme that date back to Chomsky (1981) and earlier: “the grammar—a certain system of knowledge—is only indirectly related to presented experience, the relation being mediated by UG [Universal Grammar, to be defined

sub—LVS]” (Chomsky, 1981, p. 4). The concern is thus with linguistic *competence*, not *performance* (adapting a distinction made by Saussure, Chomsky, 1964, p. 10).

Some further clarification of the notion of ‘knowledge of language’, the subject matter of this thesis, is thus in order. The term ‘language’, can be and thus far has been used in a non-technical sense, an “informal rubric” that allows one to “select certain aspects of the world as a focus of inquiry” (Chomsky, 1995a, p. 1). Beyond these introductory remarks, it will be avoided in favour of more specific terms. The object of study of this thesis is language as an “element[] of the natural world, to be studied by ordinary methods of empirical inquiry” (Chomsky, 1995a, p. 1). Furthermore, the approach to language taken is an internalist one (Chomsky, 2003): language as a property *internal* to the mind/brain of an *individual*, and which is *intensional*, in that it specifies a “procedure that generates infinitely many expressions” (Chomsky, 2003, p. 263), a characterisation made plausible with the advent of the theory of computation in the 20th century, enabling the infinite to be compressed into the finite (Turing, 1936). The concern is thus with *generation*, not production, enabling a certain precision in description abandoned by 20th century structuralist-behaviourist-empiricists, as noted by Chomsky (2021b) in recent remarks.

Call this naturalistic, materialist, internalist approach to language *I-linguistics* (Chomsky, 2003, p. 263), which concerns itself with the faculty of language (FL), an organ of the mind/brain dedicated to language, and the states it assumes—call these *I-languages* (Chomsky, 1986b, p. 21). I-language is thus a biological entity, necessitating study following the same principles as any other matter of biology. This forms the central tenet of *biolinguistics*, a term coined by Massimo Piattelli-Palmarini (Morin & Piattelli-Palmarini, 1974, see Di Sciullo & Boeckx, 2011, p. 1) identifying a domain of study influentially formulated by Lenneberg (1967, p. vii), namely the study of “language as a natural phenomenon—an aspect of [man’s] biological nature, to be studied in the same manner as, for instance, his anatomy”.

I-language is characterised by what has been called the *Basic Property*, introduced by Berwick and Chomsky (2016, p. 1) and Chomsky (2016, p. 4). The property is concisely stated as such: “each language provides an unbounded array of hierarchically structured expressions that receive interpretations at two interfaces, sensorimotor for externalization and conceptual-intentional for mental processes” (Chomsky, 2016, p. 4). The Basic Property is arguably irreducible, a revival of Aristotle’s dictum that language is “sound with meaning”, placing considerably more emphasis on the exact nature of the “with”. The renewed focus on the interfaces and

the conditions they impose has been a hallmark of the contemporary iteration of the generative enterprise, the Minimalist Programme (MP) as set forth by Chomsky (1993).¹ This focus on the interfaces later became enshrined in the *Strong Minimalist Thesis* (SMT), as in (1), now central to Minimalist research.

- (1) **Strong Minimalist Thesis (SMT):** I-language is an optimal solution to interface conditions. (Chomsky, 2001, p. 1)

As Chomsky (2001) notes, the SMT only becomes an empirical thesis once the notions of ‘optimality’ and of ‘interface conditions’ are defined. These ideas will be explored in Section 1.3 and Section 1.5, respectively. For now, it suffices to state that this thesis is a biolinguistic one, insofar as it concerns I-language and its place within the mind/brain.

As set out by Mobbs (2015), the *argument of linguistic Minimalism* is actually “a collection of related, but logically independent, proposals [that have] coalesced in the literature of the past twenty-five odd years” (Mobbs, 2015, p. 1). The five proposals identified by Mobbs are summarised in (2).

- (2) **The Minimalist Proposals** (Mobbs, 2015)

- (a) **Methodological minimalism:** When faced with two empirically equivalent proposals, “we should adopt the more parsimonious explanation, that is, the one containing fewer ancillary claims” (Mobbs, 2015, p. 38).
- (b) **Ontological minimalism:** assume the SMT, “a refusal on the part of the scientist to make pre-theoretic assumptions about the design or ‘purpose’ of language” (Mobbs, 2015, p. 41).
- (c) **SMT and evo-devo:** “certain facts about language cannot clearly be explained in terms of optimality, and we are forced to propose further innate competence” (Mobbs, 2015, p. 46); this innate competence should be constrained by evolutionary-developmental concerns.

¹A stylistic note: when referring to the principles of and the arguments for the MP, the word *Minimalism* and its morphological family will be capitalised. While we’re here, it is also worth noting that single quotes ‘’ always indicate scare quotes, and double quotes “” are uniformly used for quotations, with the natural caveat that quote styles are preserved within the bodies of quotations. Generally, APA 7th Edition (American Psychological Association, 2020) is adopted with some idiosyncracies, linguistic and personal.

- (d) **The primacy of the CI interface:** “language is optimized for the system of thought, with mode of externalization secondary” (Berwick & Chomsky, 2011, p. 32).
- (e) **Variation:** “[p]arameterization and diversity [are] mostly—possibly entirely—restricted to externalization” (Berwick & Chomsky, 2011, p. 37).

The first three proposals collectively comprise the MP, the fourth and fifth proposals are “specific claims about the design and origin of FL” (Mobbs, 2015, p. 2). They stand logically separate, and need not all be assumed within a Minimalist work. For instance, much work in the comparative syntax tradition does not make assumption (2 d); rather, the parameters along which different I-languages may vary, encoded as formal features, remain a focus of investigation (Biberauer et al., 2010; Roberts, 2019; Sheehan et al., 2017, see).

The first proposal, methodological minimalism, is the least controversial, but perhaps also most important. It is effectively a reformulation of *Occam’s Razor*, the principle that “[w]e may assume the superiority *ceteris paribus* of the demonstration which derives from fewer postulates or hypotheses” (Aristotle, *Posterior Analytics*, cited by A. Baker, 2016; emphasis original). Appropriately, Galileo also adopted the principle: “it is said that Nature does not multiply things unnecessarily; that she makes use of the easiest and simplest means for producing her effects; that she does nothing in vain, and the like” (Galilei, 1632/1967, p. 397). Simplification must come with justification, however. Chomsky (1981, p. 13) notes this explicitly: “it is evident that a reduction in the variety of systems in one part of the grammar is no contribution to these ends if it is matched or exceeded by proliferation elsewhere” ... “[i]t is only when a reduction in one component is not matched or exceeded elsewhere that we have reason to believe that a better approximation to the actual structure of mentally-represented grammar is achieved”. In light of methodological minimalism, Chomsky’s objection here goes both ways: one must be perpetually wary of the power of the explanatory devices introduced within a theoretical framework, in other words, to avoid “the temptation to offer a purported explanation for some phenomenon on the basis of assumptions that are of roughly the order of complexity of what is to be explained” (Chomsky, 1995b, p. 233). As a methodological guideline, this serves of vital importance, especially in the discussion to follow in Section 2 and Section 3, and in the formalisation itself in Section 4. Different construals of labelling have vastly different implications in terms of complexity (theoretical and computational) and need to be considered within the broader framework

of the Minimalist theories they are a part of. Methodological minimalism also serves as a reminder of the consequences of introducing richer theoretical devices, by encouraging a holistic approach, in which the overall complexity of the theory is under constant scrutiny.

The Minimalist principles, now fully established, are thus very useful in the study of biolinguistics. To demonstrate but one example, take again the first Minimalist proposal (2 a). A perpetual problem of biolinguistics which is perhaps most clearly expressed by Embick and Poeppel (2015) and Poeppel and Embick (2005) concerns the interrelation of abstract theories of I-language with the neurobiological models that are supposed to implement the algorithms proposed in theories such as that of this thesis. Known as the ‘granularity problem’, the question arises as to whether it is possible to reduce the often complex and theoretically rich proposals within the linguistics literature—and arguably cognitive science more generally—to the simple constructs of neurobiology. There are a number of ideas adopted within the biolinguistic literature to make this chasm of separation less ominous. Firstly, the adoption of methodological minimalism encourages the reduction of theoretical complexity, ultimately making it more likely that such a model could bridge the explanatory gap. Another set of proposals which will provide some useful framing to the present work are the three computational levels for information processing systems proposed by Marr (1982, p. 25), adapted from Marr and Poggio (1976), as shown in (3).

- (3) (a) **Computational level:** What does the computation do, and what are its goals?
- (b) **Algorithmic/representational level:** How are the inputs and outputs of the computation represented, and what is the algorithm that computes said outputs?
- (c) **Physical/implementational level:** How are the representations and algorithms encoded physically?

Though he was working on human vision, Marr’s motivation for proposing these levels of analysis is analogous to the reasoning employed by Chomsky (1964, 1965) in devising the metric of *explanatory adequacy* to evaluate a linguistic theory, later adapted into MP as the principle of *genuine explanation* (Chomsky, 1993, 1995b). Specifically, Marr (1982, p. 15) notes that “neurophysiology and psychophysics have as their business to *describe* the behaviour of cells or of subjects but not to *explain* such behaviour”. Indeed, finding anything at all interesting from observational or descriptive work (again, generalising the definitions of these terms given by

Chomsky, 1964) is if anything surprising: “[i]f one probes around in a conventional electronic computer and records the behaviour of single elements within it, one is unlikely to be able to discern what a given element is doing” (Marr, 1982, p. 14). It should thus be surprising that work along these lines in neurobiology has indeed had a lot of success. In vision, as in linguistics (in the form of the descriptive success of the structuralists, and of early investigation in neuro-linguistics), some sense of progress had thus been made along such lines, but these results fall short of being explanatory when this is defined in a principled sense. In sum, “understanding computers [equivalently, human brains—LVS] is different from understanding computations” (Marr, 1982, p. 5). This is where proper consideration of Marr’s levels of analysis becomes particularly revealing, offering a positive way out, in which study of neurobiology is not the be-all-and-end-all, but rather an analysis of one aspect of a computational system. Investigation of the computational and algorithmic levels is arguably of equal importance, both to constrain analysis of the more fine-grained phenomena and also, more generally, to gain a holistic understanding of the system, and on a meta level to constrain our conception of what we can even begin to learn about the system in the first place.

Beyond the introduction, the implementational level will receive no further attention. It is hoped, however, that by clarifying the computational level and providing at least tentative steps towards an algorithmic representation, the explanatory gap between linguistics and neuroscience can be reduced, as should be a major objective of biolinguistics, which emphasises the building of interdisciplinary bridges (Boeckx, 2013; Di Sciullo & Boeckx, 2011; Watumull, 2013). It could be argued that the biolinguistics programme has been circumspect since its inception in its historical focus on the computational level. As Marr and Poggio (1976, p. 2) state, “although the top [computational—LVS] level is the most neglected, it is also the most important[, because] the structure of the computations that underly perception depend more upon the computational *problems* that have to be solved than on the particular hardware in which their solutions are implemented” (emphasis original). This view of computational neuroscience is revisited by Gallistel and King (2010), who repeatedly emphasise that understanding general computational theory, as well as the specific instantiations of computations that must be taking place in order for an organism to do certain things at all, should indeed preclude analysis of the neurobiological mechanisms that could plausibly implement such computations. This ‘top-down’ approach thus equates to a stronger thesis than Marr’s. Another important aspect of the

research programme advanced by Gallistel and King (2010) is that, as a logical consequence of adopting a computational model, the *symbols* making up the mental representations that are manipulated by computational procedures become of vital significance. Meanwhile, the implementational details of such representations is not inherently necessary to understanding the computation, as the three level model would indeed suggest. As Gallistel (2001) outlines concisely: “[w]hat matters in representations is form, not substance”. Further, it is these mental representations which should be formalised as part of the algorithmic level, following the definition provided in (3 b).

This focus on the computational level in search of reducing the explanatory gap further relates to a point raised by Chomsky (1994b, p. 187): it is often the case in the history of science that the “more ‘fundamental’ science has had to be revised” in order that the different levels of analysis be unified. One example given is of the dawn of quantum physics by the 1930s, which allowed theories of chemistry that failed to fit in with the classical model to be explained. In the case of language, the current state of knowledge in neuroscience is very far from being able to give any real explanation as to how I-language is implemented (cf. Gallistel & King, 2010). An understanding of the computations, and in following the algorithms, can be sought without overreliance on the implementation. Thus, I adopt a more tempered stance on the biolinguistic programme than Martins and Boeckx (2016), who effectively insist that implementational details are required for work to be characterised as biolinguistic.

Another useful heuristic that applies to the investigation of any biological system is the ‘triple helix’ model as proposed by Lewontin (2000). An organism and its components cannot solely be explained through genetics, and the same is evidently true for I-language. Rather, one must, alongside the gene, consider the development of the organism itself, and the pressures and constraints of the environment. Analogously, Chomsky (2005) adapts this notion to I-language, proposing the *three factor model*. The first factor is the *genetic endowment*, assumed to be effectively uniform for all humans, and which allows us to interpret part of our environment as linguistic and construct grammars. The theory of this factor can be termed *Universal Grammar* (UG). The ‘domain-specificity’ of UG has come under considerable scrutiny since its inception (e.g. Tomasello, 2003), but in reality the point is somewhat moot. The first factor must *a priori* play a role, since there must be something that enables a human child to acquire language, uniquely within the animal kingdom. This is a weak hypothesis, but it need not be

much stronger, when approaching I-language from the perspective of the Basic Property, or in earlier terms “from bottom up” (Chomsky, 2007, p. 4). Secondly, factor two is the environmental factor, the *Primary Linguistic Data* (PLD; Chomsky, 1981, p. 10). The interaction and tension between the first and second factors forms the fundamental basis of the argument from the poverty of the stimulus (Chomsky, 1980, 1986b) and also the well-known tension between descriptive and explanatory theoretical adequacy (Chomsky, 1964, 1965). More specifically, the primitives provided by the genetic endowment should be sufficient to map the child’s intake to the PLD, in other words to interpret the data available to the child into linguistic terms—the notion of *epistemological priority* Chomsky (1981, p. 10). As such, the PLD constitutes *intake* as opposed to *input*, following the terminology used by Evers and van Kampen (2008). Finally, there is the third factor, effectively a catch-all term encompassing the external, non-language-specific constraints imposed by nature. As stated by Chomsky (2005, p. 6), the most important third factors to consider in the case of I-language are “principles of structural architecture and developmental constraints”, in particular “principles of efficient computation, which would be expected to be of particular significance for computational systems such as language”.

As an aside, it is worth noting that this model was made explicit at least as early as Chomsky (2004, p. 105) and, as noted by Chomsky (2006, p. xi), the presence of third factors has been recognised since the genesis of biolinguistics, whilst the development of MP merely made the scale of such concerns more apparent. The exact position is stated already by Chomsky (1965, p. 59): “there is surely no reason today for taking seriously a position that attributes a complex human achievement entirely to months (or at most years) of experience [=2nd factor—LVS], rather than to millions of years of evolution [=1st factor—LVS], or to principles of neural organization that may be more deeply grounded in physical law [=3rd factor—LVS]”. MP and the three factor model are thus a natural step to take in search of the answers to the Galilean challenge.

What is however less often appreciated, as noted by Boeckx (2014), is that the critical insight of the triple helix model as originally conceived by Lewontin (2000) is in the *interaction* between the factors. It is thus arguably incoherent to talk about ‘factors’ in isolation, for instance, by conceiving of some or another proposed linguistic module as ‘belonging to the third factor’, as has become somewhat commonplace (see Gallego, 2011). Indeed, due to the maximally general definition of ‘third factors’, this is arguably not a characterisation that carries much theoretical

content. As such, an understanding of the interaction between the factors will be explicitly pursued here. Another example of such an approach is offered by Roberts (2019) in the context of understanding parameters in a Minimalist context, the hypothesis being that the emergence of parameters in the acquisition process can be explained by utilising a combination of all three factors.

These issues are of particular relevance within the discussion of labelling. As will become clear as a central theme of Section 2 and Section 3, the somewhat precarious position of syntactic labels since the advent of Bare Phrase Structure (Chomsky, 1994a) has led to a sizeable push towards their relegation into being a third factor constraint: “The simplest assumption is that LA [the labelling algorithm—LVS] is just minimal search, presumably appropriating a third factor principle” (Chomsky, 2013, p. 43). The following two subsections will seek to clarify further exactly what “simplest” should mean in this context. Furthermore, Boeckx’s aforementioned objection will be adopted in what follows, emphasising the points of *interaction* between factors. In the context of labelling, these will predominantly be the first and third factors: how does “minimal search” (MS) interact with (constrain, enable) the computational provisions of the genetic endowment in deriving linguistic structures?

A final useful heuristic to keep in mind is the three-way distinction between *metatheory*, *theory* and *analytic* work, as first deconstructed by Chametzky (1996). Metatheoretical work could be considered as much a philosophical endeavour as a linguistic one: it defines the evaluation of theories, as well as how they ought to be constructed, and hence is the focus of this introduction. Theoretical work, on the other hand, is the “deployment of metatheoretical concepts” (Chametzky, 1996, p. xviii)—the construction of primitives and theorems derived from them. This is finally complemented by analytic work, which involves direct investigation of the phenomena in question. By applying theories to data, they can be tested and refined. Analytic enquiry is, without doubt, the most important, and rightly most time-consuming area of linguistics—as Chametzky (1996, p. xviii) states, “[f]or linguistics to be the science of language, this must be where linguists do their work”. This being said, for analytic work to be productive it needs a sufficiently well-defined formal framework. Furthermore, it must fall in line with the metatheoretical aims of linguistic enquiry, in order to ensure that the data being studied actually have an evidential relation with the theory being tested. Following the introduction, this thesis is firmly theoretical, operating at the computational level of analysis in Section 2 and Section 3,

progressing closer to the algorithmic level in Section 4.

To summarise the crucial metatheoretical concerns, and following Chomsky (2021a, p. 7), it is clear that UG must meet three contradictory conditions:

- (4) (a) **Learnability:** UG (in conjunction with third factor principles) must be rich enough to enable language acquisition, overcoming the poverty of the stimulus.
- (b) **Evolvability:** UG must be simple enough to have plausibly emerged under the conditions of human evolution.
- (c) **Universality:** UG, as the name implies, is universal to all humans.

(4 a) and (4 b) evidently contradict each other: the theory must be both rich enough that a child has the resources to acquire the language(s) of their environment, but refined enough such that it could have evolved in a relatively short space of time (cf. Berwick & Chomsky, 2011, 2016, for more precise discussion of the evolutionary context, and Balari & Lorenzo, 2012 for a different approach). Satisfying learnability and evolvability is the “austere requirement” that constitutes the bare minimum for “genuine explanation” (Chomsky, 2020b, p. 14). (4 c) is another way of stating the *Uniformity Principle* of Chomsky (2001, p. 2): “assume languages to be uniform, with variety restricted to easily detectable properties of utterances”. It can also be restated as the typological problem: why do languages appear to vary so much on the surface, and how is this variation constrained? This is effectively the problem solved with Chomsky’s Principles and Parameters model (cf. Chomsky & Lasnik, 1993/2015, for an overview), but this model needs to be adapted from its early, Government-Binding form (Chomsky, 1981) into something more obviously compatible with Minimalism. Some notes on this, with particular reference to how labelling may play a role, are provided in Section 1.5.

1.2 Formalisation and mathematical linguistics

To adopt the concise phrasing of Chris Collins (p.c.), formal work is needed in the domain of labelling in particular, “since otherwise we don’t really understand what we are doing”. It is not, though, a mischaracterisation to say that formalisation has been a concern of the generative enterprise since its inception. One need only look at one of the foundational documents of the enterprise to find the view expressed that there is an intra-theoretical benefit of formalisation: “a formalised theory may automatically provide solutions for many problems other than those

for which it was explicitly designed”, avoiding reliance upon “[o]bscure and intuition-bound notions” (Chomsky, 1957, p. 5). This being said, there remains only a very small component of the literature that focuses on formalisation. Using the terminology established in Section 1.1, much theoretical work instead centres on computational concerns (3 a) as opposed to algorithmic details (3 b), and Chomsky’s own output often drifts into the metatheoretical, which is wholly on the computational level, dealing as it does with more abstract ontological concerns. Attempts to provide a complete summary of the theory are typically pedagogical (for example Adger, 2003; Hornstein et al., 2005; Radford, 2004; Sportiche et al., 2014), and tend to struggle to extract a completely coherent theory from the Minimalist literature (Asudeh & Toivonen, 2006). Collins and Stabler (2016) provide the partial formalisation of Minimalist syntax which forms the bedrock of the present work. Nevertheless, and by their own admission, their formalisation is incomplete, covering only a small subset of the full range of operations typically assumed in the analytical literature. Furthermore, some of its proposals will necessarily require revision in order to be in line with recent developments, notably those of Chomsky (2019a, 2021a), as will be discussed in Section 4.

When it comes to formalising the theory of syntax there are effectively two approaches that can be taken. The first is typified by the field of *mathematical linguistics*, perhaps more accurately characterised a subfield of mathematics rather than linguistics, which seeks to uncover properties of the formal languages originally developed out of linguistic theory. Models within mathematical linguistics may depend to varying extents on empirical considerations, and the focus is rather on the formal and computational characteristics of the grammars being studied. The approach dates back to the earliest work in generative grammar (Chomsky, 1956; Chomsky & Miller, 1963; Miller & Chomsky, 1963), and was extended into the Minimalist era by Stabler (1997), who formalised part of the new, feature-driven, derivational theory presented by Chomsky (1995b). Stabler (1997) established the formal Minimalist Grammar MG, which was subject to considerable further investigation (e.g. Graf, 2013). MGs typically adopt numerous conventions which depart from standard Minimalist assumptions, including, but not limited to, being ‘label-free’, encoding linear order, and being necessarily endocentric. Such mathematical work will not receive much more consideration here.

The second approach is the one taken by Collins and Stabler (2016), and is the one adopted in the present work. Unlike MGs, which “were simplified to facilitate computational assess-

ment”, this approach sets out to “give a precise, formal account of certain fundamental notions in [M]inimalist syntax” (Collins & Stabler, 2016, p. 43). The goal, as with the present work, is thus “to be useful as a toolkit for [M]inimalist syntacticians” (Collins & Stabler, 2016, p. 43), thus contrasting with the purely mathematical approach. As such, the goal is to abide as closely as possible to elements of Minimalist theory as they are actually used, to the extent that this is possible. This forms part of the justification for the review of the labelling literature presented in Section 3, similarly the brief review of the current state of Minimalist theory generally in Section 1.4.

This thesis does not qualify as a work of mathematical linguistics per se, which does not *a priori* have to relate to the study of I-language in and of itself, perhaps instead wavering into the domain of formal language theory, a purely mathematical endeavour, albeit one that may have language-related applications, such as within natural language processing, alongside the study of parsing (on this latter point, see Mobbs, 2008, 2015, as well as contributions to Berwick & Stabler, 2019). The goal of formalisation within the context of biolinguistics should not be to “play[] mathematical games” but to “describe[] reality” (Chomsky, 1975, p. 81). Nevertheless, the close association maintained in this introduction between mathematical formalism and the biolinguistic programme may lend the present work to a classification as *mathematical biolinguistics* in the sense of Watumull (2012, 2013), blending a biolinguistic ontology with the mathematical realism of Cohen (2008) and Tegmark (2014). Since the metaphysical baggage that this categorisation would beget would take us too far afield, I leave the matter aside, although it receives some discussion in Van Steene (2021, Section 2). The crucial point here is the justification of formal investigation of the properties of natural language on biolinguistic grounds, in accordance with the resolution of the granularity problem and in the search of the simplest model that accords with the empirical facts, in line with the Galilean challenge.

1.3 Computational and substantive optimality

I-language is a computational procedure, in the sense pioneered by Turing and his contemporaries (Turing, 1936). It is therefore natural to analyse it in terms of complexity theory, terms increasingly apparent with the dawn of MP.

MP as defined by the first three proposals in (2) allows specific hypotheses and heuristics regarding computational complexity to be made. Mobbs (2015) provides a discursive synthesis

of these, culminating in the taxonomy summarised in (5) and adopted within this thesis.

(5) **A Minimalist framework for computational optimality** (Mobbs, 2015)

- *Maximise Throughput (MaxTP)*
 - *Minimise Time Complexity (MinTC)*
 - * *Minimise Redundant Operations (MinRO)*
 - *Minimise Vacuous Operations (MinVO)*
 - *No Tampering Condition (NTC)*
 - *Minimise Search (MinSearch)*
 - * *Minimise Reduplication (MinRedup)*
 - *Minimise Space Complexity (MinSC)*
 - * *Minimise Caching of Unintroduced Items (MinCUI)*
 - * *Minimise Caching of Incomplete Derivation (MinCID)*
 - * *Minimise Caching of Completed Derivation (MinCCD)*

The principle of MaxTP, coupled with the computational orientation of the theory, enables the use of ideas from computational complexity theory. The most relevant of these from the perspective of linguistic theory are effectively distilled in (5). Notably, NTC and MinSearch will receive particular attention and more precise definitions, especially within the formal section of the thesis, Section 4.

As discussed by Mobbs (2015), alongside computational optimality there is a notion of *substantive* optimality, which “can be thought of as the number of different types of symbol and computational operation (CO) the FL employs” (Mobbs, 2015, f.n. 57). In effect, substantive optimality dictates that we, as theorists, reduce the number of ‘tools’ we introduce into our theories, especially those that cannot be justified by independent means. Mobbs (2015) compresses the concept of substantive optimality into the principle of *The Less The Better* (TLTB): “[t]his principle merely observes that a proliferation of types of symbols, constraints on symbols, types of CO, and constraints on the output of COs (taken to be) used in the FL runs contrary to M[ethodological]M[inimalism], O[ntological]M[inimalism] as methodology, and the evo-devo hypothesis for language” (Mobbs, 2015, p. 61). TLTB is thus shorthand for the three Minimalist proposals which comprise MP.

One may protest that ‘TLTB’ is a hopelessly vague principle, too general to have any beneficial effect. As demonstrated in the following discussion, however, TLTB can be invoked to efficiently justify theoretical choices, without referencing individual Minimalist proposals from (2). Notwithstanding this, Mobbs (2015, f.n. 99) notes the complications that would arise from even attempting to formulate a narrower definition of substantive optimality. The emergence of more symbols and operations conceivably entails an evolutionary burden, and there will be a cognitive cost associated with a larger computational inventory. TLTB allows “some ‘principle of structural architecture’ [to] constrain the instantiation of new computational machinery – although we have little idea of its character, in accordance with the obscurity of neuronal implementation” (Mobbs, 2015, f.n. 99). As a more streamlined instantiation of Occam’s Razor, TLTB serves a useful Minimalist purpose. Indeed, specific principles posed within Minimalism are reduced to TLTB, such as the *Inclusiveness Condition* (6) and *Full Interpretation* (7).

(6) ***Inclusiveness Condition***

“[N]o new objects are added in the course of computation apart from rearrangements of lexical properties” (Chomsky, 1995b, p. 228)

(7) ***Principle of Full Interpretation (FI)***

“Features can only appear in a derivation if they are already interpretable to the interfaces, or can be properly licensed for deletion [footnote omitted—LVS] before the interfaces.” (Mobbs, 2015, p. 62, cf. Chomsky, 1986a, pp. 95–101 and Chomsky, 1995b, pp. 151, 219–220)²

Note that the combination of (6) and (7) entail that all properties within syntax are ultimately lexical. As will be discussed in Section 1.4 and Section 2, this is a crucial result that emerges from the Minimalist architecture.

In sum, answering the Galilean challenge in a principled, scientific manner, entails adopting substantive optimality in the form of TLTB, alongside the principles of computational optimality as in (5).

²Interface interpretability is covered in Section 1.4.

1.4 Informal theoretical summary

This subsection introduces the main theoretical concepts that have been accepted as standard within MP and that form the basis of the discussion to follow. The theory is ultimately founded upon empirical results that will not receive any discussion here. The goal is instead to summarise and synthesise Chomsky’s seminal Minimalist papers (Chomsky, 1993, 1994a, 1995b, 2000, 2001, 2004, 2007, 2008, 2013, 2014, 2015, 2019b, 2020a, 2021a; Chomsky et al., 2019). Whilst these papers could be said to represent a coherent development of a Minimalist theory over time, they do not, of course, identify a uniform contribution of Chomsky’s but rather a useful summary of how general concerns have evolved in MP. Closer scrutiny of labelling in particular is postponed until Section 2, where the development of labelling theory is reviewed, and Section 3, which discusses the innovations since Chomsky (2013, 2015).

Such a summary is not always provided in work in the domain of theoretical syntax, especially in more analytic work, but it is warranted here on account of the precision that can lack in theoretical syntax, and which this work sets out to begin to rectify. This is not intended as criticism—rather as testament to the nascent nature of the field and the sheer scale of complexity of the phenomena under investigation. As a metacritical aside, it may well be amusing though not picayune to suggest that Chomsky’s own body of work resembles Aristotle’s in a way criticised by Galileo explicitly in the aforementioned work.³ When presented with such a string of references to a single author as above, an oeuvre that supposedly represents a coherent thread of theory alongside the more programmatic suggestions, one might be reminded of a certain passage from the second day of the *Dialogo*. Protesting Salviati’s dismissal of Aristotelian doctrines, Simplicio holds that, in order to be qualified to do so, “one must have a grasp of the whole scheme, and be able to combine this passage with that, collecting together one text here and another very distant from it”—indeed, taking it a step further, he subsequently claims that “[t]here is no doubt that whoever has this skill will be able to draw from his books demonstrations of all that can be known; for every single thing is in them.” (Galilei, 1632/1967, p. 108). Sagredo wittily replies:

“I have a little book, much briefer than Aristotle or Ovid, in which is contained the whole of science, and with very little study one may form from it the most complete

³For similar comments from a different perspective, see Asudeh and Toivonen (2006). For a considerably more disparaging assessment of Chomsky’s recent work, see Behme (2014, 2015). A full response to these criticisms would be far too great a tangent.

ideas. It is the alphabet, and no doubt anyone who can properly join and order this or that vowel and these or those consonants with one another can dig out of it the truest answers to any question[.]” (Galilei, 1632/1967, p. 109)

Thus, Galileo offers an addendum to this fascination with the alphabet and by extension with language which Sagredo proclaims on the first day, and which is so often cited by Chomsky, as discussed in Section 1.1. A goal of this thesis is to eliminate the overreliance on appeal to authority criticised by Galileo, truly approaching I-language “from [the] bottom up” (Chomsky, 2007, p. 4)—again, not mathematical game-playing, but seeking genuine explanation. The review of Minimalist theory provided in this subsection and in Section 2 and Section 3 makes clear how entwining the various skeins of MP is by no means a trivial task, albeit certainly a worthwhile endeavour.

1.4.1 C_{HL}

An I-language L ($[F]$, Lex , $MERGE$, $AGREE$, $TRANSFER$, Φ , Σ) is a state of the faculty of language FL , a component of the human mind/brain.⁴ The initial state S_0 of FL , call this Universal Grammar UG , determines the set of *features* available for all languages F , from which L selects a subset $[F]$, and assembles this subset into a lexicon Lex consisting of lexical items (LIs). For each derivation, L selects a lexical array LA from Lex . UG also determines the computational procedure for human language C_{HL} which generates narrow-syntactic expressions, syntactic objects (SOs), out of LIs. In the definition of L above, C_{HL} consists of the three operations $MERGE$, $TRANSFER$ and $AGREE$. LIs are the ‘atoms of computation’ for C_{HL} . An LI is an SO, termed a *head* or *minimal projection* within the context of a larger SO constructed by C_{HL} . On the simplest assumptions, C_{HL} is uniform for all L .⁵ The operation $MERGE$, part of C_{HL} , recursively constructs objects out of LA , each of which can be mapped to a semantic representation SEM by the *semantic component* Σ and to a phonetic representation $PHON$ by the *phonological component* Φ . The operation $TRANSFER$ hands an object generated in the narrow syntax NS by $MERGE$ to Σ and Φ , resulting in the expression $Exp = \langle PHON, SEM \rangle$.

⁴Operations will be denoted in this subsection using capital letters, following the convention introduced by Chomsky et al. (2019). When discussing operations without any particular theory in mind, CamelCase normal text will be used—this is employed e.g. in the review sections, Section 2 and Section 3. For the formalised operations in Section 4, I adopt a different convention (see footnote 30).

⁵As proposed by Chomsky (2004, p. 107), contra Chomsky (2000, p. 100), where it is suggested that “parameter setting” be “refinement of C_{HL} in one of the possible ways”. The theory presented here assumes that variation is confined to the lexicon, as elaborated further in Section 1.5.

L generates a set of expressions *Exp*, interpreted at the interfaces. *PHON* is interpreted by sensorimotor systems SM, whilst *SEM* is interpreted by conceptual-interpretive systems C-I.

1.4.2 The interfaces

Together, SM and C-I constitute the *interfaces*, crucial to the minimalist approach proposed by Chomsky (1993): “all conditions are interface conditions; and a linguistic expression is the optimal realization of such interface conditions” (Chomsky, 1993, p. 26). As such, the interfaces SM and C-I and their respective mappings Φ and Σ deserve further attention. Σ is assumed to be uniform for all L, Φ is assumed to vary greatly. Indeed, following the Minimalist proposal (2 d), Φ is the locus of all variation (a point that I will return to in Section 1.5). As aforementioned, C_{HL} is uniform, thus linguistic variation is confined to $[F]$, *Lex*, and Φ . Φ also has the special property that it may introduce features from $[F]$ into the computation of *PHON*, violating Inclusiveness (6), which necessarily holds only for C_{HL} . Very little is understood about Σ , which may introduce features not present in SO, but we assume not from $[F]$ (Chomsky, 2004, p. 107). If this does hold, it is sensible to include Σ in the definition NS, as is typical in the literature.

Following Chomsky et al. (2019, p. 241), there is no operation *SpellOut*, which eliminates structure before transfer to SM. Further, note that ‘PF’ and ‘LF’, as internal levels of representation, are undefined, following Chomsky (2004, p. 107). Rather, there is a single, unified cycle, with *TRANSFER* handing over syntactic objects, called *phases*, to Φ and Σ . The specifics of the cycle and the precise nature of phases will be discussed further below, in Section 1.4.5.

An expression *Exp* is said to *converge at an interface level IL* if it is legible at *IL*, in other words if the interface condition *IC* at *IL*, $IC(IL)$, is satisfied. *IC* states that “the information in the expressions generated by L must be accessible to other systems” (Chomsky, 2004, p. 106), which is, evidently, a requirement that language be usable at all—the barest possible metric of “good design”, a key methodological assumption of MP. By contrast, *Exp crashes at IL* if it does not meet $IC(IL)$. By extension, the computation of *Exp converges* if *Exp* converges at both SM and C-I, otherwise it *crashes*. A derivation will only crash if it fails to remove all features from the resulting SO that are *uninterpretable* at the interfaces before *TRANSFER* takes place. A derivation that has removed all such features will always converge, but to varying levels of *deviance* as determined by the interface systems—a suggestion of Chomsky (2004, p. 112),

reinforced by Chomsky et al. (2019, p. 238): “concerns about “overgeneration” in core syntax [i.e. NS—LVS] are unfounded; the only empirical criterion is that the grammar associate each syntactic object generated to a <SEM,PHON> pair in a way that corresponds to the knowledge of the native speaker ... “overgeneration” must be permitted on purely empirical grounds, since “deviant” expressions are systematically used in all kinds of ways”. This point will prove crucial with respect to the discussion of labelling to follow, and will also receive further attention in Section 1.4.5.

1.4.3 Features and the lexicon

Features require further attention: why should uninterpretable features exist at all in a system adhering to principles of “good design”? The *Interpretability Condition*, that “LIs have no features other than those interpreted at the interface, properties of sound and meaning” is “transparently false” (Chomsky, 2000, p. 113). Rather, I-language is characterised by what Biberauer (2019, p. 54) calls “*systematic departures from Saussurean arbitrariness*”—the presence of so-called ‘formal’, grammatical features which play a role in C_{HL} but not directly at the interfaces.

An idea that persists, from its introduction in Chomsky (1995b, p. 277 *et seq.*) is that uninterpretable features exist to capture the displacement property of language—long considered an ‘imperfection’, but accepted by Chomsky (2004, note 29) as, in fact, the most Minimal option. On the original formulation by Chomsky (2000), the fact that both of these then-considered ‘imperfections’, uninterpretable features and displacement, appear to be intimately connected suggests that “the two imperfections might reduce to one” (Chomsky, 2000, p. 121). Furthermore, the optimal conclusion would be that dislocation itself is required by design—either as part of *IC* or as a consequence of the nature of the operations within C_{HL} . The latter is demonstrated to be the case by Chomsky (2004), with the introduction of *internal MERGE* (IM) and *external MERGE* (EM), building upon the unification of syntactic operations begun by Kitahara (1997). $MERGE(X, Y)$ is considered IM if X is contained within Y , else it is considered EM. Displacement, following the ‘copy’ theory of movement, comes for free as a consequence of the nature of *MERGE*, which in its simplest formulation does not bar access to objects that have already been merged.⁶ Further consequences of this will follow.

⁶As will be discussed in Section 4.2, the term ‘copy’ is a bit of a misnomer, hence the scare quotes. It is nevertheless standard to assume some lossely defined form of copy theory in Minimalist work.

1.4.4 *AGREE*

The existence of uninterpretable features forms part of the justification for *AGREE*, the final component of L as stated above. *AGREE* forms a relation between two SOs, one of which is termed a *probe*, the other a *goal*. Standardly, the probe must c-command its goal, although there are other possibilities, which will be considered in the formalisation in Section 4.8. The goal is located via some mechanism of MS, again to be clarified in Section 4.5.

In many older articulations of the theory, *AGREE* is taken as part of the more complex operation Move, which is composed of *MERGE*, *AGREE*, and a third operation, pied-piping, which remains poorly understood but is given much attention, for example in Chomsky (1995b). Following Chomsky (2004), Move will not be taken to be a part of the theory. The suggestion is that all of its empirical import can be taken over with only the more minimal operations of C_{HL} , in combination with IC and third factors. Labelling has much to reveal here, as will be discussed in the following sections. Indeed, whether *AGREE* is needed at all will come under scrutiny. A preliminary motivation for this is that both labelling and *AGREE* are effectively realisations of MS. This strongly implies some kind of redundancy. If this is the case, and *AGREE* can be abandoned, this would lead to a simplification of C_{HL} , a move in line with both methodological (2 a) and ontological minimalism (2 b). This is the approach taken in Chomsky’s most recent work, where *AGREE* receives almost no mention (Chomsky, 2021a). Further discussion is reserved for Section 4.

1.4.5 Phases and cyclicity

In the interaction of the subcomponents of L, a need arises to identify the units that are available to take part in an operation within C_{HL} . Assume therefore that C_{HL} operates within a workspace *WS*, which represents the state of a derivation at any particular point.⁷ Operations are “strictly Markovian” (Chomsky, 2021a, p. 20), beyond even the standard Markovian property of derivations—*WS* does not contain previously generated items, since these are eliminated by *MERGE*, in accordance with a property of computational optimality termed *Minimal Yield* (MY, Chomsky, 2021a, p. 19), equivalently *Restrict Resources* (Chomsky, 2019a). The formal properties of derivations beyond this will be explored in more depth in Section 4.3.

⁷The idea of the workspace within the context of modern minimalism was most notably formalised by Collins and Stabler (2016), and finds further elaboration by Chomsky et al. (2019) and Chomsky (2019a, 2019b, 2021a).

The SOs generated by C_{HL} are assumed to be *bare*, in the sense of Chomsky (1994a). Equivalently, they are formed only by the operation *MERGE*. With the notion of *WS* established, it is possible to define *MERGE* as a function between workspaces⁸ In previous formulations, *MERGE* is typically considered to be a binary operation, which takes two SOs X and Y and combines them to form the set $\{X, Y\}$, itself an SO. For Chomsky (2021a) who borrows much from Collins & Stabler, 2016: *MERGE* operates on a sequence of SOs σ , such that each SO in σ is accessible and that σ exhausts *WS*; *MERGE* is free to take any two objects and merge them together, mapping *WS* to a new workspace WS' . The definition in Section 4 offers a more precise account. An important consequence is that the application of *MERGE* is free, in the sense of Chomsky et al. (2019), meaning that constraints on *MERGE* must fall out from the conjunction of IC, third factors, and other operations such as *AGREE*. Labelling, it will be argued, surely also plays a role. This therefore does not entail that *MERGE* must be ‘triggered’, as assumed in stricter MGs and many other Minimalist theories such as that of Adger (2003).

Finally, it is worth briefly characterising the core functional categories (CFCs) and, in turn, the nature of phases. Following Chomsky (2000), the CFCs are taken to be C, expressing force and mood (and possibly abbreviating a number of categories taken to form the *left periphery*, following Rizzi, 1997), T, expressing tense and event structure, and v^* , the light verb head of transitive constructions, expressing argument structure.⁹ These functional categories are ‘core’ in the sense of being the locus of agreement and dislocation generally. Following Chomsky (2008), CP and v^*P are phases, C and v^* their respective *phase heads*. This is arguably problematic, as T is very obviously involved in Case, ϕ -feature agreement and movement—with the EPP¹⁰ being a classic example. Chomsky (2008) resolves this by making explicit the idea of *inheritance*: “for T, ϕ -features and Tense appear to be derivative, not inherent: basic tense and also tenselike properties (e.g. irrealis) are determined by C (in which they are inherent)” (Chomsky, 2008, p. 143).¹¹ Thus, “Agree and Tense are inherited from C, the phase head” (Chomsky,

⁸*WS* is analogous to a “working memory”—in a computational, not necessarily a cognitive sense, much like the tape of a Turing machine. See Watumull (2012, 2015) for a possible formalisation of the linguistic Turing machine, in which these issues come to light.

⁹Cf. footnote 25 on *v*.

¹⁰Extended Projection Principle, classically formulated as the requirement that [Spec,TP] be filled (cf. Chomsky, 1981). Now, EPP-features are interpreted more generally, as the requirement that a head needs its specifier to be filled, usually by movement of the external argument in the case of T. This is the *generalised* EPP-feature (see Haegeman, 1996; Laenzlinger, 1998; Roberts, 2004). EPP-features may be obviated by labelling and other considerations, as discussed further in Section 3. Eliminating the EPP has been a long-term goal in generative syntax—cf. Bošković (2007b).

¹¹The earliest published mention of inheritance comes from Chomsky (2007), actually inheriting the idea from

2008, pp. 143–144).

Derivations proceed *strictly cyclically*, phase-by-phase. Further, C_{HL} constructs objects in parallel in the workspace, but all operations occur effectively instantaneously at the phase level. As stated by Chomsky (2004, p. 116): “TRANSFER has a “memory” of phase length, meaning [] that operations at the phase level are in effect simultaneous”. This, presumably, makes the apparent countercyclicality of inheritance only apparent. Further, operations apply freely—order does not matter; any deviant or crashing derivations that result are discarded by the interfaces. More conclusions are possible, as reiterated by Chomsky (2008, p. 143): “along with Transfer, all other operations will also apply at the phase level, as determined by the label/probe. That implies that IM should be driven only by phase heads”. Labels clearly play a significant role: the label of the phase is always the probe for *AGREE* (obscured by the fact that the agreement properties of T are inherited from C). The interactions between labelling and agreement are discussed in Section 4.8.

One of the most important consequences of strict cyclicity is the Phase-Impenetrability Condition (PIC) as in (8), from Chomsky (2000, p. 108).

(8) *Phase-Impenetrability Condition*

In phase α with head H, the domain of H is not accessible to operations outside α , only H and its edge are accessible to such operations.

The PIC proves critical in discussions of locality, taking the place of Subjacency (Chomsky, 1973) and Barriers (Chomsky, 1986a) in previous frameworks. The place of labelling within the context of locality and the phase will be a key point of analysis within Section 3.

1.5 Interfaces, variation and variability

It is worth briefly expanding upon Section 1.4.2 in order to provide some more detail on the precise status of the ‘interfaces’ in contemporary Minimalist theory, in particular with respect to features.

The concept of ‘features’ is elaborated in Section 1.4.3. Following the Borer-Chomsky Conjecture (M. C. Baker, 2008), all linguistic variation—equivalently, everything that is learned; the second factor—is restricted to the lexicon, and hence to the arrangement of lexical features

Marc Richards, later published as M. D. Richards (2007), which works off of a manuscript version of Chomsky (2008) distributed even earlier.

within the lexicon, and their treatment by I-language. One could question exactly what these features are—some properties, like interpretability, have already been discussed. One may further question how distinct the difference sets of features (phonological, semantic, and syntactic) are. Standardly, features come in interpretable-uninterpretable (or valued-unvalued) pairs, where interpretability is an interface property, as will be assumed in Section 4.6. Nevertheless, some authors argue for a ‘substance-free’ system, adopting the term introduced by Hale and Reiss (2008): a system is *substance-free* if it involves “computation over abstract mental entities” (Hale & Reiss, 2008, p. 22), in other words being *symbolic* in the sense applied to cognitive science by Gallistel (2001), Gallistel and King (2010) and Pylyshyn (1984), inter alia. By contrast, a theory in which ‘symbols’ are actually embodied in percepts would be considered *substanceful*. For example, a phonological theory in which phonemes directly correspond to aspects of phonetic substance, as in standard generative phonology (Chomsky & Halle, 1968), is substanceful. In the present case, a syntactic theory in which categories ‘leech’ off of semantic properties would be considered substanceful. Zeijlstra (2014) argues for this approach in syntax, although a full discussion of this would take us too far afield.

Classic formulations allow the label of a syntactic object to be a (categorial) feature, a bundle of features, or a lexical item, and in more recent approaches more complex objects are allowed to serve as labels (see Section 3). As such, an understanding of the feature inventory F will be essential to formalising the answer to the question of what can be a label. The question of what (kinds of) features are allowed thus has direct bearing on the topic of this thesis. However, adopting the BCC, as typically done in Minimalist work, entails that features also have a significant impact on the class of humanly computable I-languages.¹² Thus, the question of language variation—in effect, rephrasing (4 c)—is unavoidable, but also inevitable in a feature-based theory. These issues will be explored further in the subsequent discussion.

Another aspect of the theory presented in Section 1.4 is that *MERGE* can apply freely. There

¹²Using the more precise term introduced by Hale and Reiss (2008, p. 3), as opposed to a weaker alternative like ‘possible languages’. Adapting ideas originating in Evolutionary Phonology (Blevins, 2004) to I-language generally, the set of possible I-languages may be irrevocably restricted by historical, cultural, and anthropological factors of a very different nature to the first and third factors considered here to go into a theory of I-language. This is, effectively, a truism that emerges upon consideration of the second factor: the data of the environment have no *a priori* justification to be the way they are except for the fact that they are *a priori* constrained by first and third factors. A different course of history could have led to there being a completely mutually exclusive set of possible languages available to the linguist to study and the child to learn throughout time, but if a child from our world were to travel to this hypothetical world, we still want to say that they could learn the language. Hence the concept of *humanly computable*.

seem to be a number of good reasons, theory-internal but formalisation-external, to adopt the free Merge approach. The approach receives particular justification by Chomsky et al. (2019), and further reasons crop up on occasion in the following discussion. The general point is that ‘overgeneration’, traditionally thought of as the bane of a sound theory, is actually good, if considered in a restricted manner. Namely, it allows ‘deviant’ structures to be generated, which perhaps satisfy constraints at one interface, but are to some variable degree uninterpretable at another. This brings the theory of I-language more in line with general empirical observations, in which grammaticality is a gradient property of expressions (Sprouse et al., 2018).

1.6 Summary and outline

This introduction has served to illuminate the concerns central to the biolinguistic research programme in which context this thesis is situated. It has provided some novel synthesis of the most recent ideas in MP and provided broader comment on the ways in which the Galilean challenge can be tackled.

Labelling itself is to receive more attention in the following sections. Section 2 constitutes a historical review of approaches to labelling, beginning with the earliest work in generative grammar. The conclusions from this section frame the discussion in Section 3, which unravels the central issues within contemporary approaches to labelling. The formalisation itself comes in Section 4, which will extend Collins and Stabler’s (2016) formalisation of Minimalist syntax, including a novel definition of MS and, in turn, the labelling algorithm. The overarching goal is to create a formal model of syntax that is both internally consistent and has the potential to provide genuine explanation of linguistic phenomena. As will become clear in the following sections, a precise understanding of labelling is essential to this.

2 History of labelling

Initially, it must be established precisely *what* labelling is, best demonstrated with a historical account of the position of labels within generative grammar. Whilst this account is abbreviated in places, it provides ample detail to ground the discussion to follow, without any severe misrepresentation of the ideas in question.

In the course of this review, original analysis of historical proposals in a modern context will emerge. This provides opportunity to discuss the proposals in an informal context, shedding light on some of the conceptual issues which the informal approaches described in the literature display. This will leave a number of trails only partially explored, to be picked up in Section 3, which reviews more recent developments.

2.1 Early generative grammar

Early generative grammars made use of *production systems* consisting of sets of rewrite rules, as originally formalised by Post (1944) and first applied to language by Chomsky (1951). The *phrase structure rule* (PS-rule), familiar from Chomsky and Miller (1963), takes a form resembling (9), taken from Chomsky (1965).

$$(9) \quad S \rightarrow NP Aux VP$$

The so-called *Standard Theory* (ST), represented by Chomsky (1965), allows a particular I-language to specify a set of PS-rules that generate the sequence of *base phrase-markers* that are permitted to serve as *deep structures* for sentences that are grammatical within the I-language. In order to derive the surface representations, a series of (syntactic, phonological) *transformations* is applied.

In ST, labels have little explanatory significance, as they are merely a symbolic description of a category. The choice of symbols, from the perspective of the system, is entirely arbitrary: there is nothing preventing alternative symbols being used to represent the exact same symbolic rule. For instance, take the PS-rule (10), which is isomorphic to the rule in (9).¹³

¹³Isomorphic literally means ‘having the same form’. A precise mathematical definition of an isomorphism depends somewhat on the framework a mathematician is working in—e.g. set theory, category theory, geometry, etc. This will not be important here. For discussion of the meaning of isomorphism within the context of symbolic representations in cognitive science, see Gallistel (2001).

(10) $\alpha \rightarrow \beta \gamma \delta$

Similarly, the rule (11 a) is isomorphic to the rule (11 b).

(11) (a) $NP \rightarrow Det N$

(b) $\beta \rightarrow \epsilon \zeta$

As such, PS-rules create an incredibly powerful, recursively enumerable grammar, as proven by Post (1944, 1947). That such a phrase-structure grammar could be part of a genuine explanation for I-language is intuitively problematic, as it readily poses issues with learnability and fails to account for the empirically verifiable limits on syntactic variation. A full review of the insufficiency of phrase structure grammars would be a large diversion, but in simplistic terms “[i]t just gives the wrong results because it doesn’t express the natural relationships or capture the principles” (Chomsky, 2009). PS-rules alone simply do not represent the dependencies that natural language clearly employs. Similarly, they allow relationships between categories to be expressed which have no basis in natural language, such as the nonsensical rule (12).

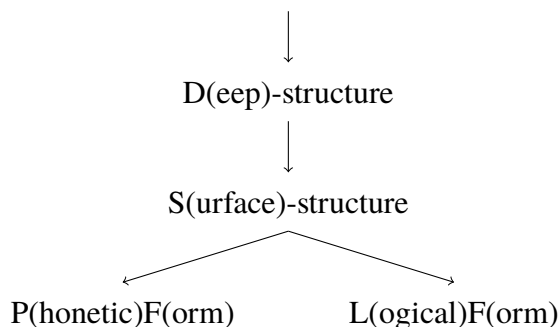
(12) $NP \rightarrow V PP$

In short, a theory of UG based on PS-rules thus cannot meet explanatory adequacy, neither accounting for the limits of variation as per (4 c) nor the facts of language acquisition as per (4 a).

2.2 EST and levels of representation

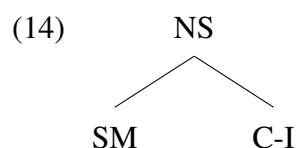
For mostly independent contemporaneous reasons, ST’s deep/surface-structure dichotomy was abandoned in favour of the more articulated *Y-model*, postulated within the Extended Standard Theory (EST) as represented by Chomsky (1973, 1976, 1977) and subsequent work, and diagrammed in (13).

(13)



Each node represents a *level of representation*, while each arrow represents a set of transformational rules, or base rules in the case of generation of deep structure. Further, each level of representation may impose representational *filters*, like the Case Filter (Vergnaud, 1977/2008), which disqualify certain structures from attaining a valid interpretation. This more finely articulated theory affords further resolution to the kinds of transformational rules available, clearly demonstrating a ‘divide-and-conquer’ approach to theorising, the kind which could be said to have influenced the extensive modularisation characteristic of the later development of the EST into the Government-Binding (GB) model summarised by Chomsky (1981) and explored below. Crucially, too, this approach entails that both PF and LF are generated on the basis of S-structure, a radical departure from ST which took deep structures to represent the semantic structure of a sentence.

Though with the caveat that this temporarily breaks from the chronology, it is beneficial to compare the EST’s Y-model with the Minimalist architecture introduced by Chomsky (1993), graphically represented in (14), and that makes up a significant part of the foundation of the theory outlined in more detail earlier, in Section 1.3.



As is visually clear comparing (13) and (14), the more recent model abandons the levels of representation (and of derivation, following the application of transformational rules) which obscure the relation between narrowly syntactic structures and interface representations. Curiously, then, this is in spirit a return to the more direct access to structures enabled by ST, albeit without the added complexity of the transformational component, following the unification of the generation and transformational components enabled with the operation Merge, introduced by Chomsky (1993).¹⁴

Thus far I have described the computational symbols involved in ST’s PS-rules to be entirely arbitrary, as the examples in (10) and (11) demonstrate. This could, however, be regarded as a misrepresentation of Chomsky’s (1965) position. Chomsky (1965) explicitly comments on the issue of the set of symbols which he assumes throughout the work, noting that they are not necessarily ‘substance-free’ (see Section 1.5). He asks “whether the formatives and category sym-

¹⁴Move was at this point still considered a more complex operation than Merge, as detailed in Section 1.3.

bols used in Phrase-markers have some language-independent characterization, or whether they are just convenient mnemonic tags, specific to a particular grammar” (Chomsky, 1965, p. 65), suggesting that “these elements [] are selected from a fixed, universal vocabulary”, though conceding that the question “is generally held to involve extrasyntactic considerations of a sort only dimly perceived” (Chomsky, 1965, p. 66), which I would interpret as being considerations of (semantic) substance—correlating with the phonetic substance proposed as a basis of phonological features (Chomsky & Halle, 1968). This prospect relates to the idea that a ‘semantic spine’ of some sort could have a direct and significant constraining effect on the set of grammatical syntactic outputs, which forms a central hypothesis within a range of theories that could be considered Minimalist to greater or lesser extents. Whilst a full review is out of the question, such approaches are detailed for instance by Adger (2013), Brody (2000), Starke (2004) and Wiltschko (2014). This interesting property of the Minimalist architecture (14) marks a consequence of the reduction of the Y-model which is incredibly important yet not immediately obvious. In the Y-model (13), it is impossible for LF considerations to dictate in any way the operations of the transformational component which generates S-structure, as a result of the modularity entailed by the architecture. The Minimalist architecture, meanwhile, enables regular and rapid interactions between narrow syntax and interpretation—indeed, this is allowed to such an extent that “access can in principle take place at any stage of the derivation” Chomsky (2021a, p. 7).¹⁵ Models that make use of so-called ‘telescoped’ representations, in particular that of Adger (2013), rely on labelling being heavily driven by C-I (see also footnote 18). Whilst only assuming a backseat role in discussions around the (E)ST period, within the theory, labelling is actually held as a deeply ingrained assumption, inimicly tied to the procedure of structure generation. The base rules necessarily directly encode the labels of constituents, which then proceed to have an impact at every subsequent stage of the derivation. One issue that immediately arises from this is that the labels need to be maintained in computational memory for this entire period—once the derivation reaches S-structure, there is no guarantee that the label is predictable from the label-less structure. This massively violates MinSC, as per the framework for computational optimality introduced in (5). This contrasts with the Minimalist architecture, in which labels can be separated from the narrow syntax, precisely because their information

¹⁵With the caveat that this access standardly occurs at the phase level—“[a]ccess at any other stage of the derivation will yield some form of deviance or incoherence” Chomsky (2021a, p. 23). On deviance, cf. Chomsky (2020a) and Chomsky et al. (2019).

is not needed until interpretation. There appears to be a link between these two concepts: the degree of interpretive influence on the state and continuation of the derivation, and the theoretical status of the identifiers assigned to syntactic objects. These observations are apparent (in hindsight) even at this early stage of the development of generative theory, and they prove to provoke probing questions still in the modern era.

Another argument against the EST model arises from this abstraction of the structure-building component from the interfaces. The use of arbitrary symbols as labels severely violates substantive optimality and the SMT, since there is no way of justifying and constraining label selection, inflating the role of the syntax beyond what is necessary. The model can also be shown to introduce vast amounts of computational complexity using the framework in (5). The fact that PS-rules must individually be crafted for every permutation of categories allowed as constituents within a particular context entails massive reduplication of rules, violating MinRedup, and in turn clearly violating MinTC, as this large inventory of rules will need to be searched repeatedly during a derivation. Such search also entails proliferation of choice points in the derivation, necessarily leading to mass caching of incomplete derivations whilst the correct parse is found for a particular expression, hence violating MinCID, and in turn MinSC. Thus, PS-rules fail to maximise throughput (MaxTP), exacerbating the conceptual issues already discussed. In sum, whilst a theory formulated with PS-rules could plausibly meet the goal of descriptive adequacy, on account of their generative power, explanatory adequacy as formulated in terms of the MP remains a remote prospect.

2.3 X-bar theory and ‘projection’

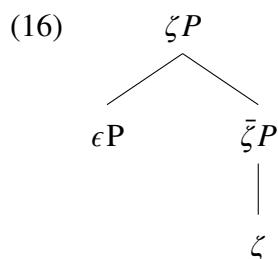
With the development of the EST and the Y-model, however, also came a major refinement of the rule system in the form of the development of X-bar theory (Chomsky, 1970; Jackendoff, 1977), which also marks a major step in the development of labelling.¹⁶ The central notion of X-bar theory is that every phrase is *endocentric*—to wit, every phrase has a *head*. In modern terms, as elaborated in Section 1.4, a head is an LI. It would not be a mischaracterisation, then, to say that a critical aspect of the development of X-bar theory was in the adoption of what in modern

¹⁶The presentation of X-bar theory here naturally glosses over details that were subject to debate at the time, such as the number of ‘bar’-levels that a head is able to project. A model roughly following Chomsky (1981) is assumed, incorporating some suggestions towards unification proposed by Muysken (1982). Chronologically, X-bar somewhat predates the Y-model, instead being created to deal with certain properties of nominalisations (Chomsky, 1970); this and other historiographical concerns extend beyond the scope of this discussion.

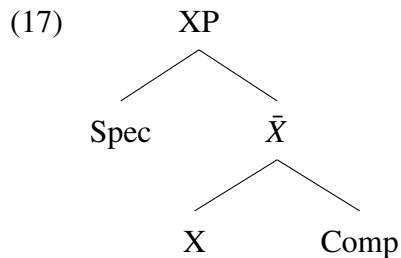
terms would be called a deterministic labelling algorithm—what was in the terminology of the time called *projection*. The head *projects* its properties up through the structure. Returning to our examples, the rule (11 a) is no longer isomorphic to the arbitrary rule (11 b), because the relation between ζ and β (ζ is the head of β) must be represented in the structure. Hence, an endocentric formulation of the rule is as in (15).

$$(15) \quad \zeta P \rightarrow \epsilon P \zeta$$

Introducing endocentricity is not sufficient within the X-bar framework, however. Its second vital contribution was of intermediate ‘bar’ levels: intermediate projections that were effectively *exocentric* when considered maximally locally. Bar-level projections offer a position in the structure for specifiers and adjuncts. The structures entailed by bar-levels are exocentric in terms of immediate dominance, as only the intermediate projection that immediately dominates the head dominates a head at all, since higher intermediate projections alongside the maximal projection dominate only other non-minimal projections. The head is still determined at every point, however, since a bar-level intermediate projection is marked as such (hence the bar), meaning that its category must continue to project up the tree, maintaining this property of projection, as the head is represented at every level within a category. When understood less strictly than in terms of immediate dominance, X-bar structures do thus conform to endocentricity, in the sense that every structure has a head and the head of a given structure can be determined at every point. Meanwhile, the maximal projection in the specifier, complement or adjunct position is barred from projection, being already marked as a maximal projection. As a result, ϵ in (11) is also altered in the endocentric (15), since it is neither a phrase, nor a head of any other phrase. It also needs to be part of an endocentric structure, in order to satisfy the X-bar schema. A X-bar compliant representation of the structure described by the example grammar presented thus far is therefore shown in (16), with the bar-levels of ζP represented, and the internal structure of ϵP abbreviated.



The significant development here is that the schema can be generalised. The major claim of X-bar theory is thus that all phrases satisfy the general schema in (17), the informal rule equivalent given in (18).



- (18) (a) $XP \rightarrow (Spec) \bar{X}$
 (b) $\bar{X} \rightarrow \bar{X} Adj$ [optional, unordered, can be repeated]
 (c) $\bar{X} \rightarrow X (Comp)$

The *specifier* Spec, *complement* Comp, and adjuncts must be maximal projections, viz. they themselves need to be fully qualified XPs. The presence of Spec and Comp and their required properties in particular instantiations of the schema reduce to selection (equivalently, subcategorisation), which is a lexical concern.

In effect, what this system does is greatly reduce the generative capacity of the PS-rule system by constraining the available rules to those presented in (18). It also makes a strong claim of endocentricity, captured explicitly in the *Projection Principle* of Government-Binding theory (GB; Chomsky, 1981, p. x). GB constitutes the next significant development in the theory, incorporating X-bar theory, alongside a highly modular organisation of the grammar accompanied by many improvements on the specifics of the EST model. The Projection Principle is defined as in (19).

(19) *Projection Principle*

[L]exical structure must be represented categorically at every syntactic level. (Chomsky, 1986a, p. 84)

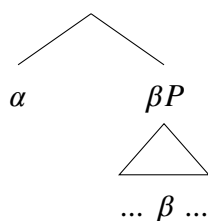
The “categorical” representation is, in modern terms, the *label*. Following the Projection Principle, the syntax is separated from the lexicon: lexical properties are *projected*, allowing their properties to move upwards in the structure, and forwards in the derivation.¹⁷ In part as a result

¹⁷Abstracting somewhat from the representational/derivational issue. See Section 1.3.

of this principle, phrase structure rules can be eliminated entirely—all structure is formed via X-bar projections from lexical items. In modern terms, the labels within the structure allow it to be correctly interpreted at the interfaces, subsuming the GB notions of filters and conditions at the various levels of representation. Importantly, endocentricity comes as a corollary of the Projection Principle and the X-bar schema: there is simply no way of generating a structure that is not ultimately headed/projected by an LI, viz. one that is exocentric. Crucially, this interpretation is made possible by the assumption that the putatively exocentric bar-level projections are invisible in terms of endocentricity. Bar-invisibility holds for other formal relations involved in GB, here glossing over the technical details of the contemporaneous theory. Nevertheless, the principle that syntactic objects can be invisible to future operations (hence being in some sense ‘frozen’ in place) continues to be of relevance in present-day theorising and affords further attention in order to break down the assumptions ingrained within the X-bar framework, an objective which indeed proves to be a theme of subsequent developments, as will become clear notably in Section 3.

Note that the fact that the Projection Principle deduces a deterministic method of identifying labels, allowing categorial information to be preserved throughout the derivation, was not part of the original motivation for the principle within the contemporaneous theoretical context. Instead, the Projection Principle was necessary to deal with issues surrounding subcategorisation. As detailed by Chomsky (1981, pp. 29–34), in ST there is a redundancy between lexically-specified subcategorisation frames and PS-rules such as (11 a), which specifies that, within an NP constructed by this categorial rule, a determiner can have a noun complement. In our terms, this redundancy is unsatisfactory both by methodological minimalism and computationally in terms of MaxTP. Reducing the categorial component to the X-bar schema and forcing lexical information to project upwards through the representation was the proposed solution. Furthermore, the Projection Principle coupled with X-bar theory naturally leads to a further hypothesis on the nature of syntactic structures. The hypothesis is dubbed by Hornstein et al. (2005, p. 178) the *Periscope Property*, and it claims that “there are [] no known cases where a syntactic relation cares about anything, but the head” (Hornstein et al., 2005, p. 178). Generally speaking, in configurations like (20), β may be selected by α no matter the intervening material.

(20)



As per Hornstein (2021, p. 98), the Periscope Property highlights three facts about subcategorisation (21).

- (21) (a) In the configuration (20), *α* can select for *β*.
(b) In (20), *α* cannot select for anything else within *βP*.
(c) The selection relation is *linearly unbounded*.

To illustrate, take a determiner D selecting for an NP. The structure of the NP is irrelevant to the syntactic operation of selection—the contents of any complement, specifiers or adjuncts are ignored, even if the heads of these projections are in some sense ‘closer’ to the selecting head. Rather, the N head of the selected NP is always considered the target for subcategorisation. Violating this principle would be catastrophic for any theory of subcategorisation that otherwise adheres to the model; retaining the effects of the periscope property thus seems to be essential in any theory of syntactic relations that resembles GB in this manner.¹⁸ This is not to say, however, that projection must retain its traditional form. Referring back to the example given, the periscope effect may in fact be derivable from the properties of the search operation which finds the ‘nearest’ head. This hints at the prospect of decoupling labelling from structure building, a much later development in the theory, and one given considerable attention below.

2.4 Bare Phrase Structure

For now, the next step in the development of labelling theory comes with the introduction of the Bare Phrase Structure model (BPS; Chomsky, 1994a). It may initially appear that endocentri-

¹⁸There is a growing body of literature that does in fact reject the Periscope Principle in favour of what Brody (2000) denotes *telescoped* representations. Such a theory as developed by Adger (2013) attributes the purported effects of the periscope illusion to properties of C-I, which is purported to provide a skeletal schema to aid interpretation of telescoped structures. Whilst this work is thus thoroughly Minimalist in character, reducing properties of the grammar to interface conditions, it requires a radically different approach to labelling than is more standardly assumed in the literature, and is thus beyond the (peri)scope of this thesis. One must also keep in mind Chomsky’s warning as mentioned above, that “reduction in one component [not be] matched or exceeded elsewhere” (Chomsky, 1981, p. 13). The reader is referred to Adger (2013) for a complete exposition of a telescoped theory and some consequences.

city can no longer result from a generalisation of the Projection Principle as described above: either the simplest combinatorial operation doesn't project at all, leaving the nature of the object formed unknown, or projection has to be encoded into the operation directly, which appears to be a stipulation. In the original formulation of BPS, Chomsky (1994a, 1995b) predominantly adopts the latter option: the operation 'Merge' produces a labelled object $\{K, \{\alpha, \beta\}\}$, where $K \in \{\alpha, \beta\}$ —with this latter stipulation, the Projection Principle remains intact, now without the baggage of the full X-bar structure. One consequence of this is in setting out a programme by which extraneous phrase structure introduced by the X-bar can be eliminated wherever possible, a challenge notably taken up by Bošković (1997) in his postulation of the *Minimal Structure Principle* (Bošković, 1997, p. 25), an evaluation metric which selects the representation that makes use of the fewest projections.

Another important consequence of BPS is that the notions of 'intermediate' and 'phrasal' projections are eliminated from the representation. With the minimalist stipulation $K \in \{\alpha, \beta\}$, it is simply not possible for these diacritic properties to be stored in the structure. This is a positive development: from the perspective of methodological minimalism, diacritics are highly disfavoured, as they present issues of arbitrariness and complexity along similar lines to the objections against PS-rules discussed above. Put simply, they violate the Inclusiveness Condition (6), a subcondition of TLTB (see Section 1.3). Furthermore, the information previously represented by the bar ($\bar{}$) and phrase (P) diacritics can instead be derived relationally from the structure relatively trivially using the set of definitions in (22), adapted from Chomsky (1995b, pp. 242–243), actually following a suggestion originally from Muysken (1982) in an X-bar context (cf. also Chomsky & Lasnik, 1993/2015).

(22) **Levels of Projection** (Chomsky, 1995b; Muysken, 1982)

- (a) **Maximal Projection** (X^{max} , XP): a category that does not project any further.
- (b) **Intermediate Projection** (\bar{X} , X'): a category that both is a projection and itself projects.
- (c) **Minimal Projection** (X^{min} , X° , X): a category that is not a projection, i.e. a head.

Note that the diacritic symbols used in (22) are not to be interpreted literally—that is, they exist solely for expositional convenience and are not actually present in the computational system. As a result, including the label within the Merge operation appears to enable maximally efficient

computation of the projection level of a particular constituent: minimising space complexity by omitting diacritics, and minimising the search space and in turn time complexity of calculating the projection level. Endocentricity is also maintained; indeed, endocentricity is even more apparent: since diacritically labelled ‘bar-levels’ are completely eliminated from BPS, and adopting the $K \in \{\alpha, \beta\}$ assumption, the head of a phrase is identical to the label of the phrase. In all cases, the head therefore must be the first element identified by search.¹⁹ Similarly, periscoping is maintained as all features of the lexical head are necessarily accessible at every level of projection, as all features of the head travel with the label, because of the condition that K must be identical to (viz. a copy of) the head. A second consequence of BPS, one that is supported by the definitions in (22), is that a category can serve as both a minimal and maximal projection simultaneously, represented $X^{min/max}$. Hence, the full projection that is required by the default X-bar schema, is no longer stipulated in BPS, instead being optional, determined by other properties of a particular derivation. The BPS model is thus much more flexible than that of X-bar, a flexibility that hinges upon the nature of the implicitly computed labels of different categories.

Nevertheless, the earlier concern over the stipulation of $K \in \{\alpha, \beta\}$ warrants further investigation. It is clear from the perspective of TLTB that it would be more optimal to eliminate labels in favour of *Simplest Merge*, $Merge(X, Y) = \{X, Y\}$, assuming that the necessary generalisations, such as the derivability of the X-bar projection levels in (22), were able to be maintained. In fact, precisely this is achieved by Collins (2002), who argues that labels can be eliminated from syntax in favour of Simplest Merge. As admitted, however: “[s]ince virtually every syntactic analysis in the generative tradition makes use of labels on phrasal categories[], the task of eliminating labels from syntactic theory is enormous” (Collins, 2002, p. 44). In spite of this, the postulation of a ‘label-free’ syntax plausibly constitute a productive research programme, with clear empirical consequences.

¹⁹Unfortunately, this generalisation does not actually hold when considering the full exposition of BPS presented by Chomsky (1994a). As pointed out by Seely (2006), Chomsky (1994a) deliberately excludes labels from being *terms* in his definition thereof. Seely (2006) argues that labels must therefore be syntactically inert, incorporating this into an argument for the elimination of labels following Collins (2002). This argument will receive further attention in Section 2.5 below; for now, though, it suffices to state that the heads are apparent at every level of a BPS representation, without further stipulations.

2.5 Merge and labels

The possible options for the (informal) definition of the fundamental combinatorial operation of syntax are thus as listed in (23).

(23) (a) $Merge(X, Y) = \{K, \{\alpha, \beta\}\}$, where $K \in \{\alpha, \beta\}$

(b) $Merge(X, Y) = \{X, Y\}$

Some final notes on option (23 a): it is worth considering why the value of K should be restricted as stipulated. Chomsky (1994a, p. 4) justifies the restriction of the label K by appealing to both IC and TLTB. First, by TLTB (both Inclusiveness and FI) K must be α , β , or their union or intersection. The union can be ruled out: “the union will not only be irrelevant but contradictory if α , β differ in value for some feature”, assumed to be the “normal case” (Chomsky, 1994a, p. 4). On the other hand, the intersection also does not suffice: “the intersection of α , β will generally be irrelevant to output conditions, often null” (Chomsky, 1994a, p. 4), null for the same reason that the union would be contradictory, viz. because features may differ in value. The latter two options would be nonsensical at the interface, hence K must be one of the two mergees. Therefore, projection is maintained, explicitly, encoded into the combinatorial operation. Abandoning labels entirely would obviate the need to consider this issue, but if labels are maintained in any form, it is clearly important to consider how the set of possible labels is restricted.

Chomsky et al. (2019, p. 247) note that it is a nontrivial question as to why the label in (23 a) cannot undergo head movement. Indeed, an additional problem here is that labels are already indistinguishable from copies formed by movement, without further inspection. Whilst structure preservation dictates that movement of a head in such a manner would be prohibited in the case where $K = X = \alpha^\circ$, for α° a head, at higher levels of projection the syntactic object that serves as the label would be indistinguishable from a phrase at the point of labelling, and thus could be considered a case of raising.²⁰ However, assuming the definition of terms introduced by Chomsky (1995b) and highlighted by Seely (2006), this additional problem is avoided (see also footnote 19). For Chomsky (1995b), any structure formed by Merge is a term, and the members of the members of any term are terms. This recursive “members of

²⁰With ‘structure preservation’ here formalised as the Uniformity Condition on Chains by Chomsky (1995b, p. 253), see Roberts (2001, f.n. 1).

members” definition ensures that labels, which are merely members of the set formed by (23 a), can never be terms. However, another consequence of defining terms as such, as Seely (2006) demonstrates, is that labels must be invisible to the syntax anyway, as, not being terms, they cannot participate in syntactic relations. Thus, the postulation of labels within the syntax seems somewhat pointless, notwithstanding their potential significance at the interface. Indeed, as Collins and Seely (2020) argue, the now mainstream adoption of (23 b) suggests that label-free syntax is in some fundamental sense the right approach.

2.6 ‘Label-free’ (narrow) syntax

Returning to the debate encapsulated in (23), we can now consider some consequences of adopting (23 b). As per Collins (2002), an alternative mechanism is needed to determine the next step in a derivation. He terms this device the *locus of derivation*, and it effectively serves as a temporary label stored in working memory which directs the progression of the derivation.²¹ This, in effect, allows the derivation to keep track of its current state, without storing this state as a label within the structure. In turn, this enables a key simplifying consequence, according to Collins (2002, p. 48): “[t]he major difference between a locus and a label is that there is only one locus in a derivation, while there are many labels [since] each constituent has a different label”. Interestingly, however, as noted by Seely (2006), this cannot be the case—rather, “it is arguable that there are as many [l]oci in a derivation as there are lexical categories with unchecked probes/selectors” (Seely, 2006, p. 213). Furthermore, in the case of categories that serve as probes/selectors, the label and the locus are equivalent, as per Collins (2002). Therefore, it is not at all clear that labels have been fully eliminated under the Collins (2002) model. Rather, a middle ground has appeared to have emerged, in which labels are eliminated from Merge as per (23 b) but labels/loci are retained in order to make derivations possible at all.

An alternative middle ground is already staked by Chomsky (2000), asking the question: “Are labels predictable?” (Chomsky, 2000, p. 133), effectively approaching the issue from the other side to the Collins (2002) approach. Firstly, assume $label(\alpha) = \alpha$, where α is an LI (already implicit in Chomsky, 1994a). (Set-)Merge is assumed to be symmetrical, unlike the original formulation above, where the formation of the label is intrinsic to the operation, stipulated with recourse to IC. This being the case, in order to create the asymmetry presumed to be

²¹Using ‘working memory’ in a computational, not neurobiological, sense.

required for a label to be formed, and thus for the structure to be legible, one of α , β must be a *selector*, with which the other has merged in order to satisfy a selectional requirement. The selector is assumed to provide the label in every case; C_{HL} is therefore able to identify the label. On the basis of this, Chomsky (2000, p. 135) claims that “[i]n all cases, then, the label is redundant ... The label is determined and available for operations within C_{HL} or for interpretation at the interface, but is indicated only for convenience”. The exact position is staked out by Chomsky (2004, p. 109): “a label [] is always a head. In the worst case, the label is determined by an explicit rule []. A preferable result is that the label is predictable by general rule. A still more attractive outcome is that [I-language] requires no labels at all”. The latter option would be the ‘label-free’ option advocated by Collins (2002) and Seely (2006). Further, “operations are “driven” by labels” (Chomsky, 2004, p. 109)—if this is the case, it leads to the even stronger conclusion that there can be no general Spec-head relation, as was essential to the analysis in Chomsky (2005) as a consequence of the definition of ‘checking domain’, a hangover of GB’s m-command which allowed Spec-head relations to take place. Similarly, as noted by Blümel (2017), there is also consequently “no upper limit to the number of specifiers, a notion that has no status in the theory” (Blümel, 2017, p. 51), a welcome result, abandoning a stipulation of X-bar theory in line with TLTB and also enabling a number of empirical results to be derived (see N. Richards, 2001).

Chomsky (2008) offers an informal definition of the aforementioned “general rule”—in other words, a *labelling algorithm* (LA)—as restated in (24)

(24) *The Labelling Algorithm of Chomsky (2008, p. 145)*

- (a) In $\{H, \alpha\}$, H an LI, H is the label.
- (b) If α is internally merged to β , forming $\{\alpha, \beta\}$ then the label of β is the label of $\{\alpha, \beta\}$

The IM condition of Chomsky (2000) is thus retained, albeit within a theory adopting Simplest Merge plus LA. This is implicitly problematic: there is in principle no formal difference between the operations EM and IM themselves—they are merely instances of Merge with slightly different inputs—so there is a key stipulation in play. A further problem, noted by Chomsky (2008, f.n. 34), is that the LA (24) is unable to label $\{\alpha, \beta\}$ structures that are formed by *external* Merge. However, this scenario must occur at least twice in a derivation. For one, the very first step of

a derivation must by definition be EM of two LIs. Secondly, EM of the external argument (to vP) necessarily involves EM of two phrases. In addition to these issues, it is apparent that this labelling algorithm, as with its predecessors, needs to take place at the time of Merge. This is required by LA (24) because of (24 b), where the context of Merge determines the label. These issues receive further attention in Section 3 below.

This primitive first attempt at an LA is subsequently developed, notably by Chomsky (2013, 2015), into what Bošković (2016a, p. 4) terms a ‘label-or-not’ system. Within such a theory, labelling is not a part of Merge, as in (23 a), and thus Simplest Merge (23 b) is adopted, as in Chomsky (2004). However, unlike in the label-free system (Collins, 2002; Seely, 2006), it is still required for there to be a way of assigning labels “that license[] SOs so that they can be interpreted at the interfaces” Chomsky (2013, p. 43). Alongside this move, Chomsky (2013) abandons the stipulation that the label may be determined by the operation of IM. There must, then, be a way of assigning labels which enables the correct interpretation to be provided for non-deviant sentences. To achieve this, Chomsky (2013) introduces an LA which applies at the phase level, before transfer to the interfaces. Various, more specific conceptions of LA are the topic of Section 3 to follow. Notably, the converse—that labels may trigger operations including IM—remains a possibility. Indeed, this possibility is reframed as a central argument for the importance of LA, which appears to recast labels in a manner very similar to that of the loci of Collins (2002), albeit without abandoning the use of labels entirely.

2.7 Summary

With this, it is worth pausing the exposition in order to recapitulate the key developments of the status of labels within generative grammar as they arose under different guises.

There have been developments in the overall architecture of the grammar, from ST to MP. These have implications for the nature of the cycle, the timing of labelling and the accessibility of SOs. The nature of the rules and operations have evolved, from PS-rules which directly encode labels, to simplest Merge, which does not encode a label at all. There has also been a shift in what elements can label, from categorial features to entire LIs.

All of these issues continue to play a role in the discussion of labelling to this day. I proceed in Section 3 to detail the status of labels, now narrowly construed as an area of investigation, in the era following Chomsky (2013), summarising the important questions and the answers they

receive (if any) in present work.

3 Review of recent approaches

This subsequent discussion introduces and evaluates some contemporary proposals and assumptions regarding labelling, such that an approach can be formalised in Section 4. Labelling emerges as critical in the search for genuine explanation of I-language.

3.1 Key questions

The present era of labelling research assumes at minimum some kind of LA. However, almost every aspect of LA, on both the computational and algorithmic levels, is up for debate. As detailed in Section 2, the issues at hand have been up for discussion under different guises at least since the dawn of the generative programme. (25) summarises the central questions.

- (25) (a) How is the label of a particular structure chosen?
(b) What features can be identified as labels?
(c) At what point in the derivation does labelling occur?
(d) How do labels interact with other (post-)syntactic processes?

The following discussion will review the ways in which these questions have been given answers in the literature following the research programme instigated by Chomsky (2013, 2015). The discussion will remain purely formal and will provide the foundation for the theory to be formalised in Section 4.

Before considering the questions of (25) carefully in turn, it is worth briefly summarising the LA presented by Chomsky (2013, 2015). A definition is provided in (26).

(26) *The Labelling Algorithm of Chomsky (2013, 2015)*

- (a) For $\alpha = \{H, XP\}$, for H an LI and XP a complex SO, $label(\alpha) = H$.
(b) For $\alpha = \{XP, YP\}$, for XP, YP complex SOs, either:
(i) if YP is a lower copy of a moved SO, then $label(\alpha) = label(XP)$; else,
(ii) $label(\alpha) = \langle F_1, F_2 \rangle$, where F_1 and F_2 are ‘prominent’ features or sets of features shared by XP and YP , potentially with different values/interpretability.

- (c) For $\alpha = \{H, R\}$, for H, R LIs, H a functional head serving as a categoriser and R a root, $label(\alpha) = H$.
- (d) For H an LI, $label(H) = H$.

Note that this definition is already more articulated than anything formalised in the cited works, incorporating what I consider to be implicit therein.

If an SO does not satisfy any of the criteria in (26), it does not receive a label, and is thus, by hypothesis, rejected by the interfaces. Being an informal definition, a number of questions immediately arise as a consequence of (26). For instance, there needs to be a way of determining whether an SO is a phrase (i.e. complex) or a head. There also needs to be a way of determining the head of a complex SO. Case (26 c) also introduces a critical stipulation: that the only time $\{X, Y\}$ structures can occur (for LIs X, Y) is in the case where a root merges with its categoriser, following the theory of functional heads and bare roots in particular as espoused by Marantz (2013), cf. also Borer (2005a, 2005b, 2013). These issues will resurface in the subsequent discussion. Case (26 b)(i) requires ‘movement’ to be defined in a way that is accessible to LA, and (26 b)(ii), which I refer to as *feature-sharing*, requires articulation of a feature theory (see Section 4.6).

3.2 Finding the label

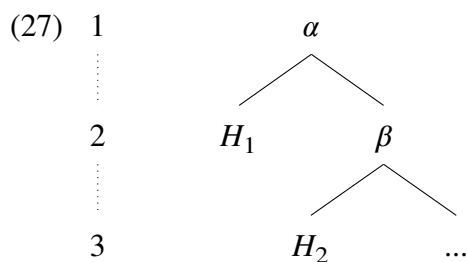
Returning to the questions in (25): firstly, consider (25 a). The basic position is outlined by Chomsky (2013, p. 43): “LA is just minimal search, presumably appropriating a third factor principle, as in Agree and other operations”. Some suggestions are provided as to how exactly this search is determined as ‘minimal’: “[i]n the best case, the relevant information about SO will be provided by a single designated element within it: a computational atom, to first approximation a lexical item LI, a head” (Chomsky, 2013, p. 43). Since labels are not found within the structure, only one level of search needs to be performed to identify whether the members of an SO are computational atoms (LIs) or are themselves sets (complex SOs). A set cannot serve as a label on first approximation, but an LI can. This approach clearly harks back to the original formulation of labels within BPS in Chomsky (1994a), in which it was stipulated that the label must be selected from one of the two mergees, i.e. $K \in \{\alpha, \beta\}$. With the BPS stipulation, but adopting (23 b) and LA, it is implied that search would have to find one of α or β , and would not

be allowed to search through either SO to find the label. This is motivated by MinTC: a certain conception of MinSearch that would entail that the members of the SO are examined first, and assuming optimality, one of these must then be chosen as the label.

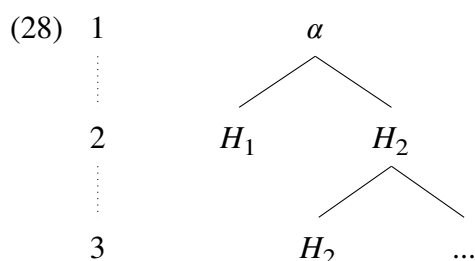
There are two clear problems with this approach. Firstly, such a simplistic LA simply does not suffice empirically, as noted in Section 2, since there are many cases where Merge applies to two non-minimal projections—namely, any time an external argument is introduced, and in all cases of movement (except head movement). Secondly, the assumption that LA may only consider LIs as ‘computational atoms’ is problematic. In the context of the BPS framework described above, it was claimed that $K \in \{\alpha, \beta\}$ was required on two bases: the introduction of external, diacritic elements would violate Inclusiveness, and the only alternatives, the union or intersection of the mergees, would be nonsensical. As pointed out by Blümel (2017), however, Chomsky’s reasoning on this matter fails to hold with an LA that instead relies on MS—specifically, a kind of MS that can ‘see into’ the feature content of LIs. Chomsky (2013) independently comes to a similar conclusion. The claim is that, where feature sharing occurs, “LA finds the most prominent element [] in both terms, and can take that to be the label” (Chomsky, 2013, p. 45). In this case, “most prominent element” clearly has a much looser definition than the earlier reliance on ‘computational atoms’, but it is one that needs to be clarified in any formalisation of LA.

Another aspect of LA is seldom discussed, but it is fundamental to its operation. The question is as to whether LA operates ‘top-down’ or ‘bottom-up’. In the latter case, the label of SOs ‘lower down’ in the structure are determined before those higher up. Unlike the earlier labelling systems which require the label to be specified at time of Merge, as in (23 a), with LA (26) it is possible to determine labels top-down at any point. This LA does not require the labels of any other SOs other than the one serving as input to the algorithm to be defined since they are computed recursively, as a result of (26 d).

It is nevertheless worth considering whether applying LA bottom-up would lead to a naturally more optimal system. For the sake of example, let’s make two assumptions regarding LA (to be derived from general principles): (a) LA starts at the most deeply embedded SO; and (b) LA keeps in memory any labels assigned at the most recently labelled level. In diagrammatic form, take (27), where H_1, H_2 are LIs.



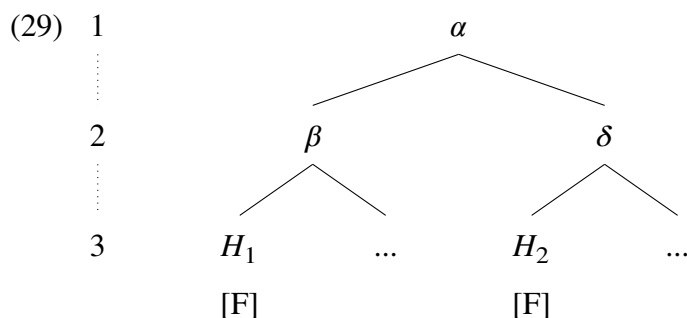
Operating bottom-up by assumption (a), LA must first determine the label of the SO $\{H_2, \dots\}$, indicated by β . Ignoring the contents of the additional structure indicated by the ellipses (...) for illustrative purposes, assume $\beta = H_2$ by (26 a). At this tier, LA must also determine the label of H_1 —by (26 d), this is simply the LI itself, H_1 . With the labels at tier 2 established, LA moves up to tier 1, and must label the SO $\{H_1, \{H_2, \dots\}\}$, indicated as α . By assumption (b), LA has access to the labels assigned at the previous tier, tier 2. As a result, LA can see that it assigned the labels $\{H_1, H_2\}$ to the members of the SO α . The representation can thus be informally diagrammed as in (28).



Maintaining assumptions (a) and (b) appears to create a dilemma. If LA looks only at the label information it is afforded by (b), i.e. the set $\{H_1, H_2\}$, then it has no way of determining that the object labelled by H_2 is in fact a complex SO, as a result of the elimination of diacritics from labels entailed by BPS. If LA instead uses the information provided by Merge, i.e. the set $\{H_1, \{H_2, \dots\}\}$, then (i) LA correctly identifies the label by (26 a), and (ii) there does not seem to be any need to postulate (b). However, there are good, independent reasons to think that something like (b) is true. The first is simply computational optimality: (b) minimises caching—MinCCD of (5)—and coupled with (a) minimises search—hence maximising throughput. If this were not the case, and LA needed to cache all computed labels, then this would entail no search optimisation, thus no benefit over top-down search.

Furthermore, option (26 b) actually *requires* that the label of complex SOs be accessible to LA. To demonstrate this, take the more complex example (29).²²

²²See Section X for an empirical cases with structures resembling (29).



Following the same logic as above, at tier 2, $label(\beta) = H_1[F]$ and $label(\delta) = H_2[F]$. What happens at tier 1 is the interesting thing: via (b), LA sees $\{H_1[F], H_2[F]\}$. It is thus able to select F , a feature shared between the two LIs, as a label, via (26 b)(ii), correctly deducing that $label(\alpha) = F$ as per Chomsky (2013). Without retaining a memory of the labels it has assigned, LA would have to search through a potentially very large structure in order to find the relevant features, and this would have to be done repeatedly, for future labels that are computed (within a given phase). This is thus a clear case of a small increase in space complexity in exchange for a massive reduction in time complexity, conforming to MaxTP.

Indeed, what this has shown is that LA (26) does not need to access the full syntactic structure at any point—rather, the information given at the previous level suffices. Let’s (informally) define a stronger version of the property (b) of LA as the *Goldfish Property* (GP) as in (30).²³

(30) *The Goldfish Property*

LA has access only to the labels computed at the previous level of a syntactic representation.

Rizzi (2015, f.n. 6) also makes the observation as above that the BPS system appears unable to locally distinguish between heads (LIs) and phrases (complex SOs), a distinction that is required by LA (26). Rizzi (2016) further develops a potential solution, namely to introduce a feature *Lex* that is stipulated to be present on all lexical items. The feature *Lex* indicates that an SO is a head, and it may or may not be included in labels created from such SOs (in the case of complex head formation via head movement, for example, *Lex* is included in the label). There are two immediate issues with this proposal: (a) the proposal violates TLTB (FI), because *Lex* does not have any clear interpretation at the interface; (b) there is no clear way of determining in

²³Alluding to the allegedly poor memory of *carassius auratus*. In reality, this turns out to be a myth—see for instance Gee et al. (1994).

what scenarios *Lex* projects from first principles. By contrast, GP appears to be able to eliminate the problematic head-phrase distinction, by moderately increasing the input to the LA.

3.3 Form of labels

Question (25 b) arises as a result of the change of direction impelled by Chomsky (2013): a label no longer needs to be an LI, rather “it must be that LA seeks features, not only LIs – or perhaps seeks only features, in which case it would be similar to probe-goal relations generally, specifically Agree” (Chomsky, 2013, p. 45). This is reminiscent a pre-BPS label system of the sort described in Section 2, where categorial features, potentially with associated features, projected. Labels are not limited to LIs, but are more free, as may be determined by the structural context. It also brings Label even closer to Agree, which has always been assumed to search for individual features. This has also previously been assumed for Move (see Chomsky, 1995b, pp. 261–271 on the ‘Move F’ operation). Selecting only the required features for the label, potentially via sharing, is clearly more optimal, by MinSC.

With this in place, it must be established whether there are any restrictions regarding which features can serve as labels. Conforming with TLTB, it would be most optimal *not* to have any stipulations on this matter. Rather, in principle, any feature can serve as a (potentially shared) label. Nevertheless, certain labels may be blocked for at least three reasons: (a) certain labels may never arise because the configurations in which they would emerge never arise; (b) some labels may create structures that are nonsensical at the interface; (c) the selection of certain labels may cause the derivation to halt irrevocably (crash).

Chomsky (2015) does stipulate that certain heads are ‘weak’ and cannot provide labels. He notes this for roots, which I derive simply from their featurelessness in Section 4.5, Theorem X. More tricky is why T in particular should be unable to serve as a label. I will note two related possibilities, which there is no room to explore: (a) Chomsky 2021 says T doesn’t exist; (b) T never serves as a label for independent structural reasons. Each of these options requires empirical investigation that is beyond the scope of this thesis.

Notwithstanding this, I will propose a hypothesis on the form of labels which should guide future analytic Minimalist work into the form of labels, on the basis of the preceding discussion.

(31) *Hypothesis on the form of labels*

The features that can compose labels are determined by the set of derivable structural configurations in an I-language. There are no extensional bans on valid labels.

I will not return to this directly, although it serves as a guiding light within Section 4.

3.4 Timing of labelling

On (25 c), Chomsky (2013) marked a significant shift from the earlier labelling theory assumed in Chomsky (2008) and earlier, by eliminating the stipulation that labelled structures were required within syntactic computation, in order for syntactic objects to be ‘visible’ to further operations. As stated by Rizzi (2015, p. 321): “labeling can be deferred until when the structure is passed onto the interpretive systems, at the end of a phase”. Blümel (2017, p. 77) expresses the issue perspicuously: “labels do not enter syntactic structures blindly and as a byproduct of Merge/Narrow Syntax [i.e. as they would in classic BPS adopting (23 a)—LVS] but at a point where they are actually needed, namely the mapping to the semantic component/transfer”. There are, however, numerous complications related to the timing of LA with respect to Transfer.

Firstly, there is the question of when Transfer itself occurs, and which structures are affected by Transfer. In the first description of phases (following its precursor, ‘multiple spellout’, spelled out in particular by Epstein, 1999; Epstein et al., 1998; Uriagereka, 1999), Chomsky (2000, pp. 131–132) states that the entire phase is sent to the interfaces at the point of transfer (maintained by Franks & Bošković, 2001). Chomsky (2001), on the other hand, maintains two possibilities: either the entire phase is transferred (Chomsky, 2001, p. 12) or the *complement* of the phase (i.e. the sister of the phase head) is transferred (Chomsky, 2001, p. 13). However, as Bošković (2016b) explores, phasal complement transfer is unsatisfying by TLTB, as phasal complements have no status. Further, the preoccupation with the phasal complement/edge distinction appears to dissolve into contradiction, as explained:

If both multiple spell-out and S[uccessive]C[yclic]M[ovement] were to be defined strictly on phases, phases would be spell-out units and SCM would target phases. A problem, however, would then arise. It is standardly assumed that what is sent to spell-out is no longer accessible to the syntax. Given this assumption, it is simply not possible to state the domain for both spell-out and SCM in terms of phases. (Bošković, 2016b, p. 39)

Bošković’s (2016b) solution is to adopt the original model whereby the entire phase is transferred at spellout, but to accompany this with the Chomsky (2001) model in which a phase is

spelled out *at the next highest phase*. This is arguably conceptually necessary in any case—as stated by M. D. Richards (2007, p. 567), building upon a suggestion of Chomsky (2004), transfer of an entire phase at the level of that phase would preclude any further computation.

However, M. D. Richards (2007) makes use of phasal complement transfer to derive an important result, namely an independent justification of the process of feature inheritance hypothesised by Chomsky (2008). To summarise the argument briefly, M. D. Richards (2007) argues that phases and non-phases must alternate so that non-phasal heads can inherit the uninterpretable features (uFs) from the phasal heads, such that the uFs on phasal heads can be transferred at the same time they are valued, which they otherwise would not be able to as their phasal head host is part of the phase edge and thus not transferred. It would be preferable to maintain the deduction of feature inheritance, an important mechanism in Minimalist theory. It seems to me that this can be done by adopting instead Bošković's (2016b) solution. In this model, with phases being spelled out at the next-highest phase, SCM is only allowed in cases where the phrase directly above a phase is not also a phase—otherwise the contents of the lower phase would no longer be accessible for movement. As a result, the phase/non-phase alternation is required for those scenarios in which SCM is allowed. This alternation then allows uninterpretable features to be inherited in the relevant cases.

3.5 Labelling interactions

Finally, there is the question of (25 d). The most obvious case to consider with respect to intra-syntactic interactions comes in the interaction between labelling and Agree. There emerges a clear redundancy here, since both Agree and LA are claimed to reduce to MS. As will be discussed in Section 4, whether this is indeed the case is not so clear, since it hinges on the precise notion of MS that is adopted.

A second, related, and arguably even more important interaction of labelling is with *movement*. Movement can be licensed in two ways: (a) movement can be *base-driven*; or (b) movement can be *target-driven*. The now standard approach to SCM of Bošković (2007b) adopts option (a)—arguably conceptually necessary in the case of SCM, since movement must begin both before the moving element is transferred and before the ultimate resting place of the moving element is merged and thus even accessible to computation. Since lower copies are invisible to the labelling algorithm, one option to resolve a labelling conflict, as suggested in (26 b)(i), is

for one of the phrases to move. A stronger hypothesis would be that all movement is driven by labelling concerns. This hypothesis is developed in the *Generalised Dynamic Antisymmetry* (GDA) framework (Moro & Roberts, 2020). This would also have consequences for Agree—there appears to be a redundancy between labelling-driven movement and Agree-driven movement.

In addition to the syntax-internal considerations above, it is worth making some specific comments regarding the interaction between labelling and the interfaces. At a fundamental level, *IC* is the reason that labelling is presumed to exist at all. Following Chomsky (2004, et seq.), the primary motivation for labels is that an SO needs a label in order to be interpreted at C-I. There are, however, a range of suggestions in the literature, some claiming that both *IC(SM)* and *IC(C-I)*, or only one or only the other, require or at least make use of labels when interpreting SOs (Barrie, 2021; Takita, 2020; Takita et al., 2016).

There is a further perspective which has not yet been taken in this discussion thus far but which warrants attention with discussion of this question in particular. The perspective concerns an issue which arises at all levels of syntactic theory: that of *symmetry*. The two options for Merge presented in (23) can be reframed as the choice between *asymmetric* Merge (23 a), which encodes the projection of a label, and *symmetric* Merge (23 b), which simply forms a symmetrical set. It is a well-founded empirical result that asymmetries are pervasive in human language syntax. This is enshrined in the definition of *c-command*, introduced by Reinhart (1976).²⁴ The notion of *asymmetric* c-command was also demonstrated to have the potential to derive linear order from syntactic properties by Kayne (1994). If only one of the two inputs to Merge can serve as the label, this enforces asymmetry. However, option (23 b) enables other options, as labels may be assigned at the interface, according to interface conditions. Furthermore, as demonstrated by Moro (2000), extending the analysis of Kayne (1994), syntax appears to have an aversion to symmetrical surface structures.

It is also worth making some more speculative comments on how the result of labelling may have an effect upon interpretation of structures at the interfaces. That this occurs is a central empirical motivation for labelling in the first place: certain SOs must be labelling at the

²⁴Reinhart (1976, p. 32) defines c-command as follows: “Node A c(onstituent)-commands B if neither A nor B dominates the other and the first branching node which dominates A dominates B”. Reinhart was not the sole progenitor of the general idea—the relation has its roots in the ‘in construction with’ relation of Klima (1964) and the ‘superiority’ relation of Chomsky (1973). There are other potential definitions of c-command, notably the derivational version proposed by Epstein et al. (1998).

interfaces in order for the structure to be interpreted correctly.

3.6 Summary

It is impossible to provide a maximally detailed account of all potential questions and solutions to issues surrounding labelling, affording to the fact that labelling strikes at the heart of fundamental questions about how syntax works, and how theories of syntax should be constructed. Nevertheless, this subsection has served as a sufficient review of the central questions in (25) and some proposals that exist in the literature for answering them. It has also provided some original analysis of some specific points, paving the way for the formalisation in Section 4, which will provide one set of answers in a precise yet necessarily provisional manner.

4 Formalisation

The formalisation to follow is framed in a manner inspired by the presentation of Collins and Stabler (2016), henceforth C&S. In Section 4.1 through Section 4.4, fundamental definitions are adapted from C&S. In most cases, however, this adaptation is not verbatim; rather, numerous adjustments are documented, in line with the discussions within the preceding sections. Section 4.5 presents the core of the labelling algorithm: the procedure of minimal search. This subsection builds upon the algorithm presented by Ke (2019), adapting it into the style of C&S and adjusting its operation in line with the results of the preceding investigation. Section 4.6 presents a barebones formalisation of features, enabling the definition of the labelling algorithm itself in Section 4.7. Agreement is given brief attention in Section 4.8. Finally, some key theoretical results are discussed in Section 4.9.

Before embarking, there are some initial caveats to bear in mind. Firstly, the model assumed by C&S is one which does not assume in full the proposals of Distributed Morphology (DM), in particular the mechanism of Late Insertion (Halle & Marantz, 1993). Adopting this aspect of DM would require a major reformulation of Transfer, which is beyond the scope of the present work (cf. Milway, 2021, f.n. 2 for a similar point). Late Insertion entails that lexical items are not directly bundled with phonological features, but such features will play only a limited role in this formalisation in any case. Nevertheless, one of the assumptions within DM will be adopted, as it is now commonly held within a range of theoretical approaches. Namely, lexical *roots* are considered to be ‘bare’, without any *syntactic* features (see Definition 3 and Bauke & Blümel, 2017; Borer, 2005a, 2005b, 2013; Marantz, 1997). Properties such as category are instead provided by closed class categorisers like *n*, *v*, and *a*.²⁵ I will follow Chomsky (2015) in Section 3.1 by assuming that roots cannot provide a label (as a result of having no syntactic features to project) and that, consequently, the root must Merge directly with its categoriser as the first step of a derivation. The root is thus the zero-element of Watumull (2015); the start

²⁵One consequence of this is that categoriser *v* must be totally distinct from the ‘light verb’ *v* that appears above VP and introduces the external argument. *v*P has a complicated history, developing out of Larsonian VP-shells (Larson, 1988), termed ‘light *v*(erb)’ by Chomsky (1995b, pp. 315–316), and extended by Kratzer (1996), who terms the phrase VoiceP. VoiceP is also used by subsequent researchers, but tends to carry more theoretical baggage. In the interest of staying true to the literature, especially in the context of phases, where *v*P standardly refers to the lower (thematic) phase, I will use *v* to refer to both the light verb and the categoriser; the semantics should be clear from the context.

symbol of the derivation.²⁶

Basic (naïve) set theory is assumed. Standard notation is as follows, borrowing partly from C&S (p. 43). Sets are written with curly braces $\{\dots\}$ and are unordered. The following symbols are used to represent relations between sets and their elements: \in (is a member of), \cup (union), \cap (intersection), \subseteq (is a subset of), \subset (is a proper subset of).²⁷ The empty set is written \emptyset or $\{\}$. Given sets A and B , the set difference $A - B = \{x \mid x \in A, x \notin B\}$. A sequence $\langle \dots \rangle$ is ordered; the empty sequence is written ε or $\langle \rangle$. A sequence of length 2 is a pair, one of length n is an n -tuple. Free variables are assumed to be universally quantified, such that x is shorthand for $\forall x, x$. The Cartesian product is represented by \times ; the following shorthand using indices will also appear: $X^n = X_1 \times X_2 \times \dots \times X_n$. The arbitrary union is notated as $\bigcup X = x_1 \cup \dots \cup x_n$ for a set $X = \{x_1, \dots, x_n\}$, and can also be given limits. The Kleene closure $A^* = \bigcup_{i=0}^{\infty} A^i$.²⁸ A *function* is a mapping between sets, and can take any number of *parameters* (when a function is invoked, these will be called *arguments*). For notational ease, where a set is passed in as an argument, it may either be taken itself as the entire argument, or as shorthand for its members being the arguments (see Definition 31 for an example of the latter case).

Tree diagrams will occasionally be used in place of complex bracketed sets. These trees will often encode more information than is present in the sets themselves (such as labels and linear order) for expositional reasons, as is standard. For lack of space, the precise relationship between graph-theoretic trees and sets will not be explored. It suffices to say that there is a surjective (many-to-one) function $Z(t)$ which maps a tree t onto its corresponding SO—multiple trees could be used to represent the same SO.²⁹

4.1 Preliminaries

C&S provide a number of definitions which lay out the foundations of a formalisation of Minimalist syntax. Many of these will, however, need some revision in light of the preceding discussions in Section 1, Section 2 and Section 3. The purpose of this subsection is to lay out a

²⁶Note that this contradicts the assertion of Adger and Roberts (to appear, p. 7) that it is phase heads which provoke the start of computation. This cannot literally be true, however, since computation must begin before the phase head is introduced, since the phase head can itself only be introduced by a computational operation (in my formalisation, *Select*, see Section 4.3). This issue is discussed further in Section 4.4.

²⁷For a set A , $A \subseteq A$ but $A \not\subset A$.

²⁸For a fuller explanation of the elements of naïve set theory beyond the scope of this thesis, see Enderton (1977) and Kaplansky (1972). See Hopcroft et al. (2013, p. 87) on Kleene closure (a.k.a. the ‘Kleene star’).

²⁹See Avigad et al. (2017, Chapter 15) on functions in set theory.

revised set of fundamental definitions.

Naturally, the first definition to come is that of I-language itself. (cf. Section 1.4.1).³⁰

Definition 1. UG is a 9-tuple:

$$UG = \langle F_{PHON}, F_{NS}, F_{SEM}, Select, Merge, Agree, Label, Transfer, FormCopy, f_{\Phi}, f_{\Sigma} \rangle$$

where $F_{PHON} \cap F_{NS} = F_{NS} \cap F_{SEM} = F_{SEM} \cap F_{PHON} = \emptyset$.

Thus, I-language consists of three non-intersecting sets of features, and six further functions. Although termed UG, constituent operations are intended to draw on domain-general mechanisms as much as possible, in line with OM and MM. The choice of feature sets is intended to be as theory-neutral as possible, without making any claims as to the precise structure of features and lexical items except where necessary. In the elaboration of *Label* and *Agree*, some development of the feature theory will be required.³¹

The definition is notably more complex than that adopted by C&S. Following Milway (2021), I have included *Agree* as a function. I have also added *Label*, to be defined. As per Section 1.4.1, f_{Φ} and f_{Σ} are the phonological and semantic mappings respectively. I include these as they ought to be defined in a complete formal theory of I-language, although they will receive little attention here. *FormCopy*, an operation adapted from Chomsky (2021a), receives justification in Section 4.2.

As described in Section 1, UG is (broadly) species invariant. Variation is accounted for via the lexicon, *LEX*, as per the BCC. In S_0 , $LEX = \emptyset$. In the final state, *LEX* consists of lexical items, composed of features. This is accounted for in the following definitions, lifted from C&S.

Definition 2. A lexical item is a triple: $LI = \langle PHON, SYN, SEM \rangle$ where $PHON \in (F_{PHON})^*$, $SYN \subseteq F_{NS}$, and $SEM \subseteq F_{SEM}$.

Definition 3. An LI $\langle PHON, SYN, SEM \rangle$ is a *lexical root* (L-root) iff $SYN = \emptyset$.³²

Definition 4. A *lexicon* is a finite set of LIs.

Definition 5. I-language is a pair $L = \langle LEX, UG \rangle$, where *LEX* is a lexicon.

³⁰Some notes on conventions: sets are indicated with capital letters (or words in all caps), functions with *CamelCase*.

³¹For one possible formal theory of features, see Adger (2010) and Adger and Svenonius (2011). Cf. also Carlson (2010), Roberts (2019), Song (2019) and Stockwell (2015) for theories of features with more or less coverage and with varying degrees of formality.

³²I leave open the possibility that *SEM* is empty for roots (see Borer, 2013). Since I leave the feature theory in Section 4.6 very vague, this is not a problem. Note however that if ‘interpretable’ features are members of F_{SEM} and can serve as labels, this would leave Theorem 1 underivable.

4.2 Copies and repetitions

The copy/repetition distinction is a fundamental problem in Minimalist syntax. The problem lies in distinguishing between minimally distinct syntactic objects like those in (32), where the ‘John’ sister of V in (32 a) represents a moved element (copy; lower copies indicated with angle brackets), and in (32 b) a repetition (referring to two different people, on the standard reading).³³

- (32) (a) {John, {was, {_{vP} ⟨John⟩, {_{vP} v+seen, {_{VP}{v, ⟨see⟩}, ⟨John⟩}}}} (passive object raising to subject via phase-edge)
- (b) {John, {T, {_{vP} ⟨John⟩, {_{vP} v+saw, {_{VP}{v, ⟨see⟩}, John}}}} (standard declarative)

This problem is unavoidable in a formalisation of Minimalist syntax—here, I intend to take a somewhat novel approach. Section 4.2.1 will briefly review some of the options and set out a path forward. Section 4.2.2 will go into some more depth on the formal nature of ‘copies’.

4.2.1 Distinguishing copies and repetitions

Following TLTB, GB-era symbols like *trace* and indices cannot be introduced to mark movement; this would also violate the NTC. Secondly, recall that there is no distinction between Merge and Move; rather, all structure building is by Merge (and by further hypothesis all that reaches the interfaces has been constructed by Merge). A syntactic object constructed by ‘internal’ Merge is identical to one constructed by ‘external’ Merge (see Definition 15 below). Further, to account for ‘trace-invisibility’ effects with respect to movement (discussed above), both NS and the interfaces need to be able to distinguish copies and repetitions. Therefore, NS, or at least all relevant operations within NS, need to be able to distinguish copies and repetitions.

Despite the centrality of this issue, Collins and Groat (2018) come to the worrying conclusion in their review that “no adequate proposal exists in [M]inimalist syntax for distinguishing copies and repetitions” (Collins & Groat, 2018, p. 2). Collins and Groat (2018) review a number of approaches; I will briefly touch on three, two of which are discussed by Collins and Groat (2018). One option is to use *chains*. As shown by C&S, however, chains introduce a vast amount of machinery that complicates the definition of Merge, and should be abandoned for the sake of

³³Morphological complications like affix-hopping (Chomsky, 1975) are ignored. The two v’s represent either the light verb or the verbal categorisers, as contextually clear (see footnote 25). For a theory of English passives with respect to labelling, see Burrows (2022).

TLTB. Further, C&S prove (in their Theorem 4) that chain-based structures and the multidominance structures they use in the rest of their paper are isomorphic, which one can infer makes the chain-based theory (or at least the formulation they adopt) inferior. A second proposal, adopted by C&S but not discussed by Collins and Groat (2018), is to augment each LI into a *lexical item token* (LI_k) when introduced into the lexical array. An LI_k is an LI with an associated unique index k . Copies of the same LI_k will thus have the same index. This option evidently violates Inclusiveness, but one could argue that this is a principled violation, since otherwise the EM/IM distinction would be unformulable.

Nevertheless, I do not adopt LI_k s in this formalisation, in the interest of staying as true as possible to the recent literature, especially Chomsky et al. (2019) and Chomsky (2021a). The third option adopted in these works is the idea of a *phase-level memory*. As noted by Collins and Groat (2018, p. 12), Chomsky separately notes two possibilities. A third is proposed by Chomsky (2021a). These options are summarised in (33).

(33) *How could phase-level memory distinguish copies and repetitions?*

- (a) It must be the case that “within each phase each selection of an LI from the lexicon is a distinct item, so that all relevant identical items are copies” (Chomsky, 2008, p. 145).
- (b) “At TRANSFER, phase-level memory suffices to determine whether a given pair of identical terms Y, Y' was formed by IM.” Y and Y' are copies if so, else they are repetitions (Chomsky et al., 2019, pp. 246–247).
- (c) There is a “convention”, “STABILITY”, which states that certain occurrences of the same symbol are related; there is a rule “FORMCOPY (FC)” which assigns the *Copy* relation to certain identical symbols and which must adhere to STABILITY. “FC applies at the phase level and is interpreted (mapped to CI), not entering into further computation” (Chomsky, 2021a, pp. 16–17).

Collins and Groat (2018) interpret both (33 a) and (33 b) as being problematic for the same reason. Briefly, with reference to (33 b), establishing whether IM or EM was applied in a particular derivational stage would require access to the previous state of the workspace, but this violates the strict Markovian property of the derivation (cf. Chomsky, 2021a, p. 20), with dire

consequences for interpretation. *Pace* Collins and Groat (2018), I interpret (33 a) to be equivalent to introducing LI_k s as done by C&S, which affords phase (lexical array) level uniqueness to tokens. It is thus unsatisfactory by Inclusiveness (*pace* the claim of Chomsky, 2008). The fate of the original conception of the *Numeration* (Chomsky, 1995b) fares similarly.

(33 c) is more cryptic yet at the same time suggestive. The strong claim, as I interpret it, is as follows. The copy/repetition distinction is required *only* at the interfaces—hence, syntactic operations cannot make reference to copies or repetitions. Further, the distinction is *determined* at the interface, not in the syntax. The corresponding illegitimacy or deviance of a derivation with respect to misinterpretation of copies/repetitions emerges from interpretation, but this interpretation, like Merge, may operate freely. To take SM as illustrative: the structure (32 a) could be pronounced “John was seen John”, but in this case the two ‘John’s would be parsed as repetitions by a rule of SM and so would be interpreted as gibberish at C-I by the θ -criterion (each DP must be assigned one and only one θ -role; cf. Chomsky, 1981, p. 36).³⁴ Indeed, this close interaction between the interfaces is an interesting result of the hypothesised proximity of the interfaces to NS (and thus to each other) established in Section 2.2. This has further implications for minimality, which will be touched upon in Section 4.6.

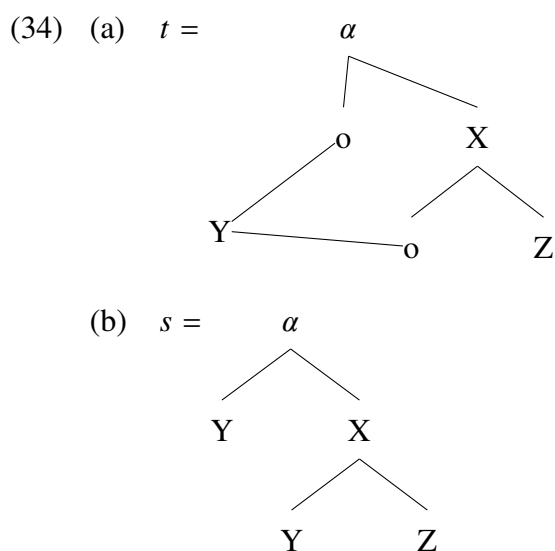
Fully exploring the consequences of the FC model proposed by Chomsky (2021a) is well beyond the scope of the present work. In particular, it will be needed to establish what impact this has on the analysis of island effects. Nevertheless, in the interest of being forward-looking, it will be adopted, accepting its preliminary nature as a caveat. With this established, it is possible to continue the formalisation. As a result of abandoning LI_k s, there will be some small adjustments in the definitions to follow as compared to C&S.

4.2.2 Copies and multidominance

It is important to note that, thus far, the notion ‘copy’ has been assumed in a non-technical sense. As is clear from the discussion in Section 1.4, the generally assumed, intuitive idea is that an internally-Merged object is identical in its source and target positions. In a formalisation that uses some set-theoretic machinery, some necessary properties become apparent. For instance,

³⁴In (33 c), Chomsky (2021a) appears to imply that the copy/repetition distinction is required only at C-I—this cannot be correct, as SM needs to be able to deduce lower copies to obviate their pronunciation. It must be a part of *Transfer*. This being said, it seems sensible that copy *formation* not be forced by SM, since lower copies *can* be pronounced, as evidenced above, and indeed *are* pronounced in certain contexts in certain languages and in child language.

a standard assumption is that sets contain unordered, unique objects—i.e. $\{a, a, b\} = \{b, a\}$. This being the case, take a structure that could plausibly be the output of IM, $X = \{a, \{a, b\}\}$. Gärtner (2022) points out that, on standard set-theoretic assumptions, the two instances of a are not ‘copies’; they are identical objects. [I shall recapitulate the proof below, after Merge has been defined. which assumes the existence of a ‘bracket-erasure’ function sp , and that a and b are *urelements*, viz. indivisible.] What this entails, then, is that a *multidominance* approach (Citko, 2011a, 2011b) to SOs appears to be in line with Minimalist assumptions. In other words, for the two trees t and s in (34), for the corresponding SOs $Z(t) = Z(s)$.



Note that the graph-theoretic complications entailed by the representation t have no theoretical status within the formalisation under discussion, they are merely the result of diagrammatic games (cf. Chomsky, 2019a). For instance, from the diagram it appears that there are two nodes that could be considered ‘roots’, α and Y , say if ‘root’ were defined graph-theoretically as ‘a node not dominated by another node’. However, from the set-theoretic representation (35), which is the only one that has any theoretical status having been constructed by *Merge*, it is clear that there is no such confusion.

$$(35) \quad \alpha = Z(t) = Z(s) = \{\alpha Y, \{X Y, Z\}\}$$

Similarly, one could protest that this would eliminating the ‘binary-branching’ property of syntactic trees, since if, say, Y were merged with α , this would result in a structure in which, diagrammatically, Y would have three branches connecting it with each of its occurrences. Again, however, this property of the tree diagram has no theoretical status. Indeed, (internally) Merging

Y with a in this way would be entirely legitimate and would continue to satisfy the Extension Condition (see Section 4.9). A further illegitimate operation, namely externally merging an LI with Y, would also not be possible, since Y is not a root (as per Definition 12 below).

Returning to Gärtner’s (2022) original concern: Chomsky et al. (2019) claim that multidominance approaches are misguided, precisely because they suggest the existence of “complex graph-theoretic objects [that] are not defined by simplest MERGE”. Following the argument as set out here, this concern is unwarranted. What is part of the system is *Merge*, which forms sets, and without added complication, the *urelements* (indivisible elements) of these sets from the perspective of *Merge* are LIs, which entails that multiple occurrences of the same LIs are not copies but one and the same object. Gärtner (2022) concludes that when analysing the formal properties of a system, one must first note whether the formal tools in use are being applied at the meta level, talking ‘about’ I-language, or whether they are at the object level, namely part of the system itself. Secondly, one must be aware of the difference between notation and content, avoiding falling into the trap of “excess notation over subject matter” (Quine, 1941/1995, p. 5). In answer to both of these points, the subject matter at hand is I-language, in particular the representations constructed by I-language, namely SOs and the labels of these SOs (features). SOs are sets formed by *Merge* using objects from the lexicon, also hypothesised to be sets. Nevertheless, properties of sets outside of the fact that they are formed by *Merge* should not be considered *a priori* allowed. In this formalisation, I assume only the most basic—in particular, set membership, and the natural operations of union and intersection, and compositions of these operations. Gärtner (2022) suggests calling these sets, with limited properties, ‘M-sets’, although I do not adopt this terminology here. Another mathematical object in use alongside sets is functions, in particular the notion borrowed from computer science, following standard computational/cognitive assumptions (cf. Gallistel & King, 2010).

As a final note on multidominant structures: I am not introducing the full complexity of multidominance as set out by Citko (2011b). This theory requires complications to be introduced to the Merge operation itself, namely ‘Parallel Merge’ and ‘Sideways Merge’, which are independently ruled out by a principle of computational optimality noted by Chomsky (2019a, 2021a) related to accessibility, which I derive in Theorem 3.

4.2.3 Defining occurrences

There are, however, occasions where different occurrences of SOs within a set-theoretic structure need to be distinguished. One can define the notion occurrence to handle this. C&S define occurrence in terms of immediate containment, presenting this alongside a number of useful definitions and theorems. I will not repeat these here, although I will adopt a compatible definition of occurrence which suits our present purposes, and which is (implicitly) corroborated by Epstein et al. (2020).

Definition 6. An *occurrence* of an SO A is a *path*, a sequence of SOs $P = \langle X_1, \dots, X_n \rangle$ where for all $0 < i < n$, $X_{i+1} \in X_i$, such that $X_n = A$. An occurrence of A at *position* P is denoted A_P .

This definition of occurrence would also enable a formalisation of *FormCopy*. Since this would take us too far afield, I leave this for future work.

4.3 Merge, workspaces and derivations

With the atoms of computation in place, it is possible to implement the most important definitions—namely, what representations syntax constructs and how these are computed.

Definition 7. X is a *syntactic object* SO iff:³⁵

- (i) X is a lexical item, or
- (ii) X is a set of SOs formed by application of *Merge*.

SOs are thus defined recursively.³⁶ Condition (ii) is much stronger than the definition in C&S. I believe it is justified following the preceding discussion in Section 4.2.2: by opening up the definition of SO to be simply any set that contains other SOs, we stray into the territory of notational games—ontologically speaking, a set is only an SO if it is formed by *Merge*, to be defined in Definition 15.

A couple of useful relations can be taken straight from C&S, primarily to simplify some definitions to come.

Definition 8. For SOs A and B , B *immediately contains* A iff $A \in B$.

³⁵‘If and only if’, i.e. logical equivalence (\leftrightarrow).

³⁶‘Recursion’ is used in this thesis in the strictly mathematical sense, as extensively discussed by Watumull et al. (2014).

Definition 9. For SOs A and B , B contains A iff

- (i) B immediately contains A , or
- (ii) for some SO C , B immediately contains C and C contains A .

I add to these a further definition for ease of exposition, effectively to represent the inverse of containment (cf. Chomsky, 2019a; Epstein et al., 2020).

Definition 10. For SOs A and B , B is a term of A if A contains B .

C&S’s definition of a ‘lexical array’ is not required following the elimination of LI_k s, in line with Chomsky et al. (2019), cf. also C&S (f.n. 4). How this plays with cyclicity and phases will be discussed in Section 4.5. Consequently, a ‘stage’ of a derivation is equivalent to a workspace: “WS [the workspace] represents the stage of a derivation at any point” (Chomsky et al., 2019, p. 245).

Definition 11. A workspace W is either a set of SOs or \emptyset .

As per C&S (p. 47), “[a] workspace includes all the syntactic objects that have been built up at a particular stage in the derivation”. Note that, using Definition 15, a workspace is not actually an SO, as in Chomsky (2020b, p. 37) but unlike in C&S.

Definition 12. For any SO X and workspace W , if $X \in W$, X is a root in W .

Careful not to confuse this with the very distinct notion of L-root in Definition 3. There may be multiple roots in a workspace. This definition will prove useful in the definition of Merge and derivations below.

Next, it is time to derive the operation *Merge* itself. It is possible, as done by C&S, to define Merge much as done in Section 1.4 and in (23 b)—namely, to say that Merge takes two inputs and returns a set as output. Formally, however, either Merge must make reference to the workspace, or there must be some additional stipulation outside of Merge as to the nature of a legitimate derivation. The latter option is taken by C&S, but this entails ‘derivation’ to have properties that extend beyond UG, but which are not given a clear third factor justification. Instead, I adopt here the ternary definition from Chomsky (2021a), which is provided more formal structure by Seely (2021), and which is adopted here. This definition further requires a notion of *accessibility*, which determines which SOs are available to be Merged.

Definition 13. An SO X is W -accessible within a workspace W iff W contains X .

Note that SOs in LEX are always accessible in some broader sense, since they can be externally Merged. This should, however, come at a cost—EM is penalised over IM because of the increased search space (and hence computationally by MinSearch). Hence, the more restricted form of accessibility, W -accessibility, is defined. This notion is to be revised below, in Definition 21 and Definition 24.

Chomsky claims that EM is more computationally complex than IM as a consequence of requiring ‘massive search’. This is accurate only by hypothesis, however. *A priori*, with no assumptions made about the structure of the lexicon, it is not possible to make any claims as to how lexical search operates. There is no reason to believe that a search algorithm similar to one used to trawl syntactic structures would be in place in the lexicon. Indeed, one could quite easily conceive of a data structure for the lexicon that requires no search at all, and that has a constant-time access algorithm. For example, imagine there is a deterministic function $\chi(x)$ that returns a unique output for each input. Each unique output corresponds to a possible location in memory where a lexical item can be stored. Assuming that the procedure of applying Merge has some kind of key k ready for which the lexicon will be ‘searched’, the procedure can merely run $\chi(k)$, which returns the needed location in memory, such that the full LI can be accessed.³⁷

Nevertheless, we want to capture the idea that ‘NS-internal’ computation is in some way more optimal than computation which accesses the lexicon. Hence, I introduce the operation *Select*, which must operate in order to introduce new items into the workspace from LEX . This definition is borrowed from C&S (f.n. 4), adapted to drop the notion of lexical array.

Definition 14. For an SO $X \in LEX$ and workspace W , $Select(X, W) = \{X\} \cup W$.

Next, the primary way of manipulating the workspace, and the most fundamental operation in syntax: *Merge*.

Definition 15. $Merge(P, Q, W) = \{\{P, Q\}, X_1, \dots, X_n\} = W'$, such that

- (i) P and Q are W -accessible within workspace W ,

³⁷This data structure is known as a *hash map* or *hash table* in computer science. It seems unrealistic that the lexicon actually works like this, but the actual implementational details are irrelevant at this algorithmic level of analysis (see Section 1.1).

(ii) $P \neq Q$, and

(iii) for all SOs Y , $(Y \in W \wedge Y \notin \{P, Q\}) \rightarrow Y \in \{X_1, \dots, X_n\}$.

In sum, the *domain* of *MERGE* is all the W -accessible SOs in a workspace; the *codomain* of *MERGE* is the (discretely infinite) set of all SOs (cf. Watumull, 2015). The final condition on Merge is required according to Chomsky (2021a) in accordance with the SMT. It sustains MY (see Section 1.4.5). Importantly, there is no condition on the nature of the input SOs P and Q other than that they are W -accessible, which is substantively necessary (by Inclusiveness), and that they are distinct, ruling out self-Merge (*pace* Adger, 2013, see C&S, p. 48). IM and EM are thus totally equivalent at time of Merge, and thus cannot be distinguished even on the phase-level, assuming strictly Markovian derivations (see Section 1.4.5 and also Definition 17 below). The only difference is that EM requires application of *Select* at the previous stage of a derivation.

Next, derivations themselves may be defined, making use of this new definition of Merge.

Definition 16. A *derivation* within L is a finite sequence of workspaces $\langle W_1, \dots, W_n \rangle$, for $n \geq 1$, such that:

(i) $W_1 = \emptyset$,

(ii) For all i , such that $1 \leq i < n$, and for some (accessible, distinct) SOs A, B :

(a) (*derivation-by-select*) $W_{i+1} = \text{Select}(A, W_i)$, or

(b) (*derivation-by-merge*) A or B is a root and $W_{i+1} = \text{Merge}(A, B, W_i)$.

The root condition entails that Merge is always ‘at the root’, which is necessary to derive MY. From this emerges a natural definition of *workspace accessibility*.

Definition 17. A workspace W is *accessible* at stage i of a derivation $\langle W_1, \dots, W_n \rangle$ iff $W = W_i$.

This captures the strict Markovian property of derivations. Indeed, workspace accessibility constitutes a generalisation of the Goldfish Property introduced in (30). Whilst in the context of its introduction, the property applied only to LA, being a principle of computational optimality it should hold for every operation within L . The property can thus be generalised into a principle, and ideally would emerge as a theorem.

Finally, the culmination of a derivation is a single SO.

Definition 18. A syntactic object X is *derivable* within L iff there is a derivation $\langle W_1, \dots, W_n \rangle$ where $W_n = \{X\}$.

4.4 Phases and cyclic transfer

A conclusion from Section 2 and Section 3 is that there should optimally be only one computational cycle involved in generating syntactic structures. Operations that are countercyclic should be avoided, as well as assumptions that there can be multiple cycles operating in series—this latter point was essential to GB but abandoned in Minimalism. As a consequence, one must beware of introducing cycles underhandedly, masquerading as other operations or as being ‘at the phase-level’.

On my view, Transfer is a potential source of accidental multiple cyclicity. One questions that could arise with respect to this is, what kinds of objects does Transfer ‘send’ to the interface? But even this may be a misnomer, as the scare quotes illustrate: as emphasised by Chomsky (2021a) and noted above in Section 2.2, derivational access can in theory be at any point. If this is the case, the interface should be able to see the entire generated structure at once. Also, along similar lines, it is the entire workspace that should be ‘transferred’, viz. viewed. Otherwise, if interfaces were arbitrarily able to select parts of the workspace to view, there would be massive overgeneration (i.e. beyond what may be covered as deviance, cf. Chomsky, 2019a, 2021a).

A problem with Transfer that is relevant here is what C&S (p. 67) dub the *Assembly Problem*, labelling an issue that arose in the original presentation of multiple spellout (Uriagereka, 1999). The problem boils down to a tension between the requirement to derive subjacency effects, namely phase impenetrability, and the need to retain memory of where, in the syntactic structure that is being derived, the transferred element was. An additional, related problem is that the internal structure of objects that have been ‘transferred’ may need to be visible to later operations—C&S (p.p. 72–73) describe how this is the case with the phenomenon of remnant movement. C&S formalise what they call the *plug-back-in* model, but this option both erases the internal structure of what is transferred and requires the definition of SO to be expanded. This would also prevent Agree from crossing phase boundaries, which is argued for by Bošković (2007a). Further, this clearly violates the NTC, contradicting computational optimality. C&S (p.p. 73–74) sketch an alternative which focuses on the crucial notion, namely *accessibility*. To capture the main effect of phase impenetrability, *Merge* should not be able to apply to trans-

ferred elements. We have already established the importance of accessibility with the notion of W -accessibility in Definition 13, so this will clearly need to be modified in order to deal with cyclic transfer. C&S (p.p. 73–74) suggest that the computational system should “keep a set of syntactic objects that have been transferred, and then block all access to those transferred elements”—in determining accessibility, the set of transferred elements needs to be checked.

Definition 19. A set of transferred elements T is either a set of SOs or \emptyset .

Definition 20. A stage of a derivation $S_i = \langle W_i, T_i \rangle$, where W_i is a workspace and T_i is the set of transferred elements.

The introduction of more complex derivational stages will require some adjustments to the definitions given in Section 4.3. First, let’s redefine W -accessibility.

Definition 21. An SO X is W -accessible at stage $\langle W_i, T_i \rangle$ iff W contains X and T does *not* contain X .

Transfer can now be defined simply, as an operation ranging over sets of transferred elements.

Definition 22. $Transfer(X, T) = T \cup \{X\}$, for set of transferred elements T and X an SO.

Transfer should be no more complex than this. However, there appears to be a conflict here with how phases operate—all operations should be on the phase-level, including Merge, Agree, Label and Transfer. This cannot literally be true, since this entails lookahead, as noted by [EKS] and accepted by [Chomsky]: operations must occur before any phase head has been Merged. When coupled with the greater interface proximity afforded by the Minimalist architecture, I believe this entails a return to a stricter, bottom-up cyclicity. Further, there is no need for any algorithm or operation to occur outside of this cycle, as all operations are interface-driven.

Phases exist to further restrict accessibility, creating subjacency effects (Chomsky, 1973). In the Minimalist literature, this comes in the form of the PIC (see Section 1.4.5). Phases also define valid Transfer domains—transfer occurring at any other point does not construct a valid derivation. Note that the principle that operations take place freely and that access to the derivation can be at any point (Chomsky, 2021a) entails that Transfer, like Merge, is not ‘triggered’ by, say, the Merging of a phase head. It may be possible to extend this formalisation to capture a broader interpretation of the idea that all operations take place ‘on the phase level’, which is

more in line with typical Minimalist assumptions (cf. Adger & Roberts, to appear). There is not space here to discuss developments along these lines.

With *Transfer* being defined as in Definition 22, it is necessary to redefine derivation, originally Definition 16, incorporating the more complex stages of Definition 20 and introducing derivation-by-transfer.

Definition 23. A *derivation* within L is a finite sequence of stages $\langle\langle W_1, T_1 \rangle, \dots, \langle W_n, T_n \rangle\rangle$, for $n \geq 1$, such that:

- (i) $W_1 = T_1 = \emptyset$,
- (ii) For all i , such that $1 \leq i < n$, and for some (accessible, distinct) SOs A, B :
 - (a) (*derivation-by-select*) $T_{i+1} = T_i$ and $W_{i+1} = \text{Select}(A, W_i)$, or
 - (b) (*derivation-by-merge*) $T_{i+1} = T_i$ and A or B is a root and $W_{i+1} = \text{Merge}(A, B, W_i)$.
 - (c) (*derivation-by-transfer*) $T_{i+1} = \text{Transfer}(X, T_i)$ for X a W-accessible SO contained in W_i and Y a root, such that $\text{Label}(X)$ and $\text{Label}(Y)$ are phasal and Y contains X .

Note that derivation-by-transfer requires *Label*, although I reserve definition of this until Section 4.7, after the definitions of MS and of features. I have also left the notion of *phasal* undefined—indeed, I leave this entirely open-ended, as there has not been space to review approaches to phases in enough detail to establish a way forward. Where necessary, I will adopt the standard approach discussed in Section 1.4.5: a label α is *phasal* if the categorial feature corresponding to C or v^* is a member of α .³⁸

4.5 Minimal search

It was established in Section 1.3 and Section 1.4 that MS is of central importance to the operation of C_{HL} , in particular in the operations of agreement and labelling. Minimal Search is a subcomponent of computational optimality, as represented in (5). Optimal search algorithms have thus been an object of study within computer science effectively since its inception. Deciding on the algorithm or class of algorithms which is employed by I-language is an empirical matter. As discussed by Ke (2019, 2021), it may be the case that different subcomponents of

³⁸Features are defined in Section 4.6, labels in Section 4.7.

I-language employ different search algorithms. There is no *a priori* reason that this should not be the case, even adhering to TLTB. Different search algorithms may be more appropriate and thus more optimal for different kinds of data. Nevertheless, Ke (2019) proposes that labelling and agree can indeed be unified under MS, following the conjecture of Chomsky (2013, 2015). I maintain this proposal here, formalising a unified MS algorithm, hypothesised to form the basis of *Label* and *Agree*, to be defined in Section 4.7 and Section 4.8, respectively.

4.5.1 Accessibility and trace-invisibility

Before discussing the algorithm itself, it must be established what elements are actually available to serve as labels. The notion of *W*-accessibility was previously defined in Definition 13 and Definition 21 to account for this, however, it will need to be revised a final time.

It was established in Section 3 that lower copies of moved elements are invisible to the labelling algorithm. This allows labelling to derive many cases of movement, as per Chomsky (2013) and subsequent work on labelling, in particular within GDA (Moro & Roberts, 2020). Optimally, if lower copies are invisible to labelling, they should be invisible to all operations.

Definition 24. An SO X is *W-accessible* at stage $\langle W_i, T_i \rangle$ iff

- (i) W contains X ,
- (ii) T does *not* contain X , and
- (iii) X is at position P_1 and there is no occurrence of X at a position P_2 such that X_{P_1} is a term of the sister of X_{P_2} .

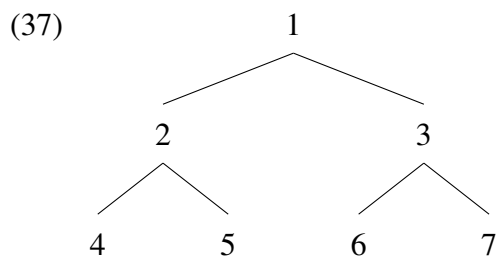
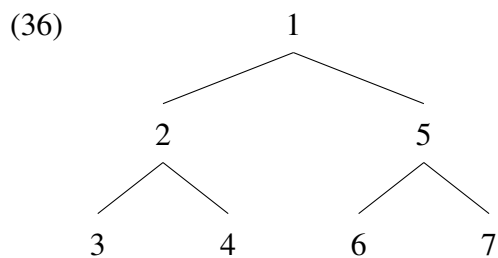
Condition (iii) is necessarily somewhat stipulative. Ideally, it would be possible to derive trace-invisibility from something more fundamental; I leave this for future work. In this regard, note that (iii) does conceal the c-command relation (cf. footnote 24). This undoubtedly carries some significance in relation to the operation of MS.³⁹

4.5.2 DFS or BFS?

The ‘minimal’ in MS comes from the “previously implicitly assumed but unnoted” property that “Minimal Search terminates whenever a target is found” (Ke, 2021, p. 3). In other words, there

³⁹Note that the output of *FormCopy* could be used to find the occurrences of X . Since I have not formalised *FormCopy*, I leave this possibility to future analysis, cf. Section 4.2.

can only be one candidate target found by MS, so no comparisons between candidate targets are necessary. The question is, then, exactly how this target is reached. Both Ke (2019) and Milway (2021) note the observation from computer science that there are two broad classes of search algorithm: *depth-first* search (DFS) and *breadth-first* search (BFS). DFS prioritises travelling ‘down’ in a tree, corresponding to travelling into more deeply embedded sets, as diagrammed in (36). BFS explores all nodes at one tier before progressing to the next tier, as diagrammed in (37). The numbers represent the order in which nodes are traversed.



Applying these algorithms to structures produced by *Merge* results in a number of issues. Most significantly, *Merge* produces sets, with no linear order, not trees, as diagrammed above, which are encoded with linear order. As illustrated, both DFS and BFS make use of linear order to determine the search order. As a result, both would be catastrophic if used in the context of labelling. For example, take the trees above: assume that LA is tasked with labelling the node indicated with 1. Assume further that the lowest tier consists solely of heads. In (36), 3 will serve as the label, in (37), 4 will serve as the label. (Note that, in this case, both algorithms reach the same node, which may not be the case in a more complex example.) Since the algorithm reaches these nodes first, and since they are heads and thus cannot be searched into, they are immediately returned by the algorithm. In the context of SOs, as opposed to trees, this entails making an arbitrary decision as to which node to choose first, which is clearly empirically unjustified. Further, DFS specifically presents issues. As Ke (2019) notes, it does not respect c-command relations, unlike BFS, which has ingrained a notion of superiority, since it prioritises exploring nodes on the same tier. Instead, DFS primarily makes use of the containment relation.

Additionally, DFS simply derives the wrong results: DFS would entail travelling potentially many levels deep into a structure, when if it just looked at the initial sister it would find a head immediately. DFS just does not capture the kinds of relations found in language, and should be discarded. Despite this, Milway (2021, p. 17) argues that DFS “retains a certain theoretical and aesthetic appeal” and thus should remain under consideration. He notes that some authors, namely Branan and Erlewine (to appear) and Preminger (2019), argue for a DFS-based MS algorithm. However, these proposals require that Merge be defined asymmetrically, making this implausible in a system that uses set-Merge as in Definition 15, eliminating linear order in line with the SMT.⁴⁰

Milway (2021) overcomes the linear order issue by appealing to what he terms ‘Minimal Tiered BFS’ (Milway, 2021, p. 15). In such a system, all SOs on a particular tier of the BFS algorithm are considered part of the same set, and are accessed simultaneously in order to identify the target. In the course of the algorithm, structure is thus ignored. I adopt Tiered BFS here.

4.5.3 Domain and target

Before presenting the formalisation of MS in Section 4.5.4, the parameters to the algorithm need to be established. As pointed out by Ke (2019), the search algorithm (SA) is only one aspect of MS. In order to operate, SA needs two further elements a *search domain* (SD) and a *search target* (ST). In order to unify MS, SA and SD may be provided as parameters. This enables SA to be highly flexible—which is a highly desirable outcome, since SA is presumed to be a third factor. Indeed, the parameterisation of SA quite neatly demonstrates the interaction between factors discussed in Section 1.1, with the first factor specifying ST. Indeed, I would suggest that the second factor is also incorporated into ST, since it presumably relies on lexical specification to determine whether the target has been found, and the lexicon is the source of syntactic variation (see Section 1.5).

There must necessarily be constraints on the parameters for SA. Ke (2019, p. 44) states that SD consists of sets and ST features. In the present formalisation, SD shall be sets of SOs in particular. ST, however, I assume to be more complex. The primary reason for this is that ST being a specific feature or set of features seems appropriate for Agree, but not for Label. In the latter case, any set of features can be considered a label—what is important is the structure

⁴⁰Milway (2021, p. 17) claims that Ke’s (2019) algorithm is “parallelized DFS”, although this directly contradicts Ke’s (2019, p. 48) own assertion that “[t]he search algorithm in the definition of minimal search [] is breadth-first”.

more generally. If SD for Label is effectively ‘anything goes’ (as long as its something that bears features, i.e. a non-root head), then a problem arises. Namely, feature sharing arrangements, discussed in Section 3 as being crucial to modern labelling theory, are impossible under such a theory. Indeed, this is noted by Ke (2019), who proposes that, in scenarios where two heads are found simultaneously, it is the pair of heads that serves as the label, rather than their intersecting features. This is effectively justified by appealing to Takita (2020), who argues that labelling is required only by SM, not C-I. This is a radical departure from the assumptions of the preceding discussion, and so ought to be avoided. A second, more general reason to suggest a more complex ST is that it allows SA to be much more flexible. As SA is presumed to be a domain-general third factor, this is a desirable outcome.

The question is then of how to formalise a generalised ST. The solution I adopt is to allow ST to be a function definition, which takes an SD as its domain. In the process of search, SA applies the function at each tier. The codomain of the ST function is then \emptyset plus the range of possible matched items, whether this be individual (sets of) features (as in the case of Agree and feature-sharing label) or entire heads (as may be also the case for Label). ST also determines the output of the search procedure itself, as SA returns whatever is matched, which is the output of ST. This enables what Shim (2018, p. 23) terms “comparison search”, but without necessarily imposing a greater computational burden as he argues.

4.5.4 Defining minimal search

It is now possible to formally define our SA.

Definition 25. For SD δ , a set of SOs, and ST τ , a unary function:

- (i) If $\delta = \emptyset$, $\Sigma(\delta, \tau) = \emptyset$,
- (ii) Else if $\tau(X) \neq \emptyset$, $X \in \delta$, then $\Sigma(\delta, \tau) = \tau(X)$,
- (iii) Else, $\Sigma(\delta, \tau) = \Sigma(\bigcup\{X \in \delta : X \text{ is W-accessible and } X \notin LEX\}, \tau)$.

Conditions (ii) and (iii) represent the crucial recursive workings of the operation. This part of the algorithm first checks if the current SD contains a matching element, and if it does, it returns whatever the matching function itself returns. Then comes the recursive step: if no matching element is found, perform the algorithm again using the arbitrary union of all SOs immediately

contained by SD that are not lexical items. Condition (i) is a fallback that allows search to fail entirely. This foreseeably results in the derivation crashing at the interfaces in most cases, although this is not *a priori* necessary. For instance, adjuncts might be unlabelled (see Blümel, 2017).

As planned, this algorithm combines the tripartite architecture formalised by Ke (2019) with Milway’s (2021) formal framework, inherited from C&S. In particular, the arbitrary union definition of tiers is taken from Milway (2021, p. 16) as is the recursive operation of the algorithm. Definition 25 also incorporates the novel flexibility of τ . A further benefit of this latter point is that the output of Σ does not have to be arbitrarily defined for each instantiation of the function. Ke (2019) has to do this in his formalisation, because (a) he does not parameterise Σ (but rather defines it as a tuple), and (b) he does not allow ST to be a function. In my case, a single call to Σ contains all the information needed to determine its precise operation.

4.6 Features

A full formalisation of features and agreement is beyond the scope of this thesis. However, it is necessary to have some conception of features, as these are what serve as labels, treated in Section 4.7. The nature of features is also inimicly tied to the nature of agreement, discussed in Section 4.8. A relatively neutral approach to the nature of features will be taken based on the system formalised by Adger (2006, 2010), in order to match up best with the labelling theories that have been discussed and the approach that will be adopted in Section 4.7.

Adger (2006, 2010) proposes a formal hierarchy of feature systems, recapitulated in (38).

(38) *Feature system hierarchy*

- (a) **Privative**: “atomic features may be present or absent, but have no other properties” (Adger, 2010, p. 187).
- (b) **Privative with interpretability**: privative features may be prefixed with *u* to indicate uninterpretability, enabling *checking* relations to be formed between features (cf. Chomsky, 1995b).
- (c) **Binary attribute-value**: features are ordered pairs $\langle Att, Val \rangle$ where *Att* is drawn from a set of attributes, and *Val* is +, – or empty.

- (d) **Multi-valent attribute-value:** features are attribute-value pairs, values are drawn from a larger set of values.
- (e) **Recursive attribute-value:** features are attribute-value pairs, values may themselves be features.

For reasons there is no room to discuss, Adger (2006, 2010) decisively argues that a multi-valent attribute-value feature system (38 d) is the option most compatible with Minimalism as generally practised. A version of this system is adopted by Milway (2021), who makes the simplifying assumption that values can be encoded as integers. This eliminates the need for much of Adger’s (2010) formal apparatus. Whilst it is safe to say that this apparatus would be required in a fully-fledged theory of features, especially with respect to interface interpretation, I will also adopt a simplifying assumption. Instead of using integers to symbolise feature values, I will use an equivalent set of atomic symbols, more similarly to Adger (2006, 2010). I will also use \emptyset as a notational convention to indicate the lack of value (as done by Adger, 2010), as a clearer alternative to blank space.

Another simplifying assumption is the adoption of the feature sets defined in Definition 1. Within this formalisation, I will assume that all possible features, i.e. all possible attribute-value pairs, are defined in UG. This greatly limits the power of the system, but is a necessary simplifying assumption to avoid too much digression. Naturally, the formalisation present here can easily be extended to accomodate a more fully-fledged feature theory of an equivalent level of complexity to the attribute-value system formalised by Adger (2006, 2010). This being said, Adger (2010) makes some assumptions that I intentionally do not assume here, since he adopts a feature-driven system, akin to ‘Triggered-Merge’ in C&S, which is in opposition to the (free) Merge-driven system I have adopted here (cf. Section 1.4). Thus, the feature system formalised here is necessarily excessively simple, which will have bearing on potential empirical consequences that there will unfortunately not be room to discuss.

With this in place, I adopt the following definitions.

Definition 26. *Att* is the set of feature *attributes*.

Definition 27. *Val* is the set of feature *values* $\{+, -, \dots\}$.

Definition 28. A *feature* f is a pair $\langle a, v \rangle$ where $a \in Att$ and $v \in Val$. v may be empty; if this is the case f is considered *unvalued*.

Definition 29. A feature f is *interpretable* iff $f \in F_{SEM}$. A feature is *uninterpretable* iff $f \in F_{NS}$.

Note that this definition allows dissociation between feature value and feature interpretability, which is employed by some feature theories, notably by Pesetsky and Torrego (2007).

It will also be useful to define a simple function *Match*, which determines if features are matching for attribute, but not value.

Definition 30. For two features $f_1 = \langle Att_1, Val_1 \rangle, f_2 = \langle Att_2, Val_2 \rangle$,

- (i) if $f_1 = f_2$, $Match(f_1, f_2) = \emptyset$,
- (ii) else if $Att_1 = Att_2$ and one of Val_1, Val_2 is empty, $Match(f_1, f_2) = \{f_1, f_2\}$ or $\{f_2, f_1\}$, such that the valued feature is first.⁴¹
- (iii) else, $Match(f_1, f_2) = \emptyset$.

This definition can be developed into a function *MatchLI*, which returns the set of all sets/pairs of features that satisfy *Match* between two LIs.

Definition 31. For two LIs X and Y , let the set $S = X \times Y$. For all $s \in S$, $MatchLI(X, Y) = \{Match(s) : Match(s) \neq \emptyset\}$.

4.7 Defining *Label*

Finally, the machinery is in place to enable us to define *Label*. Conceivably, with Definition 25 in place, all that is required is to define the SD δ and the ST τ . Establishing δ appears simple: it is the SO to be labelled itself. τ is more tricky. As discussed in Section 4.5, feature-sharing complicates the picture.

Following the discussion in Section 3.2, however, entails that we must also consider how exactly *Label* is applied within derivations. Namely, there are two broad classes of possible labelling processes, as discussed in Section 3.2: labelling can be bottom-up, or top-down. In Section 3.2, I outline an informal definition of bottom-up *Label* using GP. Since labelling is usually considered top-down, and since bottom-up operation is ruled out by Adger and Roberts

⁴¹This condition is totally arbitrary, and is adopted solely to maintain the assumption that shared labels are pairs as opposed to sets.

(to appear) and Ke (2019), I elect to formalise only top-down *Label*, *LabelTD*, in this section. In Section 4.7.2, I note how one could go about formalising bottom-up *Label*, *LabelBU*.

The following helper function will prove useful for determining the lexical items within a set of SOs.

Definition 32. For a set of SOs δ , $Heads(\delta) = \{X \in \delta : X \in LEX \text{ and for } X, SYN \neq \emptyset\}$.

The condition that a head must have syntactic features excludes L-roots from providing labels, and generally participating in syntactic relations. Ideally, this would be derived from more fundamental principles; this is reserved for future work.

4.7.1 Top-down labelling

Definition 33. For set of SOs δ ,

- (i) if $\delta = \emptyset$, then $\tau_{TD}(\delta) = \emptyset$,
- (ii) else if $Heads(\delta) = \{X\}$, for X an SO, then $\tau_{TD}(\delta) = \{X\}$,
- (iii) else if for any set of LIs $S = \{X, Y\} \in [Heads(\delta)]^2$ such that $MatchLI(S) \neq \emptyset$,
 $\tau_{TD}(\delta) = MatchLI(S)$,
- (iv) else $\tau_{TD}(\delta) = \emptyset$.

Condition (ii) is the simple case where, at a particular tier, if there is only one head, this is chosen as the label. Condition (iii) enables feature-sharing where there are multiple heads that match features. Note that if there are multiple heads but they do not match features, the function will return \emptyset . Hence, encoded in this definition is the proposal, implicit in Chomsky (2013), that the feature pair in a feature-sharing arrangement is a valued-unvalued combination.

Recursive MS can thus be defined simply.

Definition 34. For an SO X , $LabelTD(X) = \Sigma(X, \tau_{TD})$.

Top-down labelling requires a complex τ in order to account for feature sharing. Interestingly, it does not impose any restraint on its operation: since τ searches only for heads, it is not dependent on any previous computation. It can thus apply totally freely, at the expense of greater time complexity, since search needs to find the head on every occasion, even when this might be deep within a nested symmetric structure.

Despite this general appeal, my formalisation of top-down labelling has the same fatal flaw as is implicit in Ke’s (2019) formalisation: it fails in $\{XP, YP\}$ structures where the heads of each phrase are differentially nested. For instance, if X is less deeply embedded than Y , X will be selected as the head. Since Nakashima (2021) claims that this differential level of embedding does actually play such a role in labelling (in the form of his ‘Symmetry Condition on Labelling’), I leave the option of this formalisation open. However, I believe that bottom-up labelling is actually more in line with what is generally assumed in the Minimalist literature, and I proceed to formalise this option in the next subsection, replacing Definition 33 and Definition 34.

Note also that it is possible to derive a crucial conclusion of Chomsky (2015) as a theorem, namely the fact that L-roots never provide labels.

Theorem 1. *For the SO $\alpha = \{X, R\}$, for any SO X and an L-root R , $Label(\alpha) = Label(X)$.*

Another important case can be derived.

Theorem 2. *For the SO $\alpha = \{X, YP\}$, for any $X \in LEX$ such that X is not an L-root, $Label(\alpha) = Label(X)$.*

4.7.2 Bottom-up labelling

Bottom-up labelling behaves quite differently. In order to accommodate the fact that accessibility as defined in Definition 24 is sensitive to higher structure, bottom-up labelling needs to retain two kinds of memories. One, as discussed in Section 3.2, is the ‘goldfish’ memory of the labels assigned at the previous level. The other is a set of variables that are incrementally assigned as the labelling algorithm traverses up the structure, encountering symmetric $\{XP, YP\}$ structures that cannot be labelled by feature sharing. These must be assigned an indeterminate label α , which can be resolved upon movement, since at this point the lower occurrence is no longer accessible by trace invisibility. This then perlocates through the label ledger.

Labels, on this view, are not constructed on the fly. Rather, they are assembled from the ground up. Generated labels can then be placed in a ledger—similarly to transferred structures, and similarly to *Agree* (see Section 4.8). This would require a redefinition of derivation, which I do not offer here.

What this entails is that ST for *LabelBU* will be substantively simpler, but SD will be more complex, as a result of needing access to labels applied at the previous stage of derivation.

4.7.3 Comparison

In terms of optimality, there is a final comparison to be made between *LabelTD* and *LabelBU*.

LabelTD allows the label of an element to be identified at any time. If the algorithm fails, a label cannot be assigned. This presumably triggers GDA movement. *LabelBU* operates purely cyclically but requires a memory. This memory serves to reduce the overall amount of search, since only four elements ever need to be checked, namely the two labels and two SOs of the previous tier. This violates MinCCD for the sake of MinSearch, trading time complexity for space complexity. It is less clear how GDA works with *LabelBU*. Without further specifying the algorithms, a more precise comparison is not available.

4.8 Agree

The fundamental purpose of syntax is to eliminate uninterpretable features that are present in lexical items so that they can be interpreted at the interfaces (cf. Section 1.4.3). Relations between features within separate SOs encode syntactic dependencies (cf. Adger, 2010). All such dependencies need to be evaluated within the syntax in order to be legible. In many cases, application of *Merge* suffices to satisfy feature dependencies. For instance, take subcategorisation: as mentioned in Section 2 subcategorisation was one of the motivations for the development of labelling theory because of the periscope property, whereby the selecting head need only look at the label of the selected structure. The original formulation of a separate, non-Merge relation by Chomsky (2000, 2001) was motivated by two constructions in particular: English existential constructions and long-distance dependencies in many languages.

Milway (2021) formalises Agree within the C&S framework. This could reasonably trivially be adapted into the present framework. However, I would suggest a further revision: namely, that Agree operates on the labels provided by *Label* (either top-down or bottom up) rather than raw SOs. Since Agree is not the focus of this thesis, a full formalisation will not be provided. It seems reasonable to assume that there is a definable $Agree(N, R) = R'$ function, which maps a set of labels N and a ledger R to a new ledger R' .

4.9 Key conclusions

This subsection summarises some of the issues that arise from the formalisation as present.

4.9.1 Computational and substantive optimality

There are a number of properties of this theory that are in line with MaxTP and TLTB. Most importantly, this theory maintains MY as a theorem.

Theorem 3 (MY). *For consecutive stages S_i and S_{i+1} , S_{i+1} may only have up to one more W -accessible object than S_i .*

The NTC and the Extension Condition can also both be derived as theorems as done by C&S (p.p. 58-59). Note, however, that both C&S and Milway (2021) ultimately propose theories that violate NTC. They respectively claim that Transfer and Agree require lower structure to be substituted. In my formalisation, I adopt ledgers instead, again trading time complexity for space complexity.

4.9.2 Agree and Label

Crucially, *Agree* and *Label* conspire to form syntactic dependencies throughout constructed SOs. In cases where a label is a valued-unvalued pair $\langle\langle Att_i, Val_i \rangle, \langle Att_i, \emptyset \rangle\rangle$, *Agree* would not need to operate. This clearly identifies the precise nature of the redundancy between *Agree* and *Label*. Namely, it is not the fact that both operations utilise MS that creates the redundancy, since as shown by the formalisation in Section 4.5, they each use different instantiations of SA. Instead, the redundancy is in their outcomes: they both result in creating relations between features.

4.9.3 Label \in UG?

Labelling theory is not yet at the stage where *Label* can be totally dissociated from UG. As shown in Section 4.5, whilst SA is potentially domain-general, its parameters, ST and SD, are narrowly syntactic. As a result, *Label* is required to be included in Definition 1. In line with OM and MM, future work could help determine whether this stipulation could be dropped.

5 Conclusion and future prospects

The formalisation programme instigated by C&S is a novel one, and one fraught with complication as a consequence of the nascence of Minimalist theory and its diverse fragmentation. The formalisation process has thus exposed the fragility of some of the formal concepts underlying Minimalist syntax, particularly as they pertain to labelling. Elements of the formalisation presented in C&S have been majorly revised, for two reasons: (a) to bring the formalisation more in line with recent developments in MP, and (b) to introduce a novel approach to MS and labelling.

As clear from the outset, empirical discussion has not been the focus, in the interest of providing a clear formal grounding, making some steps towards unifying the diversity of labelling theories that abound in the literature. Space constraints have prevented discussion of even simple examples which would demonstrate how the algorithms proposed in Section 4 derive syntactic structures. There is clear scope for future work in this area. Particular pertinent structures which could be analysed in this framework are standard cases of *wh*-movement, English passives, and also the structures given in (39) below.

- (39) (a) Japanese multiple subject constructions (Epstein et al., 2020; Kuno, 1973; Saito, 2016):

Bunmeikoku-ga dansei-ga heikin-zyumyoo-ga mizika-i
civilized.country-NOM male-NOM average-life.span-NOM short-PRS
“It is in civilized countries that male’s average life span is short.”

(example from Saito, 2016, p. 131)

- (b) Verb movement in Romance (Schifano, 2018 and Ian Roberts, p.c.)

*Antoine confond probablement (*confond) le poème*
A. confuse.3SG.PRS probably the poem
“A. is probably confusing the poem.”

(example from Schifano, 2018, p. 63)

Consideration of these will certainly require adjustments to the formalisation. It is hoped that, by making the formal representations and operations that are employed in the literature more explicit, further insight can be gained into the operation of FL on the computational and algorithmic levels.

References

- Adger, D. (2003). *Core syntax: A minimalist approach*. Oxford University Press.
- Adger, D. (2006). Remarks on Minimalist feature theory and Move. *Journal of Linguistics*, 42(3), 663–673.
- Adger, D. (2010). A Minimalist theory of feature structure. In A. Kibort & G. G. Corbett (Eds.), *Features: Perspectives on a key notion in linguistics*. Oxford University Press.
- Adger, D. (2013). *A syntax of substance*. MIT Press.
- Adger, D., & Roberts, I. (to appear). Principles of Universal Grammar and the Minimalist Program. In K. K. Grohmann & E. Leivada (Eds.), *The Cambridge handbook of minimalism*. Cambridge University Press.
- Adger, D., & Svenonius, P. (2011). Features in minimalist syntax. In C. Boeckx (Ed.), *The Oxford handbook of linguistic minimalism* (pp. 27–51). Oxford University Press.
- American Psychological Association. (2020). *Publication manual of the American Psychological Association* (7th ed.).
- Asudeh, A., & Toivonen, I. (2006). Symptomatic imperfections. *Journal of Linguistics*, 42(2), 395–422.
- Avigad, J., Lewis, R. Y., & van Doorn, F. (2017). *Logic and proof*.
- Baker, A. (2016). Simplicity. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Winter 2016). Metaphysics Research Lab, Stanford University.
- Baker, M. C. (2008). The macroparameter in a microparametric world. In T. Biberauer (Ed.), *The limits of syntactic variation* (pp. 351–374). John Benjamins.
- Balari, S., & Lorenzo, G. (2012). *Computational phenotypes: Towards an evolutionary developmental biolinguistics*. Oxford University Press.
- Barrie, M. (2021). Where and why are labels necessary? *Studies in Generative Grammar* (), 31(1), 25–52.
- Bauke, L., & Blümel, A. (2017). Introduction. In L. Bauke & A. Blümel (Eds.), *Labels and roots* (pp. 1–16). De Gruyter Mouton.
- Behme, C. (2014). A ‘Galilean’ science of language. *Journal of Linguistics*, 50(3), 671–704.
- Behme, C. (2015). Is the ontology of biolinguistics coherent? *Language Sciences*, 47, 32–42.

- Berwick, R. C., & Chomsky, N. (2011). The biolinguistic program: The current state of its development. In C. Boeckx & A.-M. Di Sciullo (Eds.), *The biolinguistic enterprise: New perspectives on the evolution and nature of the human language faculty*. Oxford University Press.
- Berwick, R. C., & Chomsky, N. (2016). *Why only us: Language and evolution*. MIT Press.
- Berwick, R. C., & Stabler, E. P. (Eds.). (2019). *Minimalist parsing*. Oxford University Press.
- Biberauer, T. (2019). Factors 2 and 3: Towards a principled approach. *Catalan Journal of Linguistics*, 45–88.
- Biberauer, T., Holmberg, A., Roberts, I., & Sheehan, M. (2010). *Parametric variation: Null subjects in minimalist theory*. Cambridge University Press.
- Blevins, J. (2004). *Evolutionary phonology: The emergence of sound patterns*. Cambridge University Press.
- Blümel, A. (2017). *Symmetry, shared labels and movement in syntax*. De Gruyter.
- Boeckx, C. (2013). Biolinguistics: Facts, fiction, and forecast. *Biolinguistics*, 7, 316–328.
- Boeckx, C. (2014). What Principles and Parameters got wrong. In M. C. Picallo (Ed.), *Linguistic variation in the Minimalist framework* (pp. 155–178). Oxford University Press.
- Borer, H. (2005a). *Structuring sense volume I: In name only*. Oxford University Press.
- Borer, H. (2005b). *Structuring sense volume II: The normal course of events*. Oxford University Press.
- Borer, H. (2013). *Structuring sense volume III: Taking form*. Oxford University Press.
- Bošković, Ž. (1997). *The syntax of nonfinite complementation: An economy approach*. MIT Press.
- Bošković, Ž. (2007a). Agree, phases and intervention effects. *Linguistic Analysis*, 33, 54–96.
- Bošković, Ž. (2007b). On the locality and motivation of Move and Agree: An even more minimal theory. *Linguistic Inquiry*, 38(4), 589–644.
- Bošković, Ž. (2016a). Introduction: On labels. *The Linguistic Review*, 33(1), 1–16.
- Bošković, Ž. (2016b). What is sent to spell-out is phases, not phasal complements. *Linguistica*, 56(1), 25–66.
- Branan, K., & Erlewine, M. Y. (to appear). Locality and (minimal) search. In K. K. Grohmann & E. Leivada (Eds.), *Cambridge handbook of minimalism*. Cambridge University Press.

- Brody, M. (2000). Mirror theory: Syntactic representation in perfect syntax. *Linguistic Inquiry*, 31(1), 29–56.
- Burrows, E. R. (2022). *Passives in Extended Dynamic Antisymmetry* (BA dissertation). University of Cambridge. Cambridge, UK.
- Carlson, J. O. E. (2010). *On linguistic asymmetry: An investigation of the reality and significance of skewed patterns* (MPhil Thesis). University of Cambridge.
- Chametzky, R. A. (1996). *A theory of phrase markers and the extended base*. SUNY Press.
- Chomsky, N. (1951). *The morphophonemics of modern Hebrew* (Master's thesis). University of Pennsylvania. Philadelphia, PA.
- Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory*, 2(3), 113–124.
- Chomsky, N. (1957). *Syntactic structures*. Mouton.
- Chomsky, N. (1959). Verbal behavior. *Language*, 35(1), 26–58.
- Chomsky, N. (1964). *Current issues in linguistic theory*. Mouton.
- Chomsky, N. (1965). *Aspects of the theory of syntax*. MIT Press.
- Chomsky, N. (1966). *Cartesian linguistics: A chapter in the history of rationalist thought*. Harper and Row.
- Chomsky, N. (1970). Remarks on nominalization. In R. A. Jacobs & P. S. Rosenbaum (Eds.), *Readings in English transformational grammar* (pp. 184–221). Ginn.
- Chomsky, N. (1973). Conditions on transformations. In S. R. Anderson & P. Kiparsky (Eds.), *Festschrift for Morris Halle*. Holt, Rinehart and Winston.
- Chomsky, N. (1975). *The logical structure of linguistic theory*. Plenum.
- Chomsky, N. (1976). Constraints on rules of grammar. *Linguistic Analysis*, 2, 303–351.
- Chomsky, N. (1977). On wh-movement. In P. W. Culicover, T. Wasow & A. Akmajian (Eds.), *Formal syntax* (pp. 71–133). Academic Press.
- Chomsky, N. (1980). *Rules and representations*. Blackwell.
- Chomsky, N. (1981). *Lectures on Government and Binding*. Foris.
- Chomsky, N. (1986a). *Barriers*. MIT Press.
- Chomsky, N. (1986b). *Knowledge of language: Its nature, origin, and use*. Praeger.

- Chomsky, N. (1993). A minimalist program for linguistic theory. In K. Hale & S. J. Keyser (Eds.), *The view from building 20: Essays in linguistics in honor of Sylvain Bromberger* (pp. 1–52). MIT Press.
- Chomsky, N. (1994a). Bare Phrase Structure. *MIT Occasional Papers in Linguistics*, 5.
- Chomsky, N. (1994b). Naturalism and dualism in the study of language and mind. *International Journal of Philosophical Studies*, 2(2), 181–209.
- Chomsky, N. (1995a). Language and nature. *Mind*, 104(413), 1–61.
- Chomsky, N. (1995b). *The Minimalist Program*. MIT Press.
- Chomsky, N. (2000). Minimalist inquiries: The framework. In R. Martin, D. Michaels & J. Uriagereka (Eds.), *Step by step: Essays on minimalist syntax in honor of Howard Lasnik* (pp. 89–156). MIT Press.
- Chomsky, N. (2001). Derivation by phase. In M. J. Kenstowicz (Ed.), *Ken Hale: A life in language* (pp. 1–52). MIT Press.
- Chomsky, N. (2003). Internalist explorations. In Martin Hahn & Bjørn Ramberg (Eds.), *Reflections and replies: Essays on the philosophy of Tyler Burge* (pp. 259–288). MIT Press.
- Chomsky, N. (2004). Beyond explanatory adequacy. In A. Belletti (Ed.), *Structures and beyond* (pp. 104–131). Oxford University Press.
- Chomsky, N. (2005). Three factors in language design. *Linguistic Inquiry*, 36(1), 1–22.
- Chomsky, N. (2006). *Language and mind: Third edition*. Cambridge University Press.
- Chomsky, N. (2007). Approaching UG from below. In U. Sauerland & H.-M. Gärtner (Eds.), *Interfaces + Recursion = Language?* (pp. 1–30). De Gruyter Mouton.
- Chomsky, N. (2008). On phases. In R. Freidin, C. P. Otero & M. L. Zubizarrete (Eds.), *Foundational issues in linguistic theory: Essays in honor of Jean-Roger Vergnaud*. MIT Press.
- Chomsky, N. (2009). Conclusion. In M. Piattelli-Palmarini, P. Salaburu & J. Uriagereka (Eds.), *Of minds and language: A dialogue with Noam Chomsky in the Basque Country* (pp. 379–401). Oxford University Press.
- Chomsky, N. (2013). Problems of projection. *Lingua*, 130, 33–49.
- Chomsky, N. (2014). Minimal recursion: Exploring the prospects. In T. Roeper & M. Speas (Eds.), *Recursion: Complexity in cognition* (pp. 1–15). Springer.

- Chomsky, N. (2015). Problems of projection: Extensions. In E. Di Domenico, C. Hamann & S. Matteini (Eds.), *Structures, strategies and beyond: Studies in honour of Adriana Belletti* (pp. 29–42). John Benjamins.
- Chomsky, N. (2016). *What kind of creatures are we?* Columbia University Press.
- Chomsky, N. (2017). The Galilean challenge. *Inference*, 3(1).
- Chomsky, N. (2019a). *The UCLA lectures* (Presentation recording).
- Chomsky, N. (2019b). Some puzzling foundational issues: The reading program. *Catalan Journal of Linguistics*, (0), 263–285.
- Chomsky, N. (2020a). Puzzles about phases. In L. Franco & P. Lorusso (Eds.), *Linguistic variation: Structure and interpretation* (pp. 163–168). De Gruyter Mouton.
- Chomsky, N. (2020b). *The UCLA lectures* [Unpublished transcript].
- Chomsky, N. (2021a). Minimalism: Where are we now, and where can we hope to go. *Gengo Kenkyu*, 160, 1–41.
- Chomsky, N. (2021b). Reflections. In N. Allott, T. Lohndal & G. Rey (Eds.), *A companion to Chomsky* (pp. 583–593). Wiley-Blackwell.
- Chomsky, N., Gallego, Á. J., & Ott, D. (2019). Generative grammar and the faculty of language: Insights, questions, and challenges. *Catalan Journal of Linguistics*, 229–261.
- Chomsky, N., & Halle, M. (1968). *The sound pattern of English*. MIT Press.
- Chomsky, N., & Lasnik, H. (2015). The theory of Principles and Parameters. In *The Minimalist Program* (20th Anniversary Edition, pp. 11–116). MIT Press. (Original work published 1993)
- Chomsky, N., & Miller, G. A. (1963). Introduction to the formal analysis of natural languages. In R. D. Luce, R. R. Bush & E. Galanter (Eds.), *Handbook of mathematical psychology* (pp. 269–321). John Wiley & Sons.
- Citko, B. (2011a). *Symmetry in syntax: Merge, Move and labels*. Cambridge University Press.
- Citko, B. (2011b). Multidominance. In C. Boeckx (Ed.), *The Oxford handbook of linguistic minimalism* (pp. 119–142). Oxford University Press.
- Cohen, M. (2008). *On reality as a mathematical structure* (Doctoral dissertation). Ben Gurion University of the Negev. Beersheba, Israel.
- Collins, C. (2002). Eliminating labels. In S. D. Epstein & T. D. Seely (Eds.), *Derivation and explanation in the Minimalist Program* (pp. 42–64). Blackwell.

- Collins, C., & Groat, E. M. (2018). *Distinguishing copies and repetitions* [Unpublished manuscript].
- Collins, C., & Seely, T. D. (2020). *Labeling without labels* [Unpublished manuscript].
- Collins, C., & Stabler, E. (2016). A formalization of minimalist syntax. *Syntax*, 19(1), 43–78.
- Di Sciullo, A.-M., & Boeckx, C. (2011). Introduction: Contours of the biolinguistic research agenda. In C. Boeckx & A.-M. Di Sciullo (Eds.), *The biolinguistic enterprise: New perspectives on the evolution and nature of the human language faculty* (pp. 1–16). Oxford University Press.
- Embick, D., & Poeppel, D. (2015). Towards a computational(ist) neurobiology of language: Correlational, integrated and explanatory neurolinguistics. *Language, Cognition and Neuroscience*, 30(4), 357–366.
- Enderton, H. B. (1977). *Elements of set theory*. Academic Press.
- Epstein, S. D. (1999). Un-principled syntax: The derivation of syntactic relations. In S. D. Epstein & N. Hornstein (Eds.), *Working minimalism* (pp. 317–345). MIT Press.
- Epstein, S. D., Groat, E. M., Kawashima, R., & Hisatsugu, K. (1998). *A derivational approach to syntactic relations*. Oxford University Press.
- Epstein, S. D., Kitahara, H., & Seely, T. D. (2020). Unifying labeling under minimal search in 'single-' and 'multiple-specifier' configurations. *Coyote Papers*, 22.
- Evers, A., & van Kampen, J. (2008). Parameter setting and input redirection. In T. Biberauer (Ed.), *The limits of syntactic variation*. John Benjamins.
- Franks, S., & Bošković, Ž. (2001). An argument for multiple Spell-Out. *Linguistic Inquiry*, 32(1), 174–183.
- Galilei, G. (1967). *Dialogo sopra i due massimi sistemi del mondo* [Dialogue concerning the two chief world systems] (S. Drake, Trans.; 2nd ed.). University of California Press. (Original work published 1632)
- Gallego, Á. J. (2011). Parameters. In C. Boeckx (Ed.), *The Oxford handbook of linguistic minimalism* (pp. 523–550). Oxford University Press.
- Gallistel, C. R. (2001). Mental representations, Psychology of. In N. J. Smelser & P. B. Baltes (Eds.), *International Encyclopedia of the Social & Behavioral Sciences* (pp. 9691–9695). Pergamon.

- Gallistel, C. R., & King, A. P. (2010). *Memory and the computational brain: Why cognitive science will transform neuroscience*. Wiley-Blackwell.
- Gärtner, H.-M. (2022). Copies from “Standard Set Theory”? A note on the foundations of minimalist syntax in reaction to Chomsky, Gallego and Ott (2019). *Journal of Logic, Language and Information*, 31(1), 129–135.
- Gee, P., Stephenson, D., & Wright, D. E. (1994). Temporal discrimination learning of operant feeding in goldfish (*carassius auratus*). *Journal of the Experimental Analysis of Behavior*, 62(1), 1–13.
- Graf, T. (2013). *Local and transderivational constraints in syntax and semantics* (Doctoral dissertation). UCLA. Los Angeles, CA.
- Haegeman, L. (1996). Verb second, the split CP and null subjects in early Dutch finite clauses. *Geneva Generative Papers*, 4(2), 5–38.
- Hale, M., & Reiss, C. (2008). *The phonological enterprise*. Oxford University Press.
- Halle, M., & Marantz, A. (1993). Distributed morphology and the pieces of inflection. In K. Hale & S. J. Keyser (Eds.), *The view from Building 20* (pp. 111–176). MIT Press.
- Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2013). *Introduction to automata theory, languages, and computation* (Pearson new international edition). Pearson Education.
- Hornstein, N. (2021). The Extended Merge Hypothesis and the Fundamental Principle of Grammar. *Philosophies*, 6(4), 89–102.
- Hornstein, N., Nunes, J., & Grohmann, K. K. (2005). *Understanding Minimalism*. Cambridge University Press.
- Jackendoff, R. (1977). *X-bar syntax: A study of phrase structure*. MIT Press.
- Kaplansky, I. (1972). *Set theory and metric spaces*. Allyn and Bacon.
- Kayne, R. S. (1994). *The antisymmetry of syntax*. MIT Press.
- Ke, H. (2019). *The syntax, semantics and processing of agreement and binding grammatical illusions* (Doctoral dissertation). University of Michigan.
- Ke, H. (2021). *Minimal Search in Agree and labeling* [Unpublished manuscript].
- Kitahara, H. (1997). *Elementary operations and optimal derivations*. MIT Press.
- Klima, E. S. (1964). Negation in English. In J. A. Fodor & J. J. Katz (Eds.), *The structure of language: Readings in the philosophy of language*. Prentice-Hall.

- Kratzer, A. (1996). Severing the external argument from its verb. In J. Rooryck & L. Zaring (Eds.), *Phrase structure and the lexicon* (pp. 109–137). Kluwer.
- Kuno, S. (1973). *The structure of the Japanese language*. MIT Press.
- Laenzlinger, C. (1998). *Comparative studies in word order variation: Adverbs, pronouns, and clause structure in Romance and Germanic*. John Benjamins.
- Larson, R. K. (1988). On the double object construction. *Linguistic Inquiry*, 19(3), 335–391.
- Lenneberg, E. H. (1967). *Biological foundations of language*. John Wiley & Sons.
- Lewontin, R. C. (2000). *The triple helix: Gene, organism, and environment*. Harvard University Press.
- Marantz, A. (1997). No escape from syntax: Don't try morphological analysis in the privacy of your own lexicon. *University of Pennsylvania working papers in linguistics*, 4(2), 14.
- Marantz, A. (2013). Verbal argument structure: Events and participants. *Lingua*, 130, 152–168.
- Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. Freeman.
- Marr, D., & Poggio, T. (1976). *From understanding computation to understanding neural circuitry* (A.I. Memo AIM-357). Artificial Intelligence Laboratory, Massachusetts Institute of Technology. Cambridge, MA.
- Martins, P. T., & Boeckx, C. (2016). What we talk about when we talk about biolinguistics. *Linguistics Vanguard*, 2(1), 1–15.
- Miller, G. A., & Chomsky, N. (1963). Finitary models of language users. In D. Luce, R. R. Bush & E. Galanter (Eds.), *Handbook of Mathematical Psychology* (pp. 419–492). John Wiley & Sons.
- Milway, D. (2021). *A formalization of Agree as a derivational operation*.
- Mobbs, I. (2008). *'Functionalism', the design of the language faculty, and (disharmonic) typology* (MPhil thesis). University of Cambridge. Cambridge, UK.
- Mobbs, I. (2015). *Minimalism and the design of the language faculty* (Doctoral dissertation). University of Cambridge. Cambridge, UK.
- Morin, E., & Piattelli-Palmarini, M. (1974). *L'unité de l'homme: Invariants biologique et universaux culturels*. Éditions du Seuil.
- Moro, A. (2000). *Dynamic antisymmetry*. MIT Press.

- Moro, A., & Roberts, I. (2020). *Unstable structures and Generalized Dynamic Antisymmetry* [Unpublished manuscript].
- Muysken, P. (1982). Parametrizing the notion ‘head’. *Journal of Linguistic Research*, 2, 57–75.
- Nakashima, T. (2021). *The Symmetry Condition on Labeling* [Unpublished manuscript].
- Pesetsky, D., & Torrego, E. (2007). The syntax of valuation and the interpretability of features. In S. Karimi, V. Samiiian & W. K. Wilkins (Eds.), *Phrasal and clausal architecture: Syntactic derivation and interpretation. In honor of Joseph E. Emonds* (pp. 262–294). John Benjamins.
- Poepfel, D., & Embick, D. (2005). Defining the relation between linguistics and neuroscience. In A. Cutler (Ed.), *Twenty-First century psycholinguistics: Four cornerstones* (pp. 103–118). Taylor & Francis Group.
- Post, E. L. (1944). Recursively enumerable sets of positive integers and their decision problems. *Bulletin of the American Mathematical Society*, 50(5), 284–316.
- Post, E. L. (1947). Recursive unsolvability of a problem of Thue. *The Journal of Symbolic Logic*, 12(1), 1–11.
- Preminger, O. (2019). What the PCC tells us about “abstract” agreement, head movement, and locality. *Glossa: a journal of general linguistics*, 4(1).
- Pylyshyn, Z. W. (1984). *Computation and cognition: Toward a foundation for cognitive science*. MIT Press.
- Quine, W. V. (1995). Whitehead and the rise of modern logic. In *Selected logic papers* (pp. 1–36). Harvard University Press. (Original work published 1941)
- Radford, A. (2004). *Minimalist syntax: Exploring the structure of English*. Cambridge University Press.
- Reinhart, T. (1976). *The syntactic domain of anaphora* (Doctoral dissertation). MIT. Cambridge, MA.
- Richards, M. D. (2007). On feature inheritance: An argument from the Phase Impenetrability Condition. *Linguistic Inquiry*, 38(3), 563–572.
- Richards, N. (2001). *Movement in language: Interactions and architectures*. Oxford University Press.
- Rizzi, L. (1997). The fine structure of the left periphery. In L. Haegeman (Ed.), *Elements of grammar: Handbook in generative syntax* (pp. 281–337). Kluwer.

- Rizzi, L. (2015). Cartography, criteria, and labeling. In U. Shlonsky (Ed.), *The cartography of syntactic structures: Vol. 10. beyond functional sequence* (pp. 314–338). Oxford University Press.
- Rizzi, L. (2016). Labeling, maximality and the head-phrase distinction. *The Linguistic Review*, 33(1), 103–127.
- Roberts, I. (2001). Head movement. In M. R. Baltin & C. Collins (Eds.), *The handbook of contemporary syntactic theory* (pp. 111–147). Blackwell.
- Roberts, I. (2004). The C-system in Brythonic Celtic languages, V2, and the EPP. In L. Rizzi (Ed.), *The structure of CP and IP: The cartography of syntactic structures* (pp. 297–328). Oxford University Press.
- Roberts, I. (2019). *Parameter hierarchies and Universal Grammar*. Oxford University Press.
- Saito, M. (2016). (A) Case for labeling: Labeling in languages without ϕ -feature agreement. *The Linguistic Review*, 33(1), 129–175.
- Schifano, N. (2018). *Verb movement in Romance: A comparative study*. Oxford University Press.
- Seely, T. D. (2006). Merge, derivational c-command, and subcategorization in a label-free syntax. In C. Boeckx (Ed.), *Minimalist essays* (pp. 182–218). John Benjamins.
- Seely, T. D. (2021). *On the form & function of MERGE* (Online presentation). Arizona State University.
- Sheehan, M., Biberauer, T., Roberts, I., & Holmberg, A. (2017). *The final-over-final condition: A syntactic universal*. MIT Press.
- Shim, J.-Y. (2018). $\langle \phi, \varphi \rangle$ -less Labeling. *Language Research*, 54(1), 23–39.
- Song, C. (2019). *On the formal flexibility of syntactic categories* (Thesis). University of Cambridge.
- Sportiche, D., Koopman, H., & Stabler, E. (2014). *An introduction to syntactic analysis and theory*. Wiley-Blackwell.
- Sprouse, J., Yankama, B., Indurkha, S., Fong, S., & Berwick, R. C. (2018). Colorless green ideas do sleep furiously: Gradient acceptability and the nature of the grammar. *The Linguistic Review*, 35(3), 575–599.
- Stabler, E. (1997). Derivational minimalism. In C. Retoré (Ed.), *Logical Aspects of Computational Linguistics* (pp. 68–95). Springer.

- Starke, M. (2004). On the inexistence of specifiers and the nature of heads. In A. Belletti (Ed.), *Structures and beyond* (pp. 252–268). Oxford University Press.
- Stockwell, R. (2015). *Emergent Features: Insights from imperatives* (MPhil Thesis). University of Cambridge. Cambridge.
- Takita, K. (2020). Labeling for linearization. *The Linguistic Review*, 37(1), 75–116.
- Takita, K., Goto, N., & Shibata, Y. (2016). Labeling through Spell-Out. *The Linguistic Review*, 33(1), 177–198.
- Tegmark, M. (2014). *Our mathematical universe: My quest for the ultimate nature of reality*. Penguin Books.
- Tomasello, M. (2003). *Constructing a language: A usage-based theory of language acquisition*. Harvard University Press.
- Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, 2nd series* 42(1), 230–265.
- Uriagereka, J. (1999). Multiple spellout. In S. D. Epstein & N. Hornstein (Eds.), *Working minimalism* (pp. 251–282). MIT Press.
- Van Steene, L. (2021). *Minimax Feature Merge: The Featural Linguistic Turing Machine* (BA dissertation). University of Cambridge.
- Vergnaud, J.-R. (2008). Letter to Noam Chomsky and Howard Lasnik on “Filters and Control,” April 17, 1977. In R. Freidin, C. P. Otero & M. L. Zubizarrete (Eds.), *Foundational issues in linguistic theory: Essays in honor of Jean-Roger Vergnaud* (pp. 3–15). MIT. (Original work published 1977)
- Watumull, J. (2012). A Turing program for linguistic theory. *Biolinguistics*, 6(2), 222–245.
- Watumull, J. (2013). Biolinguistics and Platonism: Contradictory or consilient? *Biolinguistics*, 7, 301–315.
- Watumull, J. (2015). *The linguistic Turing Machine* (Doctoral dissertation). University of Cambridge. Cambridge, UK.
- Watumull, J., Hauser, M. D., Roberts, I. G., & Hornstein, N. (2014). On recursion. *Frontiers in Psychology*, 4.
- Wiltschko, M. (2014). *The universal structure of categories: Towards a formal typology*. Cambridge University Press.

Zeijlstra, H. (2014). On the uninterpretability of interpretable features. In P. Kosta, S. L. Franks, T. Radeva-Bork & L. Schurcks (Eds.), *Minimalism and beyond: Radicalizing the interfaces* (pp. 109–129). John Benjamins.