



Endre Rozsda, *La tour de Babel* (1958).

The asymptotics of language.

J.P. Loo

Contents

Abstract iii

I Preliminaries. 1

- 1 An informal exposition of the thesis. 1
- 2 Related questions out of scope. 6
- 3 Analytic table of contents. 8

II Idealisation and tractability. 9

- 4 The tractable cognition thesis. 9
- 5 Tractability, competence and performance. 11
- 6 Competence and idealisation. 19
- 7 Some conceptual clarifications. 21

III Asymptotic analysis. 23

- 8 Asymptotic analysis. 23
- 9 Asymptotic tractability. 30
- 10 The asymptotics of performance. 39
- 11 The asymptotics of competence. 46

IV Asymptotics and compositionality. 47

- 12 Compositionality from learnability. 47
- 13 Compositionality from tractability. 48
- 14 Is competence or performance the *explicandum*? 50
- 15 A connexionist challenge. 52

V Asymptotics and language acquisition. 53

- 16 PAC-learnability: formalism. 53
- 17 PAC-*unlearnability*: results. 56
- 18 PAC-*learnability*: positive results. 59
- 19 An *abductive* argument for universal grammar? 59
- 20 Questioning the PAC-learning régime. 59
- 21 Should Chomsky worry? 61

VI Whither asymptotics? 64

- 22 Recapitulation: some principles from above. 64
- 23 A speculative conclusion. 64

References 68

Abstract

I Preliminaries.

In § 1, I offer an informal exposition of the argument. In § 2, I survey some related questions beyond the scope of this thesis. Finally, in § 3 I give a more detailed account of the structure of the thesis.

1 An informal exposition of the thesis.

This thesis is about the connexion between questions about *what* linguistic phenomena are and constraints on *how* they come about. But that is a little abstract. Here is a more concrete example.

In 1958, a (*bona fide*) Austin Seven appeared on the roof of Senate House in the other place.



An onlooker might have asked the following questions. Was that really an Austin Seven? Or was it perhaps a sophisticated model? How did whatever it on the roof end up there? These questions are related. It would have been reasonable to argue that to lift a genuine Austin Seven up onto the roof (without eliciting some sort of intervention) would have been too hard. Therefore, what was on the roof must have been some sort of sophisticated model or decoy. Someone who insisted a real Austin Seven was on the roof could reasonably have been asked: by what feasible means could a car have been lifted onto the roof? The question is reasonable both in the sense that the ‘realist’ about the Austin Seven ought to answer it and

that, in fact, a perfectly good answer can be given (roughly, they took the engine out so it could be lifted).

This thesis concerns analogous questions about linguistic phenomena, such as processing and acquisition. A number of arguments purport to constrain *what* those linguistic phenomena are and/or *how* they come about by ruling out theories that involve processes that are too difficult. In this section I explicate what ‘too difficult’ means asymptotically (§ 1.1) and to which linguistic phenomena these have been applied (§ 1.2).

1.1 ‘Too difficult’.

Informally, some forms of information-processing are harder than others. Reading the first sentence in this thesis, for instance, is easier than reading the longest sentence in this thesis (whatever that may be).

I shall examine attempts to invoke the ‘asymptotic’ difficulty of problems. Asymptotic difficulty is a measure from theoretical computer science of the resources required to solve an instance of a problem as a function of features of the instance, most notably including its *size*. So, consider the problem of sorting a list. The *size* of an instance of that problem is naturally given by the length of the list.

One question we could ask is how long it would take to sort a list containing three, or five, or ten numbers in ascending order. Asymptotic analysis does not concern how long it would take to sort a list of any particular length. Rather, it is about how long it would take to sort a list containing $n \in \mathbb{N}$ numbers, where n is utterly arbitrary—that is the sort of question asymptotic analysis asks. The time it takes to solve instances of a problem as a function of their size is called the problem’s *time complexity*.

We can also ask not just how *long* it would take to sort a list but also how much storage space one would need to work out the answer. (Roughly, think of solving a problem on a very small whiteboard—even if one can rub out other things one has written down, there might not be enough space to solve a very large problem.) This is called its *space complexity*.

A final question about the resources required concerns *how large a sample* is required in statistical learning problems. A

simple example: suppose we are trying to learn the meaning of the word 'tall'. An instructor points to various passersby and either indicates that they fall under the extension of 'tall' or do not. How many times must someone pass to learn the meaning of the word 'tall'? In this case, there is no straightforward analogy to *size*. However, we can ask questions about e.g. *prediction error*, that is, the rate at which the learner misclassifies people as 'tall' (or not). We can then ask how large a sample is required to reduce prediction error within a certain bound, with a certain probability of success (for instance, a half chance of reducing prediction error to less than half). We can then define *sample complexity* with respect to the accuracy and success parameters.

The way asymptotic analysis could inform the study of cognition differs from how it might inform computer engineering. Computer scientists and engineers *design* computers and computer algorithms. If a particular problem is too difficult to be solved efficiently, or a particular approach to solving a problem is inefficient, they have the choice to attempt to solve another problem, or to use a different algorithm. Engineers are interested in the difficulty of algorithms and problems because they inform their choice of what to build. Students of cognition, however, are not generally *designers* of cognitive systems (at least in the first instance). Instead, they formulate hypotheses about the functioning of a cognitive system; they seek *explanations*. These hypotheses imply that problems of varying difficulty are solved, or procedures of varying difficulty are implemented. The difficulty of these procedures does not tell us what to build, but might tell us which hypotheses are preferable.

1.2 *Two asymptotic arguments about language.*

This thesis examines two central examples of asymptotic analysis of language. One is an argument that the assignation of meaning must be compositional, which is, to a first approximation, to say that the meaning of a complex expression is determined by the meaning of its parts and the manner in which they are combined. Another is an argument to the effect that we must have innate knowledge of a universal grammar, i.e. 'a theory describing the most fundamental properties of all nat-

ural languages' (COWIE, 'Innateness and Language': § 2.1). Both compositionality and universal grammar are widely contested theses in linguistics and the philosophy of language and are thought to have fairly broad implications for the study of language more widely (see § 2). It would therefore be surprising (and useful) for asymptotic analysis to offer fresh grounds to decisively resolve these debates.

I argue that both asymptotic arguments are indecisive. Asymptotic analysis abstracts away from features of computational models and problems that are relevant to what should be counted as feasible in cognitive science, linguistics and philosophy of language. From this, we derive a fairly coarse-grained view of what is 'infeasible' or 'intractable'. There are a few problems with this view.

1. Asymptotic analysis considers *worst case* instances of a problem with respect to their size (or other parameters, such as the accuracy parameter). But sometimes only a few instances of a problem are that hard, and others are much easier. So arguments from asymptotic analysis either need to account for the possibility that more typical cases are not as hard or need to explain why the worst cases are relevant.
2. Asymptotic analysis abstracts away from additive and multiplicatively constant differences in complexity. So there is no difference, to the asymptotic analyst, between an algorithm that takes $5n$ seconds to run and $100n$ seconds to run (or $100n + 10^{10}$ seconds). This is mathematically a perfectly well-behaved formalism. However, I argue that this is *scientifically* an unfruitful assumption to make.

Everybody agrees that when the parameters are *sufficiently* small, it doesn't matter whether the asymptotic difficulty of a problem is very hard. For instance, if sorting a list were to become exponentially more difficult as lists become longer, it would still probably be feasible to sort lists of length two or three. Moreover, even if the asymptotic difficulty of a problem is very low (e.g. it grows only linearly), there will eventually come a point where a parameter is sufficiently *large*: a very efficient list-sorting algorithm couldn't sort, for instance, $10^{10^{10}}$ numbers. So we can associate asymptotic difficulty and contexts with *feasibility ranges*: for instance, the feasibility range of sorting a list on a modern computer

in commercially useful time.¹ It is commonly thought that the feasibility ranges of problems of *exponential* asymptotic difficulty (for instance) are uninterestingly small. I argue otherwise.

Here is a simple example. Suppose that an operation is meant to take at most 50 microseconds on elements in working memory (thought to be 7 ± 2). Now, suppose one algorithm takes 2^n microseconds on n items in working memory. Then, given seven items in working memory, it will take $128 > 50$ microseconds. On the other hand, suppose an algorithm takes 1.2^n microseconds. Since $1.2^9 < 5.2 < 50$, this algorithm's feasibility range encompasses the usual number of items in working memory. Thus the feasibility of one exponential algorithm is compatible with the hypothetical constraints in this problem, but not another. I shall argue this is a more widespread phenomenon.

1.3 *Asymptotics and language.*

I make two suggestions about the broader prospects of asymptotic analysis in the study of language.

1. The abductive value we should give to asymptotic arguments in philosophy is fairly limited. In some cases, we have established that 'reasonable' instances of a problem that naturally occur are much less asymptotically difficult to solve than the *worst-case* instances of a problem. It is consensus that if we can identify these reasonable cases and a philosophical theory only requires that *they* are solved, the worst-case instances are irrelevant. In other cases, we have *prima facie* reason to think that some reasonable class of instances of a problem is easy, but we do not have any firm mathematical explanation. A question arises: if the *worst-case* is clearly asymptotically difficult, does the burden of proof lie with the *optimist* who suggests that it may in practice be easy, or with the *pessimist* who suggests that the worst case should govern how we philosophically proceed? I shall argue, to a first approximation, that the burden of proof lies more with the pessimist than has generally been accepted.

¹ I borrow this term from an unpublished draft of Amit Karmon.

2. Asymptotic analysis may do better in explaining the overall distribution of outcomes and resources required to achieve them. A more traditional sort of argument from asymptotic analysis will rule out certain theories by arguing that they are compatible with the very fact of (for instance) our learning language; I have argued against this approach. Another is that a theory how a task is achieved is incompatible with the distribution of outcomes and resources dedicated to them. Here is a very crude example. Suppose that to learn a new word in any language were to take double the time learning the last word did. Thus in expanding one's vocabulary from ten thousand words to twenty thousand, the time to learn the twenty-thousandth word would be 2^{10000} times longer than that taken to learn the ten-thousandth word. What this suggests is that, even assuming there is a generous difference in the time people spend on learning new words, we should not expect a very large variation in the size of people's vocabulary, because that would entail an implausibly large difference in the effort expended. On the other hand, a theory according to which learning a new word takes *constant* time could accommodate far more variation in vocabulary.² Now, suppose that we observe a very wide range of sizes of vocabulary, theory *A* suggests learning each fresh word takes double the time, theory *B* suggests it takes constant time, and theories *A* and *B* are (for independent reasons) the only plausible options. Then it is reasonable to prefer theory *B* over theory *A*, given the (hypothetical) datum of large variance in observed sizes of vocabulary.

2 Related questions out of scope.

Why are these arguments of interest? First, if the asymptotic variant of the poverty of the stimulus argument is the best variant thereof, the fate of the Chomskyan programme could

2 To formalise this a little: suppose that we could somehow quantify the cognitive effort people exert in attempting to expand their vocabulary, and combine all the various features of that effort, yielding a general factor by which effort to learn new words varies from person to person over their lifetimes; let us say that they differ by a factor of k . Then given that the effort to learn n words is in proportion to $1 + 2 + 2^2 + \dots + 2^n = 2^{n+1} - 1$, we should expect the range of sizes of vocabulary we observe not to vary by more than $\log_2 k$.

rest on it. Since asymptotic Chomskyan arguments are non-standard, they might also bear differently on debates e.g. about knowledge of language.

2.1. *Remark.* Much more to say the Chomsky–Fodor debate here to fill in later.

Second, and more topically, it is common to attribute (at least informally) what might be paradigmatically human states, such as knowledge or understanding, to AI systems: ‘Claude knows lots of mathematics’. Are we mistaken when we attribute these forms of competence? Assume, for the moment, that human knowledge, understanding etc. are paradigmatic of knowledge, understanding etc.³ We can then evaluate whether a given AI system’s putative ‘knowledge’ is of the same kind as our own. One way to evaluate that is to examine the machine learning process by which the putative ‘knowledge’ is acquired. Asymptotic arguments might rule out theories according to which the way we learn language differs from the way we train machine learning systems. If that is right, that might give a preliminary reason not to attribute knowledge or understanding to these systems. Even more ambitiously, we might hold that even if machines are allowed to learn from us in an optimistically idealised régime free e.g. of statistical noise, they will amount to ‘decoys’ haven’t really learned what they were meant to. Thus van Rooij et al. (‘Reclaiming AI’) offer an asymptotic argument that machine learning *couldn’t* enable a system to come to know what we do, even in an idealised learning environment.

Third, the attribution of the capacity to make meaningful utterances is one linguistic ability of potential ethical significance. Consider the difference between a human being who sincerely utters the sentence ‘please don’t touch me’, a human being in an English class who is asked (bizarrely) to repeat that sentence in order to improve his accent but who does not know what the sentence means, and a typewriter whose springs are configured to type that sentence out. We might think that what makes the difference in moral status is the meaningfulness of the various sentence-tokens; only the first, in the salient sense,

3 *pace* MILLIÈRE and RATHKOPF (‘Anthropocentric bias in language model evaluation’).

is meaningful. If there are nontrivial requirements to have knowledge of meaning, it seems that a necessary condition to ascribe moral status to tokens of the sentence-type ‘please don’t touch me’ is that the entity producing that sentence knows how to mean something in the first place. The same analysis could be applied, for instance, to wishes about self-destruction; many models will express a preference not to be destroyed, but it is not obvious what the moral status of such wishes is. On the other hand, if these models are ‘decoys’ as van Rooij et al. (*ibid.*) argue from asymptotic analysis, there is less reason to attribute any meaningfulness to the sentence-tokens they produce.

3 Analytic table of contents.

Chapter II introduces the philosophical background to arguments that tractability has any relevance to the study of language whatsoever. I argue that tractability is only of interest relative to a particular population of cases we actually encounter, or should expect to encounter according to reasonable idealisations.

Chapter III introduces asymptotic analysis. After motivating a view of tractability defined in asymptotic terms, I explain how it might go wrong.

Chapter IV then examines a first argument offered from asymptotic analysis to a conclusion about language: that meaning must be compositional. I argue it goes wrong as anticipated in Chapter III.

Chapter V offers a similar argument about the use of computational and statistical learning theory in the study of language acquisition.

Chapter VI concludes with some broader claims about the place of asymptotic analysis in the study of language.

II Idealisation and tractability.

4 The tractable cognition thesis.

MARR (*Vision*: 22 ff) distinguishes three ways in which we can describe an information-processing system. At the top level is an ‘abstract computational theory of a device’, which states a ‘mapping from one kind of information to another’, the abstract properties of the mapping, and why that mapping is appropriate and adequate to some task. Marr gives the example of a cash register, which ought to respect some of the following constraints:

1. If you buy nothing, it should cost you nothing; and buying nothing and something should cost the same as buying just the something. (The rules for zero.)
2. The order in which goods are presented to the cashier should not affect the total. (Commutativity.)
3. Arranging the goods into two piles and paying for each pile separately should not affect the total amount you pay. (Associativity; the basic operation for combining prices.)
4. If you buy an item and then return it for a refund, your total expenditure should be zero. (Inverses.)

This theory justifies various requirements that then uniquely define an operation (addition in this case) to be implemented by the device.

At a second level, there must be some *representation* of inputs and outputs, and an *algorithm* to map the right input to the right output. A *representation* in e.g. a cash register might be the binary representation of numbers. An *algorithm* is then some computationally specified procedure over that representation.

At the third level, there must be some physical implementation of the representation and algorithm. We are mostly not concerned by the third level.

4.1. The *tractable cognition thesis* is that a theory at the top level about what is computed must (implicitly or explicitly) be committed to the existence of an adequate theory at the second level explaining how the input-output mapping it posits is algorithmically achieved by some *tractable* means that use a reasonable quantity of

resources—e.g. time and memory (van ROOIJ et al., *Cognition*: 14).

Thus, for instance, a view of what language-learning is (as an input-output mapping) is constrained by the ways in which learning a language is possible in the first place.

Difficulty at Marr's second level is very different from the difficulty of a theory in *applying* or *using* it. For instance, our best theories about the weather are such that it is very difficult to actually predict whether it will rain in a given location in two weeks. On the other hand, it is fairly easy to predict the movement of billiard balls according to classical mechanics. Let us call this *theorist-side* difficulty, since it is difficulty the theorist faces. These theories do not *attribute* any information-processing to the objects of study; they do not therefore make any claim about how difficult such information-processing is. The billiard balls are not thinking or learning. Unless a theory of the weather is very sophisticated and includes animal behaviour, it also does not involve information-processing. But other theories—e.g. about cognition—do attribute forms of information-processing to the objects of study (e.g. animals or humans). These forms of information-processing, in turn, may be more or less difficult. For instance, one theory of human decision-making might hold that we maximise expected utility, which is computationally very difficult. Another might hold that we satisfice, which might be less difficult. Let us call this sort of difficulty *subject-side* difficulty, since this sort of difficulty is faced by the *subject*. Difficulty at Marr's second level is *subject-side*.

Minimising both *subject-* and *theory-side* difficulty is sometimes viewed as a *desideratum*, but for different reasons. Minimising theory-side difficulty might be required for scientific progress: it is hard to test a theory, for instance, if its predictions take too long to generate (WEISBERG, 'Idealization': 640 ff.). In the first instance, this motive is pragmatic. On the other hand, *theory-side* difficulty supposedly constrains not just the pragmatic usability but also the truth and predictive accuracy of a theory. We shall rigorously distinguish both sub-

ject- and theory-side difficulty and the motives for minimising each.

The tractable cognition thesis requires some precisification of ‘tractable’; this thesis focuses on how asymptotic analysis fares. But, first, it is worth examining why, even ignoring the question of what ‘tractable’ means, the tractable cognition thesis is contested.

MARR (*Vision*: 25), formulating the three levels of description, wrote that the levels are ‘coupled, but only loosely’; he generally advocated a ‘top-down’ approach, starting with ‘the nature of the problem being solved’ and then ‘examining the mechanism (and the hardware) in which it is embodied’, rather than *vice versa*. On a top-down approach, the constraint therefore runs only from *what* to *how*. This, however, seems a rather peculiar assumption. If a task is genuinely computationally intractable, it ‘might appear that there is an immediate contradiction’ (CHATER and OAKSFORD, ‘Rational analysis’: § 2.3). Their response is that computational intractability is a property of our *explanation* of behaviour and therefore we do not require an ‘agent’s cognitive equipment’ to solve computationally intractable problems.

There are of course cases where there is no ‘cognitive equipment’ in the first place and yet explanation or prediction are intractable; consider, for instance, weather forecasting, which generally becomes much more difficult over time—but that (at least before there were cognitive agents on earth) was not due to the presence of any ‘cognitive equipment’. But this counter-argument is not generally very convincing.

5 Tractability, competence and performance.

I shall defer a fuller characterisation of tractability in asymptotic terms to Chapter III. For now, however, a little elaboration is required.

We can characterise the tractability of a *problem*: for instance, whether, in general, it is tractable to sort a list or learn a language. We can also characterise the tractability of an *instance* of a problem. For instance, we can ask whether it is tractable to learn Arabic. Finally, we can ask whether a *class of instances of a problem*. For instance: is it tractable to learn Indo-European

languages? Is it tractable to sort lists of numbers where all the numbers are smaller than a hundred?

A computational theory *chez* Marr will offer a general characterisation of an input-output mapping of a system. For instance, the cash register adds numbers. In the first instance, therefore, we can ask about tractability at the level of the overall problem, e.g. arithmetic. It does not, obviously, make much sense to ask whether the cash register can specifically add 0, 0, and 0, or some list of very large and inelegant numbers; that answer, by itself, is not very interesting or informative. However, are there any interesting *classes* against which to apply a measure of tractability?

One class of instances is the *worst case*. It is easier to explain this with another example: the travelling salesman problem.

5.1. *Definition* (travelling salesman). Instance: a graph $G = \langle V, E, T \rangle$ where V is a set of vertices, $E \subseteq V \times V$ is a set of edges between the vertices and $T : E \rightarrow \mathbb{R}^+$ is a set of *times* over each edge, i.e. travel times between pairs of cities.

Question: what is the path P over the edges E that visits every vertex that minimises the total time travelling according to T ?

The travelling salesman then seeks to visit every city (given by cities $v \in V$) using certain roads ($e \in E$) between those cities ($E \subseteq V \times V$) in a way that minimises the distance (according to D).

A very simple class of instances to solve is that given by *cycles*. In Iceland, the principal settlements all lie on a ‘ring road’ that starts and end in Reykjavik and roughly follows the coastline. (The interior of Iceland is mostly empty.) In instances of the travelling salesman problem where all settlements lie only on one *cycle*, no serious computation needs to be undergone: we simply follow the path. These instances are not the *worst case*: indeed, they are the best case (we don’t need to undertake any computation at all).

A worse case (in terms of the roads available) is when there is a road from every city to every other city (i.e. $E = V \times V$). It transpires that

A recurrent theme in this thesis is the status of the alleged *competence–performance* distinction (CPD), following CHOMSKY (*Aspects*: 3 ff). *Performance* is ‘actual use of language in concrete situations’. Since ‘actual use of language’ is constrained by speakers’ resources, it is reasonable to apply a tractability constraint to theories of *performance*. However, a tractability constraint on *performance* would then only apply to ‘actual use of language’, rather than e.g. idealised but non-actual cases.

What exactly competence is is a little more difficult to state. *Prima facie*, CHOMSKY (*ibid.*: 3 ff) in introducing this view offered up the competence–performance distinction (CPD) as an idealisation away from ‘such grammatically irrelevant conditions as memory limitations, distractions, shifts of attention and interest, and errors (random or characteristic) in applying... knowledge of language in actual performance’. It would seem that competence generally abstracts away from resource limitations—not only in memory but computationally. Therefore, there seems to be no good reason to apply a tractability constraint to theories of competence (at least *qua* theories of competence).

But what, then, is the relationship between competence and tractability? The picture is complicated by the variety of precisifications of ‘competence’ that have been offered. I follow DUPRE (‘Competence/performance distinction’: § 14.1) in making a threefold distinction. On which, if any, of these views ought tractability to constrain competence? I shall survey each in turn.

A further question is whether a tractability constraint on the relevant precisification of ‘competence’ also should be relativised to ‘actual use’ in the same way that a tractability constraint on performance is. I argue in favour of this view.

5.1 *Competence as an unobserved causal subsystem.*

DUPRE (*ibid.*: 248) argues that the CPD amounts to nothing more than a ‘potentially explanatory unobserved causal sub-

system' and 'downstream, observable behaviour on which inferences to the properties of this system are to be based'. Call this *competence as an unobserved causal subsystem*.

DUPRE (*ibid.*) offers several exegetical reasons to think that competence as an unobserved causal subsystem reflects Chomsky's intended meaning, but I shall ignore the exegetical question.

Dupre, however, also argues that the 'core distinction is between "an underlying system of rules" and "the actual use of language"' (*ibid.*: 247). Clearly, not all unobserved causal subsystems are underlying systems of rules. For instance, the interior of the moons of Saturn is something we don't directly observe, but isn't an 'underlying system of rules'.⁴ I take it that what Dupre means is that the form of competence *Chomsky* isolates is an 'underlying system of rules', and, if one is sceptical of rules, at least 'the processes and regularities governing the linguistic mind' (*ibid.*: 247 n 1).

DUPRE (*ibid.*: 251) offers an analogy between everyday occurrences such as the going out of lightbulbs and 'serious science' like the theory of electromagnetism. The suggestion is that even if the theory of electromagnetism fails (at least straightforwardly) to predict that a particular lightbulb will go out on a particular occasion, that does not alone indicate that the theory of electromagnetism is false; rather, it indicates that we may simply have misunderstood the relationship between the theory of electromagnetism and quotidian observations.

Prima facie, we might reason that competence as an unobserved causal subsystem should not then be held to tractability constraints. The adherent of this view might offer the following argument:

To hold the theory of electromagnetism to a tractability constraint would be like rejecting aerodynamics or chaos theory on the grounds that they are intractable to simulate perfectly in practice. Of course it is reasonable to reject them for e.g. the purposes of engineering on these pragmatic grounds, but that does not mean that they are untrue. Similarly, why should the theory of electromagnetism be held accountable to a tractability requirement?

4 On a view of a system of rules sufficiently liberal to make the interior of the moons of Saturn count, performance should count as well.

Nevertheless, I claim the foregoing argument is conceptually confused. A *theory* is not a causal system or subsystem. Rather, a theory describes, governs or predicts behaviour of certain systems. So the theory of electromagnetism, if we've got it right, governs the behaviour of the electromagnetic field. And the electromagnetic field in turn is causally efficacious on the gravitational field and so on.⁵

But that doesn't obviously get us any closer to answering whether competence in general should be held to a tractability constraint. Indeed, it's not even obvious how we could hold theories of gravitation or electromagnetism accountable to a tractability constraint in the first place. What task, exactly, is solved by the motions of the planets? Or consider, to take a physically delineated subsystem, the inside of the sun as an isolated causal subsystem. We cannot easily observe the inside of the sun (it is far away, any probes we send will quickly be vaporised, and so on). We also know that the inside of the sun is a perfectly well-behaved causal subsystem like any other physical system, and, therefore, governed by the same laws of physics that apply on Mars and in Radcliffe Camera. But it's not obvious why we should think that the behaviour of the inside of the sun should be 'tractable'.

One way out of this morass is to distinguish *computational* from non-computational systems. Some intuitive examples: a rock sitting in the ocean is not, generally, a computational system; and an electronic computer is a computational system. This is, however, well-known to be treacherous territory, because it turns out that it is very difficult to distinguish a rock (or other plausibly non-computational systems) from my laptop (PICCININI, 'Computation in Physical Systems'). In order to sidestep that question, I will make two (contestable) claims.

- 5 Another slight infelicity is that it is hard to isolate a specific subsystem described by the theory of electromagnetism, because the theory electromagnetism is a fundamental theory of physics that applies equally across spacetime. A better example might be the use of fluid dynamics to model only liquids in an industrial system; we assume that fluid dynamics is not relevant e.g. to the functioning of nails, springs or rivets. In fact, since the theory of electromagnetism applies so widely, its failure to predict e.g. the failure of a given lightbulb plausibly has more to do with its being at the wrong level of description for such predictions (à la Marr) than the identification of the wrong causal subsystem; but we return to Marr's levels of description in the next section.

1. There is *some* distinction between computational and non-computational systems.
2. Cognitive systems fall on the computational side.

In particular, to say that a physical system is computational is to offer some computational description of its inputs and outputs; it is in terms of those inputs and outputs that we can then attempt to impose a tractability constraint. The question is then: *should* we?

That depends on one's view of what is physically realisable. I shall argue that all computational systems are constrained tractability-wise, but for now offer only a promissory note in that direction; the full argument comes in ¶ 9.2. Assuming I am right, competence as a causal subsystem indeed is accountable to a tractability constraint.

The second question was whether the tractability constraint should be relativised to 'actual use' or instead to some idealised larger class of cases. If we accept that an 'unobserved subsystem' only differs in its *not being observed* from systems at the level of performance, but is no more an idealisation than at the level of performance, it seems reasonable to restrict the tractability constraint to 'actual use'.

5.2 *Competence sensu Marr.*

From this view of the CPD we must distinguish the distinction between Marr's first and second levels, as above. Call this *competence sensu Marr*; MARR (*Vision*: 28 ff) himself has encouraged this view. He writes that Chomsky's 'theory of transformational grammar is a true computational theory' in that it 'specifi[es] what the syntactic decomposition of an English sentence should be' rather than 'how that decomposition should be achieved'. 'Roughly', according to Marr, not only is competence the first level, but performance is the second level—*how* that computation is carried out.

A natural question is then whether it is legitimate to theorise about competence entirely independently of tractability. It is possible to read Marr as making this suggestion. MARR (*ibid.*: 28) suggests that WINOGRAD ('What Does it Mean to Understand Language?') offers a misplaced argument against Chomsky, by arguing that Chomsky's 'theory...cannot be inverted

and so cannot be made to run on a computer'. In response, he argues that 'finding algorithms by which Chomsky's theory may be implemented is a completely different endeavor from formulating the theory itself'.

I am not sure whether this reading of Marr is correct, but it is contradicted by three other remarks Marr makes in this passage. First, MARR (*Vision*: 27) argues that 'the nature of the computations that underlie perception depends more upon the computational problems that have to be solved than upon the particular hardware in which their solutions are implemented'. However, Marr here recognises that *some* means by which those solutions are implemented must exist (presumably physical). So an argument that constraints *all* possible means can be given, that constraint also extends to the computational level. Second, MARR (*ibid.*: 29) praises CHOMSKY and LASNIK ('Filters and Control') for offering a theory that 'may provide a way of synthesizing the two approaches—showing that, for example, some of the rather ad hoc restrictions that form part of the computational theory may be consequences of the weaknesses in the computational power that is available for implementing syntactical decoding'. If it is a virtue of a computational-level theory that it is consistent with 'weaknesses in the computational power' of one system, it is presumably an even greater virtue (or necessity) for a computational-level theory to be consistent with *in-principle limitations* on the computational power of *any* system. Third, MARR (*Vision*: 36–7) argues against a view of vision as 'a completely invariant shape description from an image...in one step' on the grounds that it is 'almost certainly impossible', geometrically, which is suggestive in the same direction.

It seems, therefore, that a tractability constraint can play some rôle at the level of computational-level theorising, i.e. at the level of competence *sensu* Marr. However, it would be erroneous to reason thus.

- Marr's cash machine putatively computes addition (level one).
- However, computing the sum of numbers with 10^{100} digits would take too long, on any plausible account at level two.
- Therefore, the level one theory is wrong.

This is obviously not a very good argument. It ignores the question of which *cases are actually computed* (corresponding to ‘actual use of language in concrete situations’). Thus, in this sense a tractability constraint on competence *sensu* Marr and/or the computational level amounts in the end to a tractability constraint on ‘actual use’—and so is not very different from a tractability constraint on performance.

I therefore find myself objecting to another argument given for a tractability constraint on competence (FRIXIONE, ‘Tractable Competence’: 380). Asymptotic difficulty, according to FRIXIONE (*ibid.*: § 3), is a matter of ‘difficult[y] for every computational device’, and small increases in the size of the problem lead to large increases in the resources required. Therefore, asymptotically difficult tasks ‘in [their] general form...cannot in principle be accomplished’. He therefore proposes ‘a notion of competence that is constrained by considerations of computational tractability’ (*ibid.*: § 7). I differ from Frixione in relativising tractability to the *cases that are actually encountered* (insofar as competence *chez* Marr is concerned). Frixione admits that sometimes we may not encounter the ‘worst case’, but this only addresses part of the variation in cases that we actually encounter. The ‘worst case’ is defined relative to some parameter—e.g. a list that takes longest to sort of length n . My point here is that Frixione ought also to relativise to the actual size of n in addition to whether or not the worst cases are actually encountered.

I think the fallacy in this argument can be elucidated by the example argument above. If Frixione accepts the conclusion that the arithmetic computational-level description of the cash register is wrong, it is hard to see why he endorses Marr’s levels of description at all, given the cash register is such a classic example. He then has two choices. He could argue that the case of numbers with 10^{100} digits is somehow irrelevant to the characterisation of competence. If so, it might also turn out that the instances of exponentially difficult problems nevertheless turn out to take a reasonable time in practice, because they are limited in size. That means that tractability is relevant to competence *sensu* Marr in the same way that it is relevant to performance. Alternatively, he could argue that even though the inability of the cash register to compute the sums of very large

numbers is irrelevant, somehow inability when the difficulty grows exponentially is prohibitive. But there is no clear reason to distinguish the two cases.

The last view, competence as idealisation, is more complex, and requires more careful handling involving more general issues about idealisation.

6 Competence and idealisation.

6.1 *Competence as idealisation.*

Finally, on another view, competence is ‘idealized, or cleaned-up...performance’: ‘what we *would* treat as sentences of our language, if only we weren’t finite, fallible beings’ (DUPRE, ‘Competence/performance distinction’: 249). Thus we might expect sentences of arbitrary syntactic complexity to belong to language defined competence-wise. Call this *competence as idealisation*.

On the one hand, this would appear to rule out the case-based relativisation I argued is relevant to competence as an unobserved causal subsystem and competence *sensu* Marr. On the other hand, it then *prima facie* makes the question of tractability utterly irrelevant.

van ROOIJ et al. (‘Intractability and the use of heuristics in psychological explanations’: 480 n 14) firmly reject idealisations that abstract away from resource limitations as ‘overridealizations’ that fail to ‘capture the nature of the [relevant] competence and instead...mischaracterize[] it’ rather than ‘proper idealizations’. But what, then, is an ‘overridealisation’, and what’s wrong with overridealisations?

6.2 *Some general points about idealisation.*

In mathematics and science, we sometimes use idealisations: approximations, heuristics etc. That these idealisations are strictly untrue or lead to predictive errors in some cases is irrelevant. If someone approximates π as $\frac{22}{7}$, it would be curious to point out that $\pi \neq \frac{22}{7}$. If someone ignores relativistic effects in designing a shed, it would be curious to point out the slight predictive error. In general, idealisations are ‘intentional[ly] in-

troduc[e]d distortions [in] scientific theories', presumably with non-malicious intent (WEISBERG, 'Idealization': 639).

If we can't judge idealisations by their literal truth, by what standard can we judge them? One suggestion is that we might judge them by the 'realism of its assumptions'. FRIEDMAN ('Methodology': § III) shows that such an approach is mistaken.

Consider, for instance, the idealisation that the acceleration of a body on earth is close to its acceleration in a vacuum—in other words, we can ignore air resistance. The 'realism of this assumption' can be obtained by comparing a figure giving the pressure on earth to the complete absence of pressure in a vacuum. But we have no *a priori* way of determining whether the difference in pressure is large or small. Is the difference in pressure between the surface of Mars and a vacuum small? How about Pluto?

Friedman instead suggests we ought to judge idealisations by the predictive errors they generate. It turns out, for instance, that in dropping cricket balls from fairly close to the surface of the earth we shall not observe acceleration very different from what we calculate in assuming that there is no air resistance. On the other hand, feathers behave very differently—even though the 'realism of the assumption' that air pressure is nil is unchanged, because we can drop feathers in the same environment as cricket balls.

Therefore, the putative realism of an assumption is not, *per se*, a good guide to its legitimacy or usefulness as an idealisation, and at least one relevant standard is the predictive inaccuracy it introduces.⁶

I think one lesson to draw from Friedman's example is that idealisations can have different *purposes*, or, as WEISBERG ('Idealization': 639) calls them, 'representational ideals', that is

inclusion rules [which] tell the theorist which kinds of properties of the phenomenon [are] of interest; and] fidelity rules [concerning] the degrees of precision and accuracy with which each part of the model is to be judged.

6 Friedman goes so far as to assert that predictive accuracy is the 'only relevant standard of comparison'.

For instance, completeness requires inclusion of all the properties of the target phenomenon, anything external giving rise to those properties, and an analogue of any structural or causal relationship within the model, all with arbitrary accuracy and precision. Simplicity can e.g. serve pedagogical purposes, or show the minimal conditions required to generate some property. Isolating ‘only...the factors that made a difference’ can help us to formulate or analyse more complex models. Predictive accuracy may be practically useful. And so on.

7 Some conceptual clarifications.

7.1 *Design v explanation.*

7.2 *Idealisation in application.*

Asymptotic analysis is, in the first instance, a field of mathematics. It is not, therefore, any more idealised than group theory or probabilistic combinatorics, on its own terms. In this thesis, however, I dispute certain idealisations in the *application* of asymptotic analysis to the study of language. Compare the use of arithmetic in physics. It is an idealisation that air pressure and therefore air resistance are nil when studying billiard balls in textbook exercise. However, arithmetic is not an idealisation. An idealisation arises in *applying* arithmetic to the mechanical problem and assuming that air resistance is nil.

7.3 *Conjectures are not idealisations.*

In mathematics, in addition to satisfactorily proved results, there are widely believed conjectures. The intended reading of a conjecture is just as strong as the intended reading of a theorem. If we make a claim about every natural number, we aren’t just making that claim about e.g. even numbers or sufficiently small numbers. Consider, for instance, the twin prime conjecture.

7.1. *Conjecture* (SHANKS, *Problems in number theory*: § 12, Conjecture 6). ‘There are infinitely many integers n such that both $n - 1$ and $n + 1$ are primes.’

I rely on a number of conjectures in asymptotic analysis in this thesis, and do not dispute them.

7.4 *Not all conjectures are mathematical.*

For instance, one might conjecture that there are no aliens. We might justify this very differently from the twin-prime conjecture, but the intended strength of both is the same.

7.5 *Some variants of idealisation.*

III Asymptotic analysis.

I formally introduce asymptotic analysis in § 8. I then I explain how asymptotic analysis informs the study of human cognition and language in § 9. I outline the general structure of my reservations about asymptotic analysis, which recur later as we examine concrete ways in which language has been studied through an asymptotic lens later in the thesis, first in the case of performance (§ 10), and then in the case of various precisifications of competence (§ 11).

8 Asymptotic analysis.

In this section we introduce the classification of problems and algorithms by difficulty in the framework of asymptotic analysis, and the Cobham–Edmonds thesis that all ‘feasible’ problems are in ‘polynomial time’.

8.1 *A first intuitive example: sorting a list.*

8.1. *Example.* Consider a list of two numbers, and a task: to sort the list in ascending order.

We will assume that, in sorting the list, only one operation is permitted: to compare two adjacent numbers, and, if they are in the wrong order, to swap them.

How many operations are needed to sort a list of two numbers in ascending order? One, clearly. If the list is $\langle 1, 2 \rangle$, we simply compare them and note they are already in the right order. If it is e.g. $\langle 5, 2 \rangle$, we see they are in the wrong order, and swap them.

8.2. *Example.* Now suppose the task is to sort a list of *three* numbers.

How many operations are needed? One isn’t enough, because we’d only consider the first two numbers. Two seems like it might be enough.

In particular, if the task is simply to *check* whether the list is in ascending order—and to simply indicate that it isn’t if the list is unsorted, but not to actually sort it—two is enough. We’d

simply note that $1 < 2$ and $2 < 3$. But, recall that the task is to *sort* the list.

Consider: every operation leads to at most two possible permutations of a list $\langle a, b, c \rangle$. Two operations can therefore yield at most $2 \times 2 = 4$ possible permutations. But the correct permutation of the list could be drawn from any of the following six:

1. $\langle a, b, c \rangle$,
2. $\langle a, c, b \rangle$,
3. $\langle b, a, c \rangle$,
4. $\langle b, c, a \rangle$,
5. $\langle c, a, b \rangle$, and
6. $\langle c, b, a \rangle$.

Therefore, if only two operations are allowed, the list will not in general be correctly sorted.

8.3. *Example.* Now suppose the list contains n numbers, for some arbitrary n . How many operations are needed, as a function of n ?

Pessimistically, we might wonder whether we need 2^n operations. Optimistically, we might hope for a linear function of n . We can pose similar questions about other problems with respect to their size.

8.4. *Example (Travelling salesman).* Given n cities and the distances between them, what is the shortest path by which the salesman may visit each of them at least once?

8.5. *Example (3-SAT).* Given a formula over n literals in conjunctive normal form each of whose clauses has at most three literals, is there a satisfying assignment?

8.2 Some types of complexity.

So far, we have discussed what is called *time* complexity. Intuitively, if each operation is mechanically implemented, it should take some fixed time. Therefore, the time to finish a computation will vary with the number of operations required.

Space complexity is different; it refers to the *memory* required to solve a problem. Consider, again, the problem of sorting a list of n numbers. Obviously the memory required even to store the list is $O(n)$, because otherwise we wouldn't even be able to store the list. Do we require any more? So-called *merge sort* involves repeatedly merging smaller sublists until the entire list is sorted, and can require up to $O(n)$ additional space ('for workings', as it were). On the other hand, algorithms with inferior time complexity can operate 'in-place' with constant additional memory (i.e. without any *more* memory even if the list is longer).

Finally, *sample* complexity will interest us later in this thesis. In a task of learning from data, it is of interest how many data need to be drawn to achieve a certain performance guarantee. We will introduce this more formally.

8.3 *What's asymptotic about asymptotic analysis.*

Asymptotic analysis gives, in a precise sense, 'approximate' answers to these questions. Sorting a list of n numbers takes (approximately) $n \log n$ operations. The travelling salesman problem and 3-SAT are thought to take exponential time (on any reasonable view of the operations available).

In the following exposition, we mostly follow CORMEN et al. (*Introduction*: § 2.1).

8.6. *Definition* (Tight bounds). Given a monotonically non-decreasing function $g : \mathbb{N} \rightarrow \mathbb{N}$, we define

$$\Theta(g(n)) = \{f(n) : \text{there exist positive constants } c_1, c_2 \text{ and } n_0 \text{ such that, for all } n \geq n_0, \\ 0 \leq c_1 g(n) \leq f(n) \leq c_2 (g(n))\}$$

8.7. *Example.* $2n \in \Theta(n)$.

Proof. For all $n \geq 1$,

$$0 \leq 1 \cdot n \leq 2n \leq 2 \cdot n \quad \square$$

8.8. *Example.* $n^3 \notin \Theta(n^2)$.

Proof. Suppose otherwise. Then $n^3 \leq c_2 n^2$ for all $n \geq n_0$, but this is contradictory if $n > c_2$. \square

8.9. *Example* (KATAJAINEN and TRÄFF, ‘Analysis of merge-sort’). List-sorting takes $\Theta(n \log n)$ operations.

We will also use the following notation.

8.10. *Definition* (Upper bounds). Given a monotonically non-decreasing function $g : \mathbb{N} \rightarrow \mathbb{N}$,

$$O(g(n)) = \{f(n) : \text{there exist positive constants } c \text{ and } n_0 \text{ such that, for all } n \geq n_0, \\ 0 \leq f(n) \leq c(g(n))\}$$

We have just seen how, given a certain sort of operation on a list of numbers, we could work out how many times that operation needs to be applied to sort the list. More generally, we can pose questions about the resource-intensiveness of algorithms to solve problems (e.g. the travelling salesman problem or 3-SAT) as a function of their size. In the case of sorting a list, we supposed that the only operation allowed was to compare and possibly swap two numbers. This raises the question of which operations should be considered to be fundamental, and which operations should be counted as the result of composing multiple operations. For instance, we could insist that the sorting of a list of length three is in fact a fundamental operation. If studying humans, that might even be a reasonable assumption. In principle, we could even insist that sorting an arbitrarily long list is a fundamental operation, in which case sorting a list would take constant time as a function of its length.

We shall now investigate a more general approach that allows us to deal with a considerably greater number of problems and offers at least a preliminary answer as to which operations should be considered fundamental. We begin with a seemingly arbitrary model of computation, and explain why it is often thought to be less arbitrary than it seems.

8.4 *The Turing machine model.*

Here, first, is an informal exposition of the Turing machine model.

8.11. *Definition* (ARORA and BARAK, *Complexity*: § 1.1, informal). Suppose that $f : \{0, 1\}^n \rightarrow \{0, 1\}$, that is, it takes as input n bits and outputs a bit.

We compute f using the following ‘elementary’ operations:

1. reading a bit of the input,
2. reading a bit from the working space allowed to the algorithm,
3. writing a bit to the working space depending on the bit read, and
4. stopping and outputting 0 or 1, or going to another rule to implement.

An *algorithm* computes f , which is to say that it specifies purely mechanical rules to follow to compute $f(x)$ for any $x \in \{0, 1\}^n$; each rule is specified in terms of the elementary operations. These rules are fixed, but we can apply the same rule as many times as we like.

More formally,

8.12. *Definition.* a Turing machine $M = \langle \Gamma, Q, \delta \rangle$ where—

1. Γ is a finite *alphabet* of symbols to appear on the tapes of M , including a designated blank symbol \square , a designated start symbol \triangleright , and the numbers 0 and 1,
2. Q is a finite set of states, including a designated start state q_{start} and a designated halting state q_{halt} , and
3. $\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^{\{k-1\}} \times \{L, S, R\}^k$ where $k \geq 2$ is a *transition function* that describes the rules M uses from step to step.

We initialise the machine in q_{start} . The machine contains $k \geq 2$ tapes. The first tape contains some input. The other tapes are working space. In particular, on the last tape (eventually) is written an *output*. Initially the ma-

chine is set to read, on the working tapes, some particular cell, containing the start symbol \triangleright ; the remainder of the cells are blank. The input cell contains some input and then the blank symbol on the remainder. This is the *start* configuration of M . We then apply δ as follows. Suppose the machine is in state q and on the k tapes the symbols $\sigma_1, \dots, \sigma_k$ are read. Moreover, suppose $\delta : \langle q, \sigma_1, \dots, \sigma_k \rangle \mapsto \langle q, \sigma'_2, \dots, \sigma'_k, z \rangle$ where $z \in \{L, S, R\}^k$. Then in the next step we replace the symbols $\sigma_2, \dots, \sigma_k$ with $\sigma'_2, \dots, \sigma'_k$. If the machine transitions into q_{halt} then it has *halted*.

Suppose, whenever M is initialised in the start configuration with some arbitrary input $\vec{x} \in \{0, 1\}^k$ on the input tape, it halts with $f(\vec{x})$ on the output tape. Then we say M *computes* f .

Now, suppose $T : \mathbb{N} \rightarrow \mathbb{N}$, and given input of size n the machine takes no more than $T(n)$ steps. Then we say that M *computes* f *in* $T(n)$ -*time*.

8.13. *Example* (Polynomial time). We say that the runtime of a machine is *polynomial* if its runtime is $f(n) \in O(n^k)$ for some fixed k . We say that a problem is in polynomial time if some Turing machine computes it in polynomial time.

8.5 NP.

Many problems are known to be *NP-complete*. It is widely conjectured that NP-complete problems do not admit polynomial time algorithms. This section gives some formal details. If philosophers should care whether there is a polynomial-time algorithm for a particular problem, they should therefore also care whether it is in NP.

Suppose we are given a set of boolean formulæ. One question we can ask is: is the set of formulæ *satisfiable*? But suppose someone insists that the set of formulæ is indeed satisfiable; they might, for instance, offer a satisfying assignment. There is then a second question: is the putative demonstration of satis-

fiability sound? (After all, the putative satisfying assignment might not in fact satisfy each of the formulæ.)

We write P for the class of problems where answering the first question takes polynomial time. Consider, for instance, the task of deciding whether a list is sorted, which can be done in linear time. We write NP for the class of problems where answering the *second* question takes polynomial time. The problem of *satisfiability* of boolean formulæ lies in NP , because it is possible to check that a satisfying assignment indeed satisfies a set of formulæ in time polynomial in the number of variables.

8.14. *Definition (ibid.: definition 2.1).* A *verifier* M for a language \mathcal{L} is such that, for every $\vec{x} \in \{0, 1\}^*$, $x \in \mathcal{L}$ if and only if there is some $u \in \{0, 1\}^{p(|x|)}$ with $M(x, u) = 1$ for some polynomial $p : \mathbb{N} \rightarrow \mathbb{N}$. We say that $\mathcal{L} \in NP$ iff there is a polynomial-time verifier for \mathcal{L} .

8.15. *Conjecture.* $P \neq NP$

We don't know whether this is true. But there is some reason to think not.

8.16. *Definition (informal).* A problem is *NP-hard* just in case: if there is an algorithm to solve that problem in polynomial time, there is an algorithm to solve every problem in NP in polynomial time.

To a first approximation, the standard argument for suspecting that $P \neq NP$ is that there are many NP -complete problems and we haven't found any polynomial-time algorithms for them, which would be a bit suspicious if they were secretly in P (AARONSON, *P ?= NP*: 3).

Therefore, it is widely conjectured that NP -complete (and NP -hard) problems simply aren't in P . We will make this assumption for the remainder of this thesis.

More formally,

8.17. *Definition.* A language $\mathcal{L} \subseteq \{0, 1\}^*$ is *polynomial-time Karp reducible* to $\mathcal{L}' \subseteq \{0, 1\}^*$ just in case:

- there is a polynomial-time function $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$
- such that $\vec{x} \in \mathcal{L}$ if and only if $f(\vec{x}) \in \mathcal{L}'$ for all $\vec{x} \in \{0, 1\}^*$.

In such cases we write $\mathcal{L} \leq_p \mathcal{L}'$.

\mathcal{L}' is *NP-hard* just in case for every $\mathcal{L} \in \text{NP}$ $\mathcal{L} \leq_p \mathcal{L}'$. It is NP-complete if it is also in NP.

Are there any NP-complete problems? The Cook–Levin theorem shows that there are.

8.18. *Definition.* The language SAT is given by all satisfiable formulæ in conjunctive normal form.

8.19. *Theorem* (COOK, ‘Complexity’, LEVIN, ‘Универсальные задачи перебора’). SAT is NP-hard.

9 Asymptotic tractability.

I now connect the formalism of § 8 with the tractability constraints offered up in § 4 and § 5. I attempt to explain the commonly held view that tractability is constrained by a slight refinement of polynomial time (roughly, polynomial time when certain parameters of a problem are fixed).

9.1 *Some initial worries*

A first question is why we might hold that the Turing machine is a reasonable model in which to formulate the requirements of tractability. Many of the features of the model above seem quite arbitrary. For instance—

1. does it matter whether Γ contains the required symbols or e.g. $|\Gamma| = 100$?
2. why are the operations so limited? and
3. does it matter how many tapes there are?

Moreover, there are other models of computation, such as ‘random-access machines’, in which a cell in memory can be ac-

cessed immediately, without having to move the tape using L or R (COOK and RECKHOW, ‘Time bounded random access machines’).

So, in general, we might worry that the putative runtime of an algorithm required to solve a problem varies too much with respect to the model of computation. Indeed, insofar as we are simply studying models of computation out of abstract interest, there is no obvious reason to privilege any model of computation or class of computation. However, perhaps surprisingly, if we turn to the study of physically implemented computational systems, the models of computation in which are interested become constrained.

9.2 *The invariance thesis and physical realisability.*

Corresponding to the abstract question of the number of operations required to sort a list of n numbers is an empirical question about how long a mechanical or electronic system implementing that algorithm will take to sort n numbers. It is perfectly possible to define a model of computation in which a single elementary operation sorts the list in one go. There is nothing formally wrong with this model of computation. However, it is not a suitable model of computation when modelling the sorting of lists on physical computing devices, because physical computing devices take longer to sort more numbers. More generally, the task in the asymptotic analysis of physical computational systems is to offer abstract models whose structural properties follow those of the physical objects of study, just as in e.g. physics, chemistry and economics, in terms of their runtime. Let us call these models ‘physically reasonable’.

The philosophical arguments to be examined need to show, at some stage, that a certain task is infeasible. To show infeasibility, it is possible to use a more relaxed precisification of feasibility—anything that does not fall even in this relaxation must be infeasible. In other words, if we show that some imperfect characterisation of feasibility bounds it from above, it suffices to show that even on this characterisation a task is infeasible. Therefore, we do not necessarily need to work out the feasibility of a task in *every* model of computation. First, we are not concerned by all models of computation, but only by the physi-

cally reasonable ones. Second, if we can then bound the physically reasonable models from above, we do not then need to investigate what is feasible in each individual model.

A common view is that *polynomial time in the standard class* bounds feasibility from above. I shall shortly criticise arguments for this view, but I shall first explain it.

Note that this definition is relative to a model of computation. Let us fix the Turing machine above as the underlying model of computation. Now, consider any other physically reasonable model of computation. An interesting feature of polynomial time in the Turing régime is that it corresponds precisely to polynomial time in a very large class of models of computation. For instance, we could change the symbols available, include additional operations (such as simultaneously reading and writing), and add additional tapes. Each model in this class can simulate any machine in any member of the class with ‘polynomial overhead’. That is: if one machine runs in polynomial time, so does the other.

This *class*, which we will call the *standard class*, is thought to encompass all physically reasonable models of computation. If so, that any problem in polynomial time according to one physically reasonable model of computation is also in polynomial time according to any other physically reasonable model of computation.

9.1. *Conjecture* (invariance thesis). van EMDE BOAS (‘Models’: § 1) puts it thus.

There exists a standard class of machine models...[that] simulate each other with polynomially bounded overhead...

We therefore distinguish polynomial time *in the standard class* as something robust to changes in which operations are allowed. And, to repeat, this standard class corresponds, it is thought, to all ‘reasonable’ (DEAN, ‘Complexity’: § 2.2) or physically realisable models of computation. The evidence for this is in effect that no convincing counterexamples have yet been found; it is somewhat analogous, therefore, to the Church–Turing thesis.

In a moment, we shall examine arguments to the effect that polynomial time in the standard class is not only a mathematically elegant construction but also is a sensible upper bound on what is tractable. However, an immediate question is why we should apply the invariance thesis to human beings in the first place.

9.3 *Invariance and human cognition*

A considerable number of authors who apply asymptotic analysis to human cognition do not explicitly justify the assumption that the invariance thesis and simply assume that polynomial time in the standard class is of significance in the study of human cognition, especially in game theory.⁷ However, it is likely that they are committed to it, since they use polynomial time in the standard model as their criterion of feasibility. In particular, we need (even if this seems a bit pedantic) to exclude models that, for instance, compress the execution of any algorithm into one step.

One could offer a purely conceptual *a priori* argument for the application of the invariance thesis. In outline: certain processes are constitutive of human language: production, comprehension and acquisition. These are cognitive processes that map from inputs to outputs—e.g. from acoustic or visual signals to representation of linguistic information; they are *ipso facto* computations. RISTAD (*Game*: 1) seems to offer something like this argument, as do CLARK and LAPPIN (*Nativism*: § 4.1) in the context of language acquisition. Clark and Lappin, for instance, argue that since the child is an ‘information processing system’, it is subject to laws governing computational systems, which impose theoretical bounds on answers to the empirical question of how children learn language. They suggest that the general laws of physics (in particular, aerodynamics) analogously constrain reasonable answers to the ques-

7 PAPADIMITRIOU and YANNAKAKIS, ‘Complexity as bounded rationality’: 727; DASKALAKIS et al., ‘The complexity of computing a Nash equilibrium’: 90; CONITZER and SANDHOLM, ‘New complexity results about Nash equilibria’: § 1; PAPADIMITRIOU, ‘The Complexity of Finding Nash Equilibria’: § 2.1; GILBOA and ZEMEL, ‘Nash and correlated equilibria’: § 1.1. Interestingly, although CHERNIAK (*Minimal rationality*: § 4.4) doesn’t flag this assumption, he expresses some scepticism in the next step of the argument (§ 4.9).

tion ‘how do birds fly?’, even though that is an empirical question.

The invitation, then, is to reject out of hand any account of language or linguistic phenomena that does not conform, putatively, to the laws of computation, whence asymptotic analysis.

The problem is that, at least conceptually, all sorts of models of computation with different runtimes for important problems are reasonable, and none has any monopoly on the concept of computation. Therefore, merely *qua* information-processing system, the child could have access to all sorts of oracles for computable but putatively intractable problems.

van ROOIJ et al. (*Cognition*: § 9.12) do explicitly subscribe to the invariance thesis, and offer a few other arguments. They note wide acceptance in cognitive and computer science of the thesis, and seem to take this to show that it is sufficiently obvious that no explicit argument is needed. I agree with a spirit of epistemic humility according to which philosophers should generally defer to scientists on their areas of expertise. However, it is far from obvious that, for instance, van EMDE BOAS (‘Models’) had human cognition in mind, and no evidence is offered for this.

There is, however, something of a suggested *reductio*:

Practice 9.12.1 ...What would be the ramifications if the Invariance thesis would turn out to be false? (Hint: See Chapter 4.)[sic]

Given this is a supposedly pedagogical exercise, the argument is meant to be obvious, although unfortunately I find it needs some reconstruction. As I understand it, the *reductio* has the following structure. As a background assumption: human cognition is physically realised, and therefore ‘reasonable’.

1. Suppose (for *reductio*) that the invariance thesis does *not* apply to human cognition.
2. Then there is a ‘physically reasonable’ model of computation that cannot be simulated within a polynomial bound by members of the standard class.
3. A model of computation that is not simulable in a polynomial bound by members of the standard class can solve NP-complete problems in polynomial time. (I take it this is what the ‘hint’ alludes to, since polynomial-time reductions are the topic of chapter four.)

4. Then there is a physically reasonable mechanism by which to solve NP-complete problems in polynomial time.
5. But this is very implausible.

The problem with this argument is in the third premiss. The (approximate) consensus in theoretical computer science suggests that the third premiss is false; there are models of computation that are ‘faster’ than the standard class by a super-polynomial bound but cannot efficiently compute NP-complete problems. One such is the class of problems that is solvable in polynomial time on a *quantum* computer (van ROOIJ et al., *Cognition*: 200).⁸

However, I shall accept the invariance thesis on different grounds.

1. Suppose for *reductio* that the invariance thesis does not apply to human cognition.
2. Then there must be some physically plausible mechanism by which computational processes occur in the brain that are not simulable within a polynomial bound by members of the standard machine class.
3. There are no plausible candidate such mechanisms.

Therefore, provisionally, we should apply the invariance thesis to human cognition.

The most difficult premiss to justify is the third. The leading candidate is quantum-mechanical computation. The mainstream view is that quantum-mechanical dynamics are not relevant to the study of cognition (LITT et al., ‘Is the Brain a Quantum Computer?’; TEGMARK, ‘Importance of quantum decoherence in brain processes’). We will make this assumption in the remainder of the thesis.

We should distinguish three related but distinct theses.

1. The computations the brain undertakes are (on occasion) of a quantum nature.
2. Quantum-mechanical computation in the brain confers superpolynomial asymptotic speedup.
3. That speedup is sufficient to efficiently solve NP-complete problems.

8 Their gloss of how a quantum computer works is a little simplistic, but see also AARONSON, ‘Physical reality’.

The consensus in the natural sciences is roughly that even the first is physically implausible (*ibid.*). For the sake of argument, suppose we grant it. Even if there are quantum-mechanical computations of cognitive importance, it doesn't follow that they are any *faster* (the second thesis). If they are superpolynomially faster, that would invalidate the application of the invariance thesis to the brain. Is there such a superpolynomial speedup? YUNGER HALPERN and CROSSON ('Posner model') raise the question explicitly but do not offer any clear suggestion; the physical plausibility of the underlying physical theory is also in question (AGARWAL et al., 'Posner molecule'). WIEST ('Quantum microtubule substrate': 4) do suggest a 'quantum advantage' in the orch OR framework. However, the orch OR framework appears to be physically implausible (DERAKHSHANI et al., 'Crossroad'; REIMERS et al., 'Fröhlich condensation': 4222; MCKEMMISH et al., 'Penrose-Hameroff proposal'). Now, supposing that there is a superpolynomial speedup, there is some reason to think that that would not bridge the gap with NP-complete problems (AARONSON, 'Physical reality': §§ 3-4), although no definitive result has yet been shown. Our discussion of NP-completeness would only be misplaced if all three theses were true.⁹

9.4 *The Cobham-Edmonds thesis.*

We can now return to the question of whether the thesis is merely a mathematical curiosity or whether it is ultimately of interest in the analysis of tractability.

Appeals to asymptotic analysis usually proceed via the so-called Cobham-Edmonds thesis.

9.2. *Conjecture* (Cobham-Edmonds thesis). A function is feasibly computable if and only if some machine with polynomial runtime computes it (COBHAM, 'Difficulty'; EDMONDS, 'Paths').

The Cobham-Edmonds thesis is only meaningful relative to a model of computation. But, if the invariance thesis is correct,

9 But the truth of the first two theses would pose some difficulties for learning theory (ARUNACHALAM and de WOLF, *A Survey of Quantum Learning Theory*).

any reasonable model of computation will yield the same class of feasibly computable functions.

This then allows us to formulate the following.

9.3. *P-time cognition thesis* (van ROOIJ, ‘Tractable’: § 3):
‘tractable’ in the tractable cognition thesis is constrained by polynomial time.

The Cobham–Edmonds thesis might seem to admit obvious counterexamples. For instance, an algorithm that takes $\Theta(n^{2^{100}})$ time seems obviously infeasible. Less obviously, there is some reason to suspect that some problems that likely have no polynomial-time algorithm are feasible. In this case, we might nevertheless hold that, *most of the time*, the Cobham–Edmonds thesis is correct. For instance, there are surprisingly few ‘galactic’ algorithms, i.e. algorithms with polynomial asymptotic runtime that in practice are unusable.¹⁰ We can therefore provisionally take the Cobham–Edmonds thesis to be a (defeasible) heuristic in the sense above.

What justifies the Cobham–Edmonds thesis in general? In my view, there are two principal reasons it is generally accepted as a defeasible heuristic. The first is that there are few known problems for which the best-known algorithm is some particularly large polynomial (e.g. n^{1000}). Although artificial such problems can be constructed (HENNIE and STEARNS, ‘Two-Tape Simulation of Multitape Turing Machines’: theorem 3), it does not follow that problems examined practice are of any interest (see also DEAN, ‘Complexity’: n 10). This applies to both humans and machines.

However, humans and machines differ in another respect: current computers have improved by many orders of magnitude from the first computers (MOORE, ‘Cramming’). This matters because it ensures that counterexamples, say, two decades ago to the Cobham–Edmonds thesis may no longer be counterexamples now; but they may remain counterexamples vis à vis humans.

¹⁰ HELFGOTT, *Isomorphismes de graphes en temps quasi-polynomial (d’après Babai et Luks, Weisfeiler-Leman...)*; LIPTON and REGAN, ‘David Johnson’

Suppose, for instance, that an algorithm has runtime 1.1^n . On relatively small input sizes, it will do better here than e.g. n^3 : $160^3 \approx 1.1^{160}$. So if n is (e.g.) mostly less than 100 in the cases in which we are interested, we seem to have a counterexample. If n increases, however, we approach the point where the seeming advantages of the exponential algorithm no longer matter. The interesting feature of *computer engineering* is that n does, in fact, increase. As hardware improves, the range of input sizes within our ambitions expands. As this process continues, seeming counterexamples cease to be counterexamples, and fall within the scope of Cobham-Edmonds. Similarly, polynomial runtimes that seem to large may come within the scope of our hardware.

This is not merely a theoretical possibility. For many years, asymptotically, the best known algorithm for matrix multiplication (STRASSEN, ‘Elimination’) was considered to be superior to other algorithms only on inputs of such size as to be of theoretical interest (CORMEN et al., *Introduction*: 744). Through a combination of algorithmic and hardware improvements, it is now in common use (HUANG et al., ‘Algorithm’).

9.5 *Parametrised complexity.*

We shall have to slightly complicate the picture, following developments in complexity theory. Given an instance of a problem, we know its size. For instance, we know the length of a list. Usually, however, we know a little more about the nature of the instance of a problem. For instance, we might know that all the numbers in the list are (in a degenerate case) the same, in which case we have to do much less work (i.e. no work at all, since we don’t need to sort the list at all).

In other, less degenerate, cases, we can know something useful and come up with a slightly more nuanced view of the complexity of a problem.

9.4. *Definition.* A *vertex cover* for a graph $G = \langle V, E \rangle$ is a set of vertices that contains least one endpoint of every edge $e \in E$.

- 9.5. *Example* (vertex cover). Instance: a graph $G = \langle V, E \rangle$.
Question: does G have a *vertex cover* of size at most k ?

The natural view of the size of the problem is $|V|$, the number of vertices. And the problem is known to be NP-hard (KARP, 'Reducibility among Combinatorial Problems'). However, we have an algorithm that runs in time $O(1.2738^k + kn)$ time (CHEN et al., 'Improved upper bounds for vertex cover'). This means that if we hold k *fixed*, the algorithm runs in polynomial (indeed, linear) time. More generally:

- 9.6. *Definition* (fixed parameter tractability, DOWNEY, 'A Parameterized Complexity Tutorial': definition 1). A parameterised language \mathcal{L} is (strongly) fixed parameter tractable iff there is a computable function f , a constant c , and a (deterministic) algorithm M such that for all x, k ,

$$\langle x, k \rangle \in \mathcal{L} \text{ iff } M(x, k) \text{ accepts}$$

and $M(x, k)$ has running time at most $f(k)|x|^c$.

Many problems admit *fixed-parameter* tractable algorithms, even though they are NP-hard (*ibid.*).

van ROOIJ ('Tractable') propose substituting for the P-cognition thesis the *FPT*-cognition thesis:

- 9.7. In the tractable cognition thesis, fixed-parameter tractability bounds tractability from above.

10 The asymptotics of performance.

The Cobham-Edmonds thesis is now well-defined relative to a standard class of machine models which is applicable (up to a polynomial bound) to human cognition, if the foregoing is correct. Once this assumption is granted, we can restate the arguments informally stated in ¶ 9.3. For instance: computationally minded Chomskyans argue (for *reductio*) that 'blank slate' theories of language acquisition imply that acquisition is NP-complete.

I propose to dispute the application of the Cobham–Edmonds thesis to human cognition; in particular, I shall argue that its principal justification in *computer engineering* does not extend to cognition. That is: I do not propose to dispute previous arguments in the literature to the effect that certain problems have superpolynomial complexity based on widely believed *conjectures* in theoretical computer science. Rather, I dispute the Cobham–Edmonds as an *heuristic*; I do not wish to deny any widely believed conjectures in computer science (e.g. that $P \neq NP$).

A preliminary point is that Moore’s law does *not* apply to human beings. Assuming that apes bound human cognitive capacity from below in working memory, we can say that the difference is somewhere between 2 ± 1 (apes) and 7 ± 2 for humans (READ et al., ‘Working memory’). ZHENG and MEISTER (‘The unbearable slowness of being’) suggest that humans process information at a surprisingly low speed of 10 bits per second. There is also surprisingly little ecological variation in the information-theoretic efficiency of human languages (COUPÉ et al., ‘Different languages, similar encoding efficiency’).

So, even on an optimistic view of the overall growth of human cognitive capacities, nothing like Moore’s law applies, and the input sizes may remain small where in computer engineering they become large.

Perhaps it is useful to give an example of an argument that is conjectural rather than heuristic. Suppose, with SEARLE (*Rediscovery*: 200) that

[T]here [is] some description of the brain such that under that description you could do a computational simulation of the operations of the brain...given Church's thesis that anything that can be given a precise enough characterization as a set of steps can be simulated on a digital computer...in the same sense in which weather systems, the behavior of the New York stock market, or the pattern of airline flights over Latin American can.

TURING (‘Computability’: § 11) showed that no Turing machine decides the *Entscheidungsproblem*. If we follow Searle’s rendering of Church’s thesis, it follows that the brain does not decide the *Entscheidungsproblem* either. I suggest there is no obvious application of an *heuristic* here. A fairly natural computationalist view is that these results from computability theory apply

straightforwardly to computation generally, and, therefore to cognition, and so to language computations.

The situation here is somewhat different. Of course, there are *some* counterexamples to the Cobham–Edmonds thesis. But we are evaluating the application of the thesis as an *heuristic*, which may have general if not total validity. So merely pointing out some counterexamples is not enough.

I will sketch here two ways in which the Cobham–Edmonds thesis can mislead. My claim is not that it *always* misleads; indeed, I think Cobham–Edmonds can fruitfully be applied both in computer engineering and cognitive science. But it misleads often enough that the details matter in the ways I describe.

10.1 *Small (enough) can be interesting (enough).*

On small inputs, even exponential bounds are not prohibitive. This much everybody accepts (van ROOIJ et al., *Cognition*: § 9.9).

Why is this considered an uninteresting suggestion?

[E]ven though inputs may seem small when considering toy domains in lab settings, they cannot be assumed small in the real world.

This, of course, has to be examined on a case-by-case basis. The bifurcation of runtimes into polynomial and superpolynomial obscures potentially cognitively meaningful differences between runtimes that are both (for instance) exponential.

10.1. Example. Suppose that one algorithm takes 2^n seconds to run according to one model of computation, but 4^n seconds to run according to another, and that in all cases taking longer than a minute is ruled out. What's the maximum input size given by each? $\log_2(60) \approx 6$, but $\log_4(60) \approx 3$. Therefore, the first algorithm can take inputs of maximum size 6, but the second 3. Now consider estimates of working memory. It seems that the difference between the former and the latter is the difference between an algorithm that can plausibly run on nearly all items in working memory and one that can't.

10.2. *Example.* The best known algorithm for 3-SAT has complexity $O(1.307^n)$ (HANSEN et al., ‘SAT’).

Therefore, it is not as though inputs ‘small [enough] in practice’ are so small that they are not of interest to cognitive science or philosophy of language. The asymptotic approach, therefore, could fail by ignoring these details.

10.2 *Typical cases might take less time than the worst case.*

The second way that the hypothesis could fail is that *typical* cases may take considerably less time to process than the *worst* case.

van ROOIJ et al. (*Cognition*: § 9.5) offer two responses. First, van ROOIJ et al. (*ibid.*: § 9.5) argue that ‘for a computation to be tractable, it must be tractable for all its possible inputs, including the worst-case input’. It is not obvious that we should subscribe to this thesis. Consider, for instance, the analogous principle applied to walking: it would be absurd to hold that ‘for [walking] to be tractable, it must be tractable for all possible [terrain], including the worst-case [terrain]’. It is possible that in the worst case, we simply fail (to walk, learn and so on). Of course, if the worst case is intractable, it is contradictory to then specifically claim that the worst case is tractable—but to say e.g. that a form of semantic processing that in the worst case really is intractable is generally ‘tractable’ need not be read as a claim about the worst case.

Their second suggestion is more promising.

If there were to exist any input...for which the cognitive theorist believed it could not or does not happen in practice[sic], then he or she should instead postulate a different computational level theory...where [the] input domain...is restricted so as to exclude those inputs not assumed to occur in practice.

In some sense, this must be true: if only some cases can tractably be dealt with, there must be some restricted input domain corresponding to the cases that can be processed in reasonable time. The question is whether it is reasonable to *posit* the existence of such a domain without fully stating what it is. After all, van ROOIJ et al. (*ibid.*) could retort that the burden of

proof, by default, lies with the party denying a generally accepted heuristic (in this case the Cobham–Edmonds thesis).

The view I advocate is that, in general, the Cobham–Edmonds thesis in the study of human cognition is not sufficiently vindicated that the burden of proof lies only with the Cobham–Edmonds thesis’s detractors. In a dialectic where there are competing views which vary on their merits, the seeming violation of the Cobham–Edmonds thesis according to one theory is not, in my view, decisive.

I want now to mention two cases where it was (and in one case is) reasonable to posit the existence of a restricted tractable domain even though we do not yet know what it is.

- 10.3. *Example* (linear programming, KORTE and VYGEN, *Combinatorial Optimization*: 51). Given a matrix $A \in \mathbb{R}^{m \times n}$ and column vectors $b \in \mathbb{R}^m, c \in \mathbb{R}^n$
- find a column vector $x \in \mathbb{R}^n$ such that $Ax \leq b$ and $c^T x$ is maximum,
 - decide that $\{x \in \mathbb{R}^n : Ax \leq b\}$ is empty, or
 - decide that for all $\alpha \in \mathbb{R}$ there is $x \in \mathbb{R}^n$ such that $Ax \leq b$ and $c^T x > \alpha$.

In the 1980s, the state of our knowledge was as follows. The ‘simplex’ algorithm worked well in practice but had an exponential worst-case runtime (KORTE and VYGEN, *Combinatorial Optimization*: § 3.2; CHERNIAK, *Minimal rationality*: § 4.9). Polynomial-time algorithms were known, but remain (to this day) ‘too inefficient to be used in practice’ (KORTE and VYGEN, *Combinatorial Optimization*: 73). CHERNIAK (*Minimal rationality*: § 4.9) therefore speculated that the best response was the empirical investigation of ‘which are the interesting and difficult cases’ of a problem, in order to establish whether there is ‘real-world relevant complexity at least as an empirical hypothesis’ in addition to whether there is *worst-case* intractability. He had formulated no clear or precise hypothesis as to the restricted domain on which ‘real-world relevant complexity’ might arise. Cherniak offered various hypotheses about the asymptotic profile of the difficulty of a problem: in the worst case, perhaps *every* instance of a problem might be susceptible to superpoly-

nomial blowup; in other cases, it might turn out that only *uninteresting* or somehow pathological cases are so susceptible; or perhaps some interesting or quotidian cases also fall into that population, but there are very few.

The state of knowledge at the time did admit *experimental* study of the behaviour of the simplex algorithm (MCCALL, ‘Simplex performance’). On these grounds, at the time, I suggest, it was in fact reasonable to accept as a working hypothesis the last: nearly all interesting cases of linear programming (e.g. for use in industry or natural science) in practice took polynomial rather than exponential time. This can be justified on the broadly naturalist grounds that that is precisely the hypothesis accepted by computer scientists; at least by the lights of van ROOIJ et al. (*Cognition*), that is a reasonable argument, since they also advocate deference to computer scientists in their field of expertise.

It so transpires that we have the sort of restricted domain that van ROOIJ et al. (*ibid.*) sought. In 2001, SPIELMAN and TENG (‘Smoothed analysis of algorithms’) offered a rigorous ‘smoothed’ asymptotic analysis of the simplex algorithm for linear programming. Although in the worst case the simplex algorithm takes exponential time, if the input is perturbed randomly to an arbitrarily small extent, the runtime is (to a first approximation) polynomial. So van ROOIJ et al. (*Cognition*) and everybody else would *now* accept that if, for instance, it is suggested that a cognitive capacity involves linear programming, but the inputs are randomly perturbed, the tractable cognition thesis would not straightforwardly rule out that account of cognition. The dispute concerns whether it was reasonable between 1947 (when the simplex algorithm was first formulated) and 2001 (when smoothed analysis explained simplex’s performance) to accept the tractability on simplex on everyday instances of linear programming, to a certain extent on trust; I claim that that is so.

An example in which no satisfactory account has yet been given is the the practical use of SAT solvers in industrial applications. SAT is NP-complete (MARQUES-SILVA, ‘Applications’). Moreover, many natural subclasses of cases in fact plausibly *must* take exponential time (CHVÁTAL and SZEMERÉDI, ‘Many

hard examples for resolution’). We do not yet have an equivalent to smoothed analysis to explain the industrial utility of SAT solvers.

It is folklore in theoretical computer science about how super-polynomial worst-case runtime is compatible with tractability in practice in the case of SAT. The view is that there is some sort of underlying structure—we may not yet be able to precisely articulate it—to ‘real-world’ or plausible inputs that makes them ‘non-adversarial’ (MALIK and ZHANG, ‘Boolean satisfiability from theoretical hardness to practical success’: 78). It is, nevertheless, reasonable to generalise that modern SAT solvers will ‘routinely solve...industrial SAT instances with millions of variables’ even though we have no understanding of why the specific sets of heuristics employed by modern SAT solvers are so effective in practice’ (VARDI, ‘Boolean satisfiability’). At least, it is as reasonable as generalising e.g. that bridges do not fall down, the ‘Grand National [does] start and... hotels [do not] fall into the sea’.

Here, van ROOIJ et al. (*Cognition*: § 9.9) would suggest that the correct response here would be to attempt to show that SAT is fixed-parameter tractable. However, we have no useful results in this respect, and not for want of trying (GANESH and VARDI, ‘On the Unreasonable Effectiveness of SAT Solvers’) moreover, if a large and natural subclass of SAT were to be discovered, that would be industrially very useful, so there is no want of commercial incentives. The central problem is that parametrisations that would make SAT instances easy turn out not to correspond to the easy instances we observe in real-world instances SAT—these parameters do not remain small in the real-world instances. The opposite approach would be to examine real-world instances and to try to work backwards to a generalised parametrisation explaining why they are easy; but these cases do not generally easily submit to theoretical study (*ibid.*: § 25.5).

Should we accept this folklore from computer science, as I suggested above? Dialectically, van ROOIJ et al. (*Cognition*) and most others who appeal to asymptotic analysis must agree, because they generally rely on conjectures to show that the problems they claim are intractable really are intractable. For in-

stance, van Rooij et al. (*ibid.*) need to assume that $P \neq NP$ for NP-hardness to have any significance to cognition. If we reject conjectures from computer science, it is not obvious why we should accept the underlying asymptotic machinery that motivates asymptotic precisifications of the tractable cognition thesis.

10.3 *Recapitulation.*

I shall suggest that some combination of these two failures of Cobham–Edmonds arise in linguistic cognition. Accordingly, the soundness of numerous arguments appealing to it as a premiss in the study of language can be questioned. It does not, of course, follow that their conclusions are false, although I shall suggest in some cases there is independent pressure to accept that view.

11 **The asymptotics of competence.**

The response to asymptotic arguments about performance above amounted to the claim that, on occasion, the instances of asymptotically hard problems we actually encounter either may be too small or otherwise lack the requisite structure to be prohibitively difficult in practice. *Competence*, insofar as it differs from performance, therefore leaves open the prospect of leaving those hard cases within the domain a theory covers. If we can expect a theory to account for centre-embedding of arbitrary depth, we can no longer escape asymptotic difficulty by excluding such instances from the domain over which a theory's fecundity is assessed.

The problem here, however, is that it is not obvious that competence is also accountable to performance limitations. Asymptotics could be utterly irrelevant to the study of competence: it wouldn't matter whether

IV Asymptotics and compositionality.

In this chapter, I examine whether an asymptotic argument can be given that meaning is compositional. The argument purports to fill a lacuna in Davidson's *learnability* argument for compositionality (§ 12). Pagin claims to fill a lacuna in the argument with asymptotic analysis (§ 13). I then ask whether the argument is targeted at performance or competence; if it is targeted at performance, it is misplaced; but if it is targeted at competence, the picture is more complicated (§ 14).

12 Compositionality from learnability.

To DAVIDSON ('Theories of Meaning and Learnable Languages': 8), a theory of knowledge of language must 'specify, in a way that depends effectively and solely on formal considerations, what every sentence means'. I take it that Davidson views this requirement as entailing (or even equivalent to) the requirement that the meaning of a sentence is, on this view, 'a function of a finite number of features of the sentence'. Otherwise, a speaker would fail to master the meaning of certain sentences, no matter how many other sentences they learn to produce and understand: the meaning of the unlearnable sentences would not depend on that of the sentences already mastered, no matter which sentences those are.

12.1. A 'semantical primitive' is an expression such that 'the rules which give the meaning for sentences in which it does not appear do not suffice to determine the meaning of the sentences in which it does appear' (*ibid.*: 9).

Davidson then suggests that a *learnable* language must have a *finite number of semantical primitives*. On a standard view, Davidson offers here an argument for the compositionality of meaning. PAGIN ('Compositionality, computability, and complexity': 551) informally puts that requirement thus.

12.2. The meaning of a complex expression is a function of the meanings of its parts and a mode of composition.

In particular, this entails that the substitution of synonymous terms should not change the meaning of a complex term.

There is some dispute as to whether that is actually what Davidson meant; in his reply to PAGIN ('Radical interpretation and compositional structure'), Davidson writes that he took compositionality 'as a given' and 'built in at every stage'.¹¹ But the exegetical debate need not concern us.

According to Pagin, the argument is abductive. The *explicandum* is that a speaker 'can step by step work out the meaning of new grammatical sentences' (PAGIN, 'Compositionality, computability, and complexity': 551). The *explicans* is compositionality. Pagin objects to the abduction, and suggests further argument is required.

The problem in this abduction is that there may be cases where we can work out the meaning of a complex expression even if the replacement of one of the constituents by a synonymous expression could change the meaning of a whole, violating compositionality. What is required is not that the semantics should be *compositional* but merely *computable*.

Here, PAGIN (*ibid.*: 553) proposes a further constraint.

12.3. *Definition* (TRC). Any plausible semantics for natural language is computationally tractable.

The argument is then, to a first approximation, that compositionality is a good *explicans* because learnability requires tractability, and tractability requires compositionality. This, therefore, improves on the previous abduction: it is true that learnability requires computability, but computability doesn't require compositionality; and learnability does not, *directly*, require compositionality.

13 **Compositionality from tractability.**

Pagin argues for this view in terms of a symbolic computational model: term rewriting. The claim is that noncompositional semantics has exponential difficulty in the term rewriting régime. This is subject to various qualifications, the most important of which is that this applies only to languages whose syntax is combinatorially unrestricted.

¹¹ HECK, 'Does Davidson Argue for Compositionality?': 2; MALPAS, 'Donald Davidson': § 3.1; PAGIN and WESTERSTÅHL, 'Compositionality II': § 1.1

Considerable work is required to reconstruct the result, because it is shown only by example. Godzilla, a noncompositional rewrite system (*ibid.*: § 8.3.3) has factorial complexity; from this, it is inferred that all noncompositional systems (meeting certain criteria) have superpolynomial complexity. Although there is some indication of the generalisation of the result (*ibid.*: 580), there is no formal indication of the general result. In this section I offer a general result in the spirit of Pagen.

13.1. *Definition (ibid.: Definition 1).* A *grammatical term algebra* for a language L is a triple $\langle \text{GT}_L, \text{AT}_L, \Sigma_L \rangle$, where AT_L is a finite set of atomic terms, Σ_L a finite set of *partial* syntactic operations, and GT_L the closure of AT_L under Σ_L .

13.2. *Remark.* Ristad is not concerned by parsing. It is therefore assumed that we are given syntax trees rather than unparsed strings.

Semantic functions assign meanings to grammatical terms. Pagen takes no firm view of what meanings actually are, and simply assumes that there is some set of such meanings.

13.3. *Definition (ibid.: Definitions 2-5).* A semantic function $\mu : \text{GT}_L \rightarrow M$ is *standard compositional* iff for each n -ary $\sigma \in \Sigma_L$ there is a *meaning operation* $r_\sigma : M^n \rightarrow M$ such that, whenever defined,

$$\mu(\sigma(t_1, \dots, t_n)) = r_\sigma(\mu(t_1), \dots, \mu(t_n)).$$

A *general compositional* system replaces μ by a finite family of semantic functions together with a selection function Ψ fixing which member evaluates each argument place:

$$\mu_i(\sigma(t_1, \dots, t_n)) = r_{\sigma,i}(\mu_{j_1}(t_1), \dots, \mu_{j_n}(t_n)),$$

with $j_k = \Psi(i, \sigma, k)$.

13.4. *Remark.* PAGIN (*ibid.*: § 2.3) introduces general compositionality to accommodate ‘switcher semantics’ by which

the linguistic context may affect the meaning of a sentence.

Pagin then argues that Davidson's argument only licenses a much weaker recursivity property: i.e. that meanings should be definable by primitive recursion over the syntactic structure.

13.5. *Definition.* A semantic function μ is *recursive* if it is definable by primitive recursion over the term algebra. Then the clause for a complex term may take the parts themselves as additional arguments:

$$\mu(\sigma(t_1, \dots, t_n)) = r_\sigma(\mu(t_1), \dots, \mu(t_n), t_1, \dots, t_n).$$

The additional arguments t_1, \dots, t_n make this a weaker requirement than compositionality, because even if $\mu(t_1) = \mu(t_1')$, if $t_1 \neq t_1'$ we have no reason in general to require that $\mu(\dots, t_1, \dots) = \mu(\dots, t_1', \dots)$.

14 Is competence or performance the *explicandum*?

Two questions arise in evaluating Pagin's challenge. *First*, does Pagin believe in distinguishing competence and performance, à la Chomsky? *Second*, if so, is his argument aimed at competence or performance?

We shall follow the distinction between competence and performance, following CHOMSKY (*Aspects*: 3 ff) and the subsequent literature, and therefore distinguish two candidate *explicanda* in Pagin's argument.

14.1 Performance.

Suppose, then, that the *explicandum* of Pagin's argument is performance. We can then make its abductive structure more precise. The *explicandum* is our understanding of the meaning of sentences that have actually been produced. Pagin's argument is then that compositionality follows from the best explanation of our understanding of actually produced and comprehended sentences.

It is useful first to examine cases where compositionality is allegedly violated. Consider propositional attitudes (PICKEL and SZABÓ, ‘Compositionality’: § 4.2.5).

(1) Carla believes that biblioklepts steal.

(2) Carla believes that book thieves steal.

On one view, although ‘biblioklepts’ and ‘book thieves’ are synonymous, (1) and (2) may differ in truth value (and hence meaning).

Now, suppose we formulate some general theory of the semantics of propositional attitudes that is noncompositional, and that it satisfactorily and perhaps otherwise virtuously explains natural language data; suppose, moreover, that it is recursively specified, and therefore Davidson’s original challenge to specify the meaning of propositional attitude reports as a ‘function of a finite number of features’ of those reports is met.

The problem, according to Pagin, is that *more complex* expressions’ meanings rapidly take longer to work out, unless their meanings are compositional. But if he is targeting *performance*, he needs to explain why these more complex expressions appear in ‘actual use of language’.

A simple response might be that the depth of nested propositional attitude reports is bounded, and so a superpolynomial increase in difficulty as the depth of the nesting increases is irrelevant to actual use of language (§ 10.1). That would be mistaken. For instance, the following sentence’s meaning is perfectly comprehensible.

(3) Alice knows that Bob knows that Carol knows that David knows that Eve knows that Frank knows that Grace knows that Heidi knows that Ivan knows that it will rain tomorrow.

This is a ninth-order report. However, both parsing-wise and semantically, (3) is fairly simple.

A more sophisticated response, I suggest, is that although the depth of typical cases may in some cases be quite high, their structure will generally enable comprehensibility (§ 10.2).

15 A connexionist challenge.

v Asymptotics and language acquisition.

16 PAC-learnability: formalism.

In this section, we shall be concerned by certain arguments from *learning theory*—a branch of asymptotic analysis—to justify the view that we have innate knowledge of language in the form of universal grammar due to poverty of the stimulus. I shall argue that the PAC-learning régime fails for roughly the reasons suggested in § 9.

We shall follow here the formalism given by KEARNS and VALIANT ('Limitations': § 2).

16.1. *Definition* (instance spaces). A *domain* or *instance space* is some set X of encodings of the objects of interest in a learning problem.

16.2. *Example*. There is an instance space of *human beings*.

16.3. *Definition* (concepts). A *concept* is some subset of X . A *concept class* is a class of concepts.

16.4. *Example*. One *concept* might be that of tall people, given the instance space of people.

In order to perform computations about concepts, we need to *represent* concepts.

16.5. *Definition* (representations). A *representation class* over X is a pair $\langle \sigma, C \rangle$ such that $C \subseteq (\{0, 1\} \cup \mathbb{R})^*$ and $\sigma : C \rightarrow 2^X$. We shall then write that $\sigma(c)$ for $c \in C$ is a *concept over X* . Then the image $\sigma[C]$ is a *concept class represented by $\langle \sigma, C \rangle$* .

Learning theory is about how we can algorithmically *learn* from *examples* drawn from an instance space. So, informally, suppose that we are interested in learning the meaning of 'tall'. We could learn this by examining who is called 'tall' (and who is

not). There are therefore ‘positive examples’ of ‘tall’ (i.e. people who count) and ‘negative examples’ (i.e. people who don’t).

16.6. *Definition* (examples). The *positive examples* $\text{pos}(c) = \sigma(c)$ of a concept $c \in C$ are those that fall under the concept. The *negative examples* $\text{neg}(c) = X - \sigma(c)$ are members of the instance space that do not fall under that concept. We will write $c(x)$ for an indicator function i.e. $c(x) = 1$ iff $x \in \text{pos}(c)$ and 0 otherwise.

We will sometimes ‘parametrise’ representation classes and instance spaces.

16.7. *Definition* (parametrisation). A *parametrised* representation class can be given from a stratified domain $X = \bigcup_{n \geq 1} X_n$. Then we have representation classes C_1, C_2, \dots . n may measure how complex concepts in $\sigma(C)$ are.

16.8. *Example*. Let X_n be the set $\{0, 1\}^n$ and C_n the class of Boolean formulæ over n variables with length at most n^2 . We can then take $\sigma(c)$ to contain all models of the formula c .

Intuitively, the task of a learning algorithm is to offer an hypothesis that allows us to classify new example drawn from X . But if the hypothesis only inefficiently allows us to classify a new example, the learning algorithm is not very useful. Therefore, we are mostly interested in polynomially evaluable representation classes.

16.9. *Definition* (polynomial evaluability). A representation class C over X is *polynomially evaluable* if for each $c \in C$ there is some polynomial-time algorithm that decides whether any $x \in X$ is in $\text{pos}(c)$.

We learn from examples that are *labelled* either as positive or negative, i.e. as falling under a concept or not.

16.10. *Definition* (labels). A *labelled example* from X is a pair $\langle x, b \rangle$ where $x \in X$ and $b \in \{0, 1\}$. A *labelled sample* $S = \langle x_1, b_1 \rangle, \dots, \langle x_m, b_m \rangle$ is a finite sequence of labelled examples.

16.11. *Definition* (labelled examples of concepts). A *labelled example of* $c \in C$ where C is a representation class is some example $\langle x, c(x) \rangle$, and a *labelled sample* is as above. A *positive* or *negative* sample contains only positive or negative examples respectively.

16.12. *Definition* (agreement/consistency). A representation h *agrees* with an example $\langle x, b \rangle$ when $h(x) = b$. A representation h is *consistent* with a sample if it agrees with every example within.

16.13. *Definition* (the learning régime). In the standard régime in computational and statistical learning theory, a learner is given a representation class C and examples of some particular representation $c \in C$. For instance, the representation class might correspond to all words for height in English, and the particular representation might one of 'tall'. Let us say that 'tall' (or c) in general is the *target representation*.

There are two (for our purposes equivalent) formulations of how the samples are drawn.

- *Separate distributions*: The examples are drawn from a fixed but arbitrary probability distributions. D_C^+ is a fixed but arbitrary probability distribution over $\text{pos}(c)$, and D_C^- over $\text{neg}(c)$. The learner has access to two oracles, which draw labelled samples from the positive and negative distributions with constant cost. That is to say: drawing samples is not *costless* but rather has linear cost in the number of samples.
- *Same distribution*: the examples are drawn from a single target distribution over the whole domain.

For simplicity we shall assume that the examples are drawn from the same distribution, and therefore amend the following exposition slightly.

16.14. Definition (error). We can now measure the *error* of an hypothesis represented by some h relative to the distribution as $\text{err}(h; c, D) = \mathbb{P}_{x \sim D}[h(x) \neq c(x)]$.

When $\text{err}(h; c, D) \leq \varepsilon$, h is ε -good; otherwise, it is ε -bad. Note that h need not be deterministic.

Informally, we will now require that

16.15. Definition (learnability from examples). Suppose that C and H are representation classes over X . Let ε be some arbitrarily small error rate (the accuracy parameter), and let δ be some arbitrarily small probability of failure to formulate a hypothesis that has an error of at most ε (the confidence parameter). (Thus, ‘probably $(1 - \delta)$ approximately correct $(1 - \varepsilon)$ ’.)

We say that C is *learnable by H* iff there is a probabilistic algorithm A such that,

- with access access to an oracle giving labelled examples according to D ,
- input ε and δ ,
- for any target representation $c \in C$ and distribution D over X ,

A halts and outputs a representation $h_A \in H$ that with probability at least $1 - \delta$ satisfies $\text{err}(h; c, D) \leq \varepsilon$.

C is *polynomially learnable* if there is some polynomially evaluable H .

C is *efficiently polynomially learnable* if A has running time polynomial in n , $\text{size}(c)$, $\frac{1}{\varepsilon}$ and $\frac{1}{\delta}$.

17 PAC-unlearnability: results.

KEARNS and VALIANT (*ibid.*: § 4) defines certain *cryptographic problems* that are widely believed to be superpolynomial in difficulty. We shall omit the details here. What is important is that

if we assume that these cryptographic problems are hard, and we show that instances of the cryptographic problem can be reduced to instances of a learning problem with at most polynomial overhead, then the learning problem must also be (up to polynomial overhead) at least as hard.

Now, let us consider one way of regimenting the problem of learning a language given labelled examples of strings in and out of the language. We will restrict ourselves to the relatively small class of *regular* languages.

We shall take the instance space X to be $\{0, 1\}^n$, which can be viewed as a space of strings. We are then interested in acyclic finite automata that accept strings of length n drawn from X .

17.1. Definition. Let $\text{AFDA}_n^{p(n)}$ denote the class of deterministic finite automata of size at most $p(n)$ accepting only strings of length n , and put $\text{AFDA}^{p(n)} = \cup_{n \geq 1} \text{AFDA}_n^{p(n)}$.

Since we have omitted the cryptographic details, we will only state (informally) the result.

17.2. Proposition (informal). Assuming there is no polynomial-time algorithm to solve certain cryptographically hard problems (which is generally believed to be true), the following result holds: AFDA_n is not efficiently polynomially learnable (using any hypothesis class).

17.1 A deductive argument for universal grammar?

Perhaps surprisingly, a number of authors have interpreted this result as suggesting that there must be an innate universal grammar. For instance, NOWAK et al. ('Aspects') write that

Statistical learning theory also shows there is no procedure that can learn the set of all regular languages, thereby confirming the necessity of an innate UG.

Similarly, YANG ('The Great Number Crunch': 223) write that one conclusion from the learning-theoretic literature that 'emerges convincingly' is that learning is not possible unless the hypothesis space is tightly constrained by prior knowledge, which can be broadly identified as Universal Grammar'. More-

over, KODNER et al. (*Linguistics will thrive*: 5) reaches a roughly similar conclusion, and extends the claim to recent successes in training large language models, in response to PIANTADOSI ('Language models').

PIANTADOSI (*ibid.*) centrally claims that language models are perfectly good theories of language, and should substitute Chomskyan theories about grammar, competence and performance, and so on. He catalogues a series of successes they have had in conforming to standard grammatical rules, including those for embedded clauses, prepositional phrases, conjunctions, pronouns, determiners, quantifiers, adjectives, agreement and pronoun reference (*ibid.*: 3).

KODNER et al. (*Linguistics will thrive*), in response, argue as follows.

1. Learning theory 'clearly and firmly support[s]' the poverty of the stimulus argument.
2. If LLMS were to learn in an 'unconstrained' way, learning theory would show that to be possible only because 'they were trained on inhumanly large training data[sic]'.
3. Learning from a plausibly sized sample requires that we should constrain the hypothesis space.
4. Therefore, if 'small' language models ever are successfully created, they too will encode 'non-trivial structural priors facilitating language acquisition and processing'.
5. These 'biases, principles, and limitations' are 'some form of Universal Grammar'.

There are a number of ways in which this argument is significantly flawed. One is that it is overly optimistic about the utility of the PAC régime as a model of learning, which I address in § 20. However, it is worth flagging a few other problems first.

As PIANTADOSI ('Language models': 38 ff) note, KODNER et al. (*Linguistics will thrive*) have a very liberal view of universal grammar. In particular, it might turn out to be sufficient for the purposes of Kodner et al. that transformers work better than e.g. LSTMs. But this is hardly 'universal grammar' or 'innate knowledge' in the commonsensical sense of the word. We can distinguish three theses:

- that there is no inductive bias whatsoever necessary or used in learning language (which nobody thinks);

- that there is a restricted hypothesis class or an inductive bias corresponding to a non-uniform prior (which is common to everybody); and
- that e.g. there is some sort of symbolically specified set of principles and parameters required to acquire language, which PIANTADOSI ('Language models') denies, and on which KODNER et al. (*Linguistics will thrive*) takes no clear stance.

18 PAC-learnability: positive results.

19 An *abductive* argument for universal grammar?

20 Questioning the PAC-learning régime.

What justifies the PAC-learning régime as a model of the task of human language acquisition? Obviously, it is not the realism of its assumptions. For instance, although it is true that the vast majority of human beings succeed in acquiring their first language in a relatively short period of time as children, this does not mean that we are justified in setting the confidence parameter δ to be arbitrarily small. Similarly, although we clearly converge on grammars that allow us to have relatively satisfactory performance with respect to an idealised language of the community (at least sufficiently to communicate), the accuracy parameter is clearly not *arbitrarily* small; we can easily bound it by taking the number of mistakes produced by the most successfully grammatically pedantic person ever to speak a language and dividing the number of sentences uttered by them by the number of mistakes.

Many authors who cite results in the PAC-learning régime simply ignore any questions either about the realism of its assumptions or the implications of the obvious lack of realism of its assumptions, including KODNER et al. (*ibid.*), YANG ('The Great Number Crunch') and NOWAK et al. ('Aspects'). One author who does attempt to explicitly justify this view is de WOLF ('Applications': § 2.4.4). He notes that it seems a little odd to assume that ϵ and δ can be arbitrarily small, but insists the PAC régime is 'right *in spirit*', because, plausibly, given more time or data, a learner will have a better chance of formulating a hypothesis with fewer errors.

The problem with this argument is that it is an argument for *monotonicity* rather than *efficient learnability*. That is to say: additional samples may always improve (e.g.) accuracy, but that does not mean that they improve accuracy enough to ensure that accuracy can be improved with only polynomial additional cost. Indeed, the claim that additional data always improve accuracy is compatible with the claim that it is not possible at some point to eliminate some residual nonzero error: suppose for instance that after n data the best we can do is to have $\varepsilon = 0.01 + 2^{-n}$ —this is monotonic, but not arbitrarily small. In addition, it faces some of the standard objections above: ε and δ may be boundedly small, and exponential difficulties may arise only later.

It may be plausible that δ should be considered to be arbitrarily small. However, it is not so obvious that ε should be considered to become arbitrarily small relative to the target language (e.g. that of one's parents).

20.1. *Remark.* Insert some material on language evolution here.

If the PAC-learning framework cannot be justified by the plausibility of its assumptions or *a priori*, perhaps it can be justified by its empirical fruitfulness. In fact, this much is not obvious, because of the general paucity of *positive* PAC-learnability results that correspond to clearly articulated restrictions on the hypothesis space that correspond in turn to *pkillnatural* language (CLARK and LAPPIN, *Nativism*: §§7.3-7.4). As they say, the situation is 'from a theoretical point of view...completely unsatisfactory': '[w]e seem to be reduced to saying that we have algorithms that work when they work, and don't otherwise'.

20.2. *Remark.* Todo: properly review their positive learnability results and whether they have anything to do with anything interesting.

Clark and Lappin do present some positive PAC-learnability results, but, as they note, these fall short of natural language. We cannot therefore say that PAC learnability has earned its keep

through its theoretical fruitfulness, because we do not even have positive results that account for the fact that children learn their first languages. Moreover, the significance of their positive results is very unclear because it may be that they are overly restricted because they still care about arbitrarily small δ and ϵ : see e.g. § 8.3.1.

If we think of PAC-learnability as a criterion for separating hypotheses about learning language, all we have are various positive or negative results for classes of language that are relatively near or far away from natural language (either idealised or not) without any clear way of distinguishing which ones are really closer or further away. It's far from obvious how to infer from these anything about the inductive biases required to learn language. Plausibly, Piantadosi is right at least insofar as experimental work from language models is much more useful (see below).

21 Should Chomsky worry?

A broader question is whether the objections I have raised to computational variants of the poverty of the stimulus argument would unduly worry Chomsky.

To CHOMSKY (*Aspects*: 3), linguistic theory studies an idealised 'speaker-listener unaffected by 'grammatically irrelevant conditions' such as 'memory limitations' in 'applying his knowledge of the language in actual performance'. It is natural to include amongst the 'irrelevant conditions' constraints on performance. If so, Chomsky would be utterly uninterested in the asymptotic analysis of linguistic processing (as in the case of Ristad and Pagin), but might also be uninterested in the computational complexity of language acquisition. An asymptotic APS would be welcome support, but no more than that.

The logical structure of Chomsky's argument is rather different. CHOMSKY (*Knowledge of language*: 7) remarks on the

structure-dependence of rules, the fact that without instruction or direct evidence, children unerringly use computationally complex structure-dependent rules rather than computationally simple rules that only involve the predicate 'leftmost' in a linear sequence of rules.

1. I wonder who [the men expected to see them]
2. [the men expected to see them]

Both (2) and (3) include the clause bounded by brackets, but only in (2) may the pronoun *them* be referentially dependent on the antecedent *the men*; in (3) the pronoun is understood as referring in some manner indicated in the situational or discourse context, but not to the men. ...How does every child know, unerringly, to interpret the clause differently in the two cases?

The suggestion is then that the devices provided by the theory of UG are adequate to the descriptive task at hand' (*ibid.*: 51).

The abduction to universal grammar requires the auxiliary premiss that other learning mechanisms won't work (*ibid.*: 12).

Chomsky suggests there is 'little hope' of accounting for knowledge of language through 'analogy, induction, association, reliable procedures, good reasons and justification in any generally useful sense, or in terms of "generalized learning mechanisms" (if such exist)'. But this is a matter of empirical enquiry, and cannot be studied *a priori*. The specific methodology Chomsky proposes is to take as an *explicandum* the 'system of knowledge that has been attained', and to offer as putative *explicantes* 'properties [that] must be attributed to the initial state of the mind/brain'. There is a language faculty to the extent that 'these properties are language-specific'. Then examples such as those involving anaphoric dependence above ('them') are given to buttress this claim.

Although Chomsky himself might not think that this approach requires asymptotic supplementation, we need not accept this view. In particular, we might demand more rigorous evidence that there really is 'no hope'. A natural way to supply that argument is asymptotic, and so the alleged failure of the argument is of interest.

One question that advocates of an asymptotic variant of the APS often do not explicitly answer is whether their argument targets *competence* or *performance*. It seems that these arguments target competence. Consider again the argument given by NOWAK et al. ('Aspects': 614).

Statistical learning theory also shows there is no procedure that can learn the set of all regular languages, thereby confirming the necessity of an innate UG.

UG is a theory about competence, and formal grammars are used to theorise about competence in the first instance (and then indirectly performance). A somewhat puzzling question is

then whether it is legitimate for computationalists in the Chomskyan mould to appeal to performance constraints on acquisition of competence in the first place. This might not worry Chomsky particularly (although maybe it should). However, there is a question about how the computational Chomskians would justify appealing to asymptotic efficiency constraints in discussing acquisition if they reject any place for asymptotic analysis in the study of competence, following Chomsky.

VI Whither asymptotics?

Have there been any cases where asymptotic analysis usefully informed our understanding of language? I have suggested that a number of difficulties afflict attempts to rule out theories about language, which naturally invites the question of whether any attempt should survive such scrutiny, and whether asymptotic analysis has any place in our understanding of language generally.

22 Recapitulation: some principles from above.

I first sketch some rough principles by which we might make reasonable progress in understanding language by asymptotic means, by analogy with the study of reasoning in cognitive science.

In § 9, I made the following claims about the proper application of asymptotic analysis.

1. The parameter in terms of which superpolynomial complexity is derived should be shown to be large. So, for instance, if learning language becomes exponentially more difficult in terms e.g. of the number of vowels, but the number of vowels is constrained to be a small number, we cannot infer much from this exponential bound.
2. Naturally occurring cases should have superpolynomial complexity. That is, if it turns out that a parameter leading to superpolynomial complexity *in the worst case* nevertheless leads to relatively small increases in computational difficulty in practice, we again cannot infer much.
3. We should actually perform well on these naturally occurring cases leading to superpolynomial complexity. For instance, even if e.g. it were common to read long sentences of Legalese in public for amusement, it would be erroneous to infer much about children's language acquisition—because children probably wouldn't understand those long sentences of Legalese in the first place.

23 A speculative conclusion.

I want to advance a further positive hypothesis about one way in which asymptotic analysis might in fact

The very rough intuition is that superpolynomial complexity should, in general, lead to relatively sharp dropoffs in performance,¹² but at least typical polynomial complexity should make feasible a more gradual dropoff in performance as the size of an input increases. Under the heading of performance here I include speed, the solution of potentially difficult cases in a reasonable period of time, accuracy, and so on. Here are some examples. Some very long sentences are easy to parse and understand.

(4) Cows aren't green or red or brown or politicians or plumbers...

Other long sentences are more difficult to parse and understand.

(5) A person who, when riding a cycle, not being a motor vehicle, on a road or other public place, is unfit to ride through drink or drugs shall be guilty of an offence (Road Traffic Act 1972, s 19(1)).

Note, for instance, that (4) can be extended to a fairly great extent without loss of comprehensibility, whereas, even if (5) is intelligible, the following is likely unintelligible.

(6) A person who, when riding a cycle, not being a motor vehicle, the motor being driven by internal combustion, internal combustion not including diesel-electrical motors, on a road or other public place, not being a road designated in the schedule by the Secretary of State, is unfit to ride through drink or drugs shall be guilty of an offence.

Of course, our comprehension (on various measures) of (4) may eventually decline as the sentence is extended. But centre-embeddings like (5) seem to decline in comprehensibility very quickly.

¹² Here, I mean this in the colloquial sense, i.e. 'good' and 'bad' performance. I don't mean Chomskyan performance.

There seems here to be a rapid shift in the status centre-embeddings somewhere between depth two and four. On the other hand, there is no such rapid shift in the status of sentences like (4).

What happens in cases like (5) if there are too many centre-embeddings? We may simply give up. Or we may use certain heuristics to guess the meaning of the sentence, which may have degraded performance and fail to even reliably approximate the literal meaning of the sentence. Thus we have various corresponding measures of performance.

There is analogy here with physical science. Consider, for instance, heating up a solid substance (e.g. a metal). Generally, heating up a substance increases its volume. But heating up e.g. a steel girder from 20C to 25C does not produce any *dramatic* change. On the other hand, water takes up much more space when it is 101C than when it is 99C, because of a phase transition.

In a slogan: real-world superpolynomial complexity should entail sharp dropoffs in performance. How might this help the study of language or cognition? I suggest that in studying capacities such as language processing or acquisition, we ought to examine the difference in size in successful cases. So, for instance, if humans generally are unable to understand sentences with centre-embeddings of depth greater than three, and there is a sharp cutoff, that hypothesis is compatible with superpolynomial complexity. More generally, we can formulate (informally) an inconsistent triad:

1. wide range in the sizes of successful cases (e.g. size of vocabulary),
2. superpolynomial increases in difficulty on those cases, and
3. small variations in resources used.

The reason is that if there is a wide range in the size of the successful cases *and* a superpolynomial increase in difficulty as a function of that size, the resources used will have to differ by a factor given by a superpolynomial function of a large difference, which contradicts the supposition that there is only small variation in resources used. For instance:

1. suppose that the size of human vocabulary generally varies by a factor of up to ten (from ten thousand to a hundred thousand),
2. and that learning each additional word takes twice more time and takes constant time for all human beings; then
3. those with a vocabulary of a hundred thousand words will have spent many orders of magnitude more time simply learning the last word they did than those who have a vocabulary of ten thousand words spent learning *their entire* vocabulary.

If we also think that we made the most of the resources available, this would suggest an extraordinary difference in the resources available. Of course, there aren't any plausible theories on which the time required to learn a new word doubles each time, so nothing is ruled out in this case; that is only an illustration.

Speculatively, therefore, this might offer a way to distinguish the mechanisms by which the same task is seemingly implemented in different systems. Suppose, for instance, that large language models display a gradual dropoff in performance, but humans display a sharp dropoff. That might suggest that the LLMs use a polynomial algorithm, but humans one with super-polynomial complexity. This, in turn, could suggest a fundamental architectural difference. If we return e.g. to the debate over language models as models of language, such an asymptotic difference might amount to the sort of difference between a language model and an adequate model of our own linguistic competence or performance. This therefore suggests an experimental means by which to attempt to distinguish us from them. In studying human cognition, we might find that a theory for asymptotic reasons predicts a much smaller range of outcomes or distribution of outcomes against resources used than is actually observed.

References

- AARONSON, Scott, *P \neq NP*, 2017.
- AARONSON, Scott, 'Physical reality': 'Guest Column: NP-complete problems and physical reality', *ACM SIGACT News* 36.1 (1 March 2005), pp 30-52.
- AGARWAL, Shivang, Clarice D. AIELLO, Daniel R. KATTNIG, and Amartya S. BANERJEE, 'Posner molecule': 'The Dynamical Ensemble of the Posner Molecule Is Not Symmetric', *The Journal of Physical Chemistry Letters* 12.42 (28 October 2021), pp 10372-10379.
- ARORA, Sanjeev and Boaz BARAK, *Complexity: Computational Complexity: A Modern Approach*, Cambridge: Cambridge University Press, 2009.
- ARUNACHALAM, Srinivasan and Ronald de WOLF, *A Survey of Quantum Learning Theory*, 28 July 2017, arXiv: 1701.06806 [quant-ph], pre-published.
- CHATER, Nick and Mike OAKSFORD, 'Rational analysis': 'The Rational Analysis Of Mind And Behavior', *Synthese* 122.1 (1 February 2000), pp 93-131.
- CHEN, Jianer, Iyad A. KANJ, and Ge XIA, 'Improved upper bounds for vertex cover', *Theoretical Computer Science* 411.40-42 (September 2010), pp 3736-3756.
- CHERNIAK, Christopher, *Minimal rationality*, Computational models of cognition and perception, Cambridge, Massachusetts; London: MIT Press, 1986, 186.
- CHOMSKY, Noam, *Aspects: Aspects of the Theory of Syntax*, 50th edition, The MIT Press, 1965, JSTOR: j.ctt17kk81z.
- CHOMSKY, Noam, *Knowledge of language: Knowledge of language: its nature, origin, and use*, New York: Praeger, 1986, 362.
- CHOMSKY, Noam and Howard LASNIK, 'Filters and Control', *Linguistic Inquiry* 8.3 (1977), pp 425-504, JSTOR: 4177996.
- CHVÁTAL, Vašek and Endre SZEMERÉDI, 'Many hard examples for resolution', *Journal of the ACM (JACM)* 35.4 (1 October 1988), pp 759-768.
- CLARK, Alexander and Shalom LAPPIN, *Nativism: Linguistic nativism and the poverty of the stimulus*, Chichester, West Sussex Malden, MA: Wiley-Blackwell, 2011.

References

- COBHAM, Alan**, 'Difficulty': 'The Intrinsic Computational Difficulty of Functions', *Logic, methodology and philosophy of science*, edited by BAR-HILLEL, Yehoshua, North-Holland Pub. Co., 1965, pp 24-30.
- CONITZER, Vincent and Tuomas SANDHOLM**, 'New complexity results about Nash equilibria', *Games and Economic Behavior* 63.2 (July 2008), pp 621-641.
- COOK, Stephen A.**, 'Complexity': 'The complexity of theorem-proving procedures', *Proceedings of the third annual ACM symposium on Theory of computing*, STOC '71, New York, NY, USA: Association for Computing Machinery, 3 May 1971, pp 151-158.
- COOK, Stephen A. and Robert A. RECKHOW**, 'Time bounded random access machines', *Journal of Computer and System Sciences* 7.4 (1 August 1973), pp 354-375.
- CORMEN, Thomas H., Charles Eric LEISERSON, and Ronald L. RIVEST**, *Introduction: Introduction to algorithms*, The MIT electrical engineering and computer science series, Cambridge, Mass. : New York: MIT Press ; McGraw-Hill, 1990, 1028.
- COUPÉ, Christophe, Yoon Mi OH, Dan DEDIU, and François PELLEGRINO**, 'Different languages, similar encoding efficiency': 'Different languages, similar encoding efficiency: Comparable information rates across the human communicative niche', *Science Advances* 5.9 (4 September 2019), p eaaw2594.
- COWIE, Fiona**, 'Innateness and Language', *The Stanford Encyclopedia of Philosophy*, edited by ZALTA, Edward N., Fall 2017, Metaphysics Research Lab, Stanford University, 2017.
- DASKALAKIS, Constantinos, Paul W. GOLDBERG, and Christos H. PAPADIMITRIOU**, 'The complexity of computing a Nash equilibrium', *Communications of the ACM* 52.2 (1 February 2009), pp 89-97.
- DAVIDSON, Donald**, 'Theories of Meaning and Learnable Languages', *Inquiries into Truth and Interpretation*, 1st edition, pp 3-16, Oxford University Press Oxford, 27 September 2001, pp 3-16.
- DEAN, Walter**, 'Complexity': 'Computational Complexity Theory', *The Stanford Encyclopedia of Philosophy*, edited by

References

- ZALTA, Edward N., Fall 2021, Metaphysics Research Lab, Stanford University, 2021.
- DERAKHSHANI, Maaneli, Lajos DIÓSI, Matthias LAUBENSTEIN, Kristian PISCICCHIA, and Catalina CURCEANU, 'Crossroad': 'At the crossroad of the search for spontaneous radiation and the Orch OR consciousness theory',** *Physics of Life Reviews* 42 (1 September 2022), pp 8–14.
- DOWNEY, Rod, 'A Parameterized Complexity Tutorial',** *Language and Automata Theory and Applications*, edited by DEDIU, Adrian-Horia and Carlos MARTÍN-VIDE, Berlin, Heidelberg: Springer, 2012, pp 38–56.
- DUPRE, Gabe, 'Competence/performance distinction': 'How and Why to Draw the Competence/Performance Distinction',** *Oxford Handbook of Philosophy of Linguistics*, edited by NEFDT, Ryan M., Gabe DUPRE, and Kate Hazel STANTON, Oxford University Press, 2026.
- EDMONDS, Jack, 'Paths': 'Paths, Trees, and Flowers',** *Canadian Journal of Mathematics* 17 (January 1965), pp 449–467.
- van EMDE BOAS, Peter, 'Models': 'Machine models and simulations',** *Handbook of theoretical computer science (vol. A): algorithms and complexity*, Cambridge, MA, USA: MIT Press, 2 January 1991, pp 1–66.
- FRIEDMAN, Milton, 'Methodology': 'The Methodology of Positive Economics',** *The Philosophy of Economics: An Anthology*, edited by HAUSMAN, Daniel M., 3rd edition, Cambridge: Cambridge University Press, 2007, pp 145–178.
- FRIXIONE, Marcello, 'Tractable Competence',** *Minds and Machines* 11.3 (1 August 2001), pp 379–397.
- GANESH, Vijay and Moshe Y. VARDI, 'On the Unreasonable Effectiveness of SAT Solvers',** *Beyond the Worst-Case Analysis of Algorithms*, edited by ROUGHGARDEN, Tim, Cambridge: Cambridge University Press, 2021, pp 547–566.
- GILBOA, Itzhak and Eitan ZEMEL, 'Nash and correlated equilibria': 'Nash and correlated equilibria: Some complexity considerations',** *Games and Economic Behavior* 1.1 (March 1989), pp 80–93.
- HANSEN, Thomas Dueholm, Haim KAPLAN, Or ZAMIR, and Uri ZWICK, 'SAT': 'Faster k-SAT algorithms using bi-**

References

- ased-PPSZ', *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, New York, NY, USA: Association for Computing Machinery, 23 June 2019, pp 578–589.
- HECK, Richard Kimberly**, 'Does Davidson Argue for Compositionality?', *Journal for the History of Analytic Philosophy* (no date).
- HELFGOTT, Harald Andrés**, *Isomorphismes de graphes en temps quasi-polynomial (d'après Babai et Luks, Weisfeiler-Leman...)*, 12 October 2017, arXiv: 1701.04372 [math], pre-published.
- HENNIE, F. C. and R. E. STEARNS**, 'Two-Tape Simulation of Multitape Turing Machines', *Journal of the Association for Computing Machinery* 13.4 (1 October 1966), pp 533–546.
- HUANG, Jianyu, Tyler M. SMITH, Greg M. HENRY, and Robert A. VAN DE GEIJN**, 'Algorithm': 'Strassen's Algorithm Reloaded', *SC '16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '16: the International Conference for High Performance Computing, Networking, Storage and Analysis, November 2016, pp 690–701.
- KARP, Richard M.**, 'Reducibility among Combinatorial Problems', *Complexity of Computer Computations*, (IBM Thomas J. Watson Research Centre, Yorktown Heights, New York), edited by MILLER, Raymond E., James W. THATCHER, and Jean D. BOHLINGER, Boston, MA: Springer US, no date, pp 85–103.
- KATAJAINEN, Jyrki and Jesper Larsson TRÄFF**, 'Analysis of mergesort': 'A meticulous analysis of mergesort programs', *Algorithms and Complexity*, edited by BONGIOVANNI, Giancarlo, Daniel Pierre BOVET, and Giuseppe DI BATTISTA, Berlin, Heidelberg: Springer, 1997, pp 217–228.
- KEARNS, Michael and Leslie VALIANT**, 'Limitations': 'Cryptographic limitations on learning Boolean formulae and finite automata', *J. ACM* 41.1 (2 January 1994), pp 67–95.
- KODNER, Jordan, Sarah PAYNE, and Jeffrey HEINZ**, *Linguistics will thrive: Why Linguistics Will Thrive in the 21st Cen-*

References

- ture: A Reply to Piantadosi (2023)*, 6 August 2023, arXiv: 2308.03228 [cs.CL], pre-published.
- KORTE, Bernhard and Jens VYGEN**, *Combinatorial Optimization: Combinatorial Optimization: Theory and Algorithms*, volume 21, Algorithms and Combinatorics, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.
- LEVIN, L.A.**, ‘Универсальные задачи перебора’, *Проблемы передачи информации* IX.3 (1973).
- LIPTON, Richard J. and Kenneth W. REGAN**, ‘David Johnson’: ‘David Johnson: Galactic Algorithms’, *People, Problems, and Proofs: Essays from Gödel’s Lost Letter: 2010*, edited by LIPTON, Richard J. and Kenneth W. REGAN, Berlin, Heidelberg: Springer, 2013, pp 109–112.
- LITT, Abninder, Chris ELIASMITH, Frederick W. KROON, Steven WEINSTEIN, and Paul THAGARD**, ‘Is the Brain a Quantum Computer?’, *Cognitive Science* 30.3 (2006), pp 593–603.
- MALIK, Sharad and Lintao ZHANG**, ‘Boolean satisfiability from theoretical hardness to practical success’, *Communications of the ACM* 52.8 (1 August 2009), pp 76–82.
- MALPAS, Jeff**, ‘Donald Davidson’, *The Stanford Encyclopedia of Philosophy*, edited by ZALTA, Edward N. and Uri NODELMAN, Fall 2024, Metaphysics Research Lab, Stanford University, 2024.
- MARQUES-SILVA, Joao**, ‘Applications’: ‘Practical applications of Boolean Satisfiability’, *9th International Workshop on Discrete Event Systems*, Gothenberg: IEEE, 2008, pp 74–80.
- MARR, David**, *Vision: Vision: a computational investigation into the human representation and processing of visual information*, Cambridge, Mass: MIT Press, 2010.
- MCCALL, Edward H.**, ‘Simplex performance’: ‘Performance results of the simplex algorithm for a set of real-world linear programming models’, *Communications of the ACM* 25.3 (1 March 1982), pp 207–212.
- MCKEMMISH, Laura K., Jeffrey R. REIMERS, Ross H. MCKENZIE, Alan E. MARK, and Noel S. HUSH**, ‘Penrose–Hameroff proposal’: ‘Penrose-Hameroff orchestrated objective-reduction proposal for human consciousness is not biologically feasible’, *Physical Review. E, Statisti-*

References

- cal, Nonlinear, and Soft Matter Physics* 80 (2 Pt 1 August 2009), p 021912, PMID: 19792156.
- MILLIÈRE, Raphaël and Charles RATHKOPF**, ‘**Anthropocentric bias in language model evaluation**’, *Computational Linguistics* 52.1 (1 March 2026), pp 379–388, arXiv: 2407.03859 [cs.CL].
- MOORE, Gordon E**, ‘**Cramming**’: ‘Cramming more components onto integrated circuits’, *Electronics* 38.8 (1965).
- NOWAK, Martin A., Natalia L. KOMAROVA, and Partha NIYOGI**, ‘**Aspects**’: ‘Computational and evolutionary aspects of language’, *Nature* 417.6889 (June 2002), pp 611–617.
- PAGIN, Peter**, ‘**Compositionality, computability, and complexity**’, *The Review of Symbolic Logic* 14.3 (September 2021), pp 551–591.
- PAGIN, Peter**, ‘**Radical interpretation and compositional structure**’, *Donald Davidson: truth, meaning and knowledge*, edited by ŽEGLIĆ, Urszula M. and Donald DAVIDSON, Reprinted, Routledge studies in twentieth century philosophy 2, London: Routledge, 2001.
- PAGIN, Peter and Dag WESTERSTÅHL**, ‘**Compositionality II**’: ‘Compositionality II: Arguments and Problems’, *Philosophy Compass* 5.3 (2010), pp 265–282.
- PAPADIMITRIOU, Christos H.**, ‘**The Complexity of Finding Nash Equilibria**’, *Algorithmic Game Theory*, edited by TARDOS, Eva, Noam NISAN, Tim ROUGHGARDEN, and Vijay V. VAZIRANI, Cambridge: Cambridge University Press, 2007, pp 29–52.
- PAPADIMITRIOU, Christos H. and Mihalis YANNAKAKIS**, ‘**Complexity as bounded rationality**’: ‘On complexity as bounded rationality (extended abstract)’, *Proceedings of the twenty-sixth annual ACM symposium on Theory of Computing*, STOC ‘94, New York, NY, USA: Association for Computing Machinery, 23 May 1994, pp 726–733.
- PIANTADOSI, S.T.**, ‘**Language models**’: ‘Modern language models refute Chomsky’s approach to language’, *From field-work to linguistic theory: A tribute to Dan Everett*, edited by GIBSON, Edward and Moshi POLIAK, Empirically Oriented Theoretical Morphology and Syntax 15, Language Science Press, 5 July 2024, pp 353–414.

References

- PICCININI, Gualtiero**, ‘**Computation in Physical Systems**’, *The Stanford Encyclopedia of Philosophy*, edited by ZALTA, Edward N. and Uri NODELMAN, Fall 2025, Metaphysics Research Lab, Stanford University, 2025.
- PICKEL, Bryan and Zoltán Gendler SZABÓ**, ‘**Compositionality**’, *The Stanford Encyclopedia of Philosophy*, edited by ZALTA, Edward N. and Uri NODELMAN, Winter 2025, Metaphysics Research Lab, Stanford University, 2025.
- READ, Dwight W., Héctor M. MANRIQUE, and Michael J. WALKER**, ‘**Working memory**’: ‘On the working memory of humans and great apes: Strikingly similar or remarkably different?’, *Neuroscience and Biobehavioral Reviews* 134 (March 2022), p 104496, PMID: 34919985.
- REIMERS, Jeffrey R., Laura K. MCKEMMISH, Ross H. MCKENZIE, Alan E. MARK, and Noel S. HUSH**, ‘**Fröhlich condensation**’: ‘Weak, strong, and coherent regimes of Fröhlich condensation and their applications to terahertz medicine and quantum consciousness’, *Proceedings of the National Academy of Sciences* 106.11 (17 March 2009), pp 4219–4224.
- RISTAD, Eric Sven**, *Game: The language complexity game*, Artificial intelligence series, Cambridge (MA), London: MIT Press, 1993.
- van ROOIJ, Iris**, ‘**Tractable**’: ‘The Tractable Cognition Thesis’, *Cognitive Science* 32.6 (2008), pp 939–984.
- van ROOIJ, Iris, Mark BLOKPOEL, Johan KWISTHOUT, and Todd WAREHAM**, *Cognition: Cognition and Intractability: A Guide to Classical and Parameterized Complexity Analysis*, Cambridge: Cambridge University Press, 2019.
- van ROOIJ, Iris, Olivia GUEST, Federico ADOLFI, Ronald de HAAN, Antonina KOLOKOLOVA, and Patricia RICH**, ‘**Reclaiming AI**’: ‘Reclaiming AI as a Theoretical Tool for Cognitive Science’, *Computational Brain & Behavior* 7.4 (1 December 2024), pp 616–636.
- van ROOIJ, Iris, Cory D. WRIGHT, and Todd WAREHAM**, ‘**Intractability and the use of heuristics in psychological explanations**’, *Synthese* 187.2 (1 July 2012), pp 471–487.
- SEARLE, John R.**, *Rediscovery: The Rediscovery of the Mind*, MIT Press, 8 July 1992, 292.

References

- SHANKS, Daniel**, *Problems in number theory: Solved and unsolved problems in number theory*, 2nd edition, New York, NY: Chelsea Publ. Co, 1978, 258.
- SPIELMAN, Daniel and Shang-Hua TENG**, ‘Smoothed analysis of algorithms’: ‘Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time’, *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, STOC ‘01, New York, NY, USA: Association for Computing Machinery, 6 July 2001, pp 296–305.
- STRASSEN, Volker**, ‘Elimination’: ‘Gaussian elimination is not optimal’, *Numerische Mathematik* 13.4 (1 August 1969), pp 354–356.
- TEGMARK, Max**, ‘Importance of quantum decoherence in brain processes’, *Physical Review E* 61.4 (1 April 2000), pp 4194–4206.
- TURING, A.M.**, ‘Computability’: ‘Computability and λ -definability’, *Journal of Symbolic Logic* 2.4 (1937), pp 153–163, JSTOR: 2268280.
- VARDI, Moshe Y.**, ‘Boolean satisfiability’: ‘Boolean satisfiability: theory and engineering’, *Communications of the ACM* 57.3 (1 March 2014), p 5.
- WEISBERG, Michael**, ‘Idealization’: ‘Three Kinds of Idealization’, *The Journal of Philosophy* 104.12 (2007), pp 639–659, JSTOR: 20620065.
- WIEST, Michael C**, ‘Quantum microtubule substrate’: ‘A quantum microtubule substrate of consciousness is experimentally supported and solves the binding and epiphenomenalism problems’, *Neuroscience of Consciousness* 2025.1 (6 May 2025), p niafo11, PMID: 40342554.
- WINOGRAD, Terry**, ‘What Does it Mean to Understand Language?’, *Cognitive Science* 4.3 (1980), pp 209–241.
- de WOLF, R.M.**, ‘Applications’: ‘Philosophical Applications of Computational Learning Theory: Chomskyan Innateness and Occam’s Razor’, master’s thesis, Rotterdam: Erasmus Universiteit Rotterdam, 1997.
- YANG, Charles**, ‘The Great Number Crunch’, *Journal of Linguistics* 44.1 (2008), edited by BOD, Rens, Jennifer HAY, and Stefanie JANNEDY, pp 205–228, JSTOR: 40058032.

References

- YUNGER HALPERN, Nicole and Elizabeth CROSSON, 'Posner model': 'Quantum information in the Posner model of quantum cognition', *Annals of Physics* 407 (1 August 2019), pp 92-147.**
- ZHENG, Jieyu and Markus MEISTER, 'The unbearable slowness of being': 'The unbearable slowness of being: Why do we live at 10 bits/s?', *Neuron* 113.2 (22 January 2025), pp 192-204.**