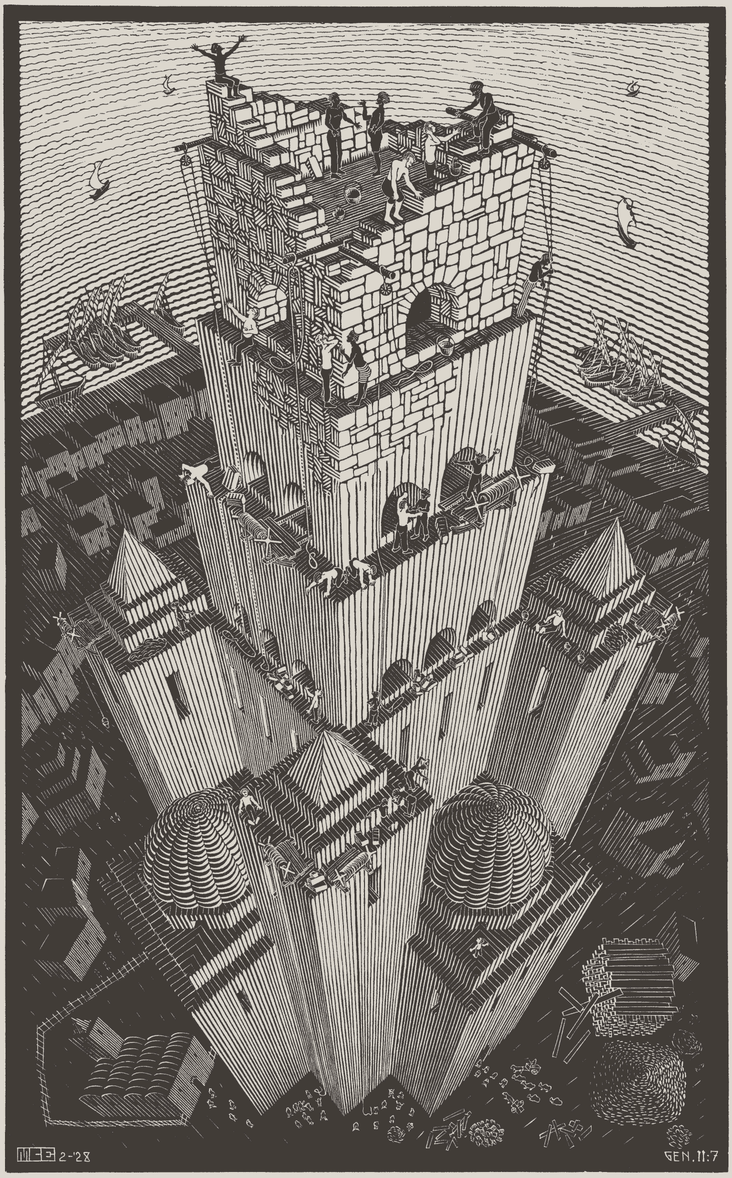
The asymptotics of language.

a thesis submitted in partial fulfilment of the requirements of the degree of Bachelor of Philosophy in Philosophy.

J.P. Loo.



The asymptotics of language.

J.P. Loo.

Nadie puede articular una sílaba que no esté llena de ternuras y de temores; que no sea en alguno de esos lenguajes el nombre poderoso de un dios.

Borges, 'La Biblioteca de Babel': 98–9.

system: Typst · typeface: Lucida by Bigelow and Homes.
on the cover: *Tower of Babel*, M.C. Escher (1928).

Contents

Abstract.

What humans and machines do is subject to resource constraints: energy, memory, computational time and so on. Asymptotic analysis is a technique from theoretical computer science that measures how resource usage grows as a function of the size of an instance of a problem. Philosophers and cognitive scientists interested in formulating theories that respect resource constraints have therefore looked to asymptotic analysis to determine which theories do and don't respect resource constraints, which I call 'tractability constraints'.

The central argument of this thesis is that almost all arguments appealing to asymptotic analysis in the study of cognition and human language make a fundamental structural mistake in considering the so-called worst case of problems. They should, I claim, instead impose a tractability constraint on the worst case that *might easily counterfactually have been observed*. This leads to a pessimistic diagnosis of the existing literature: it imposes a tractability constraint on the wrong cases.

The concrete examples this thesis examines in detail both concern language; one is about language processing and semantics, and the other about language acquisition and the Chomskyan programme. However, some broader implications are sketched, first for the resource-constrained study of cognition generally, and, second, for the Chomskyan programme's broader implications for the relationship between language models and human language.

I Preliminaries.

§ 1 I first motivate a critical examination of asymptotic analysis in philosophy, cognitive science and linguistics, by mentioning some arguments that employ asymptotic analysis.
§ 2 Asymptotic analysis is used, very roughly, to offer a working definition of 'too hard'. In three dialogues, I give an informal exposition of why I think much existing work appealing to asymptotic analysis has gone partly wrong.
§ 3 I offer an outline of the remainder of the thesis.

1 The use and abuse of asymptotic arguments.

Why should we care about asymptotic analysis in philosophy? The simple answer is that either we can *use* asymptotic analysis to deliver some quite interesting conclusions in the philosophy of mind, language, linguistics and cognitive science, or others have *abused* asymptotic analysis to deliver misleading arguments to that effect. I give three examples here. We will later discuss the merits of the language-related arguments in detail. Although I think my arguments would also largely apply, *mutatis mutandis*, to the other cases, I leave them beyond the scope of this thesis. In this section I simply propose to state their conclusions, to preliminarily motivate the questions examined in this thesis.

1.1 Our knowledge and acquisition of language.

Various asymptotic constraints on theories of language acquisition have been offered by philosophers, linguists and cognitive/computer scientists. To a first approximation, they seek to ensure that theories respect the fact that we have limited computational resources at our disposal in language acquisition, and, therefore, propose to rule out theories according to which we would use *more* resources than available. Many authors employ asymptotic criteria either as a *prima facie* argument in favour of their preferred theories or a more definitive argument against theories of language acquisition.[1]

1 CLARK and LAPPIN, *Nativism:* cap 8; ABE, 'Feasible learnability of formal grammars and the theory of natural language acquisition'; NOWAK et al., 'Evolution'; NOWAK et al., 'Aspects': 613; YANG, 'The Great Number Crunch': 223; KODNER et al., *Linguistics will thrive:* 5; de WOLF, 'Applica-

Some of these authors suggest that their theories respect resource constraints, and that is a *prima facie* (defeasible) consideration in their favour. Principally, these results are of interest in linguistics, and, depending on one's view of the syntax–semantics interface, perhaps semantics and philosophy of language. Others also use resource constraints as an argument against theoretical rivals.

In this brief survey, I will simply expound a little of the most prominent example, namely, the Chomskyan programme. Humans, Chomsky has argued, have innate knowledge of language in the form of *universal grammar*, which we can take to a first approximation to be 'an intensional definition of a possible human language' in the form of a grammar (ROBERTS, 'Introduction': 2). Universal grammar might be very simple indeed. According to the minimalist programme, a grammar strictly only requires a single operation, 'Merge', which combines two expressions to form a larger one, together with a stock of basic expressions to be so combined (DUPRE, 'Innateness and Language': § 3.3). On the other hand, the so-called 'principles and parameters' framework is more heavyweight (*ibid.:* § 3.2). A principle is a 'universal constraint on what kind of languages are possible'. A parameter is a locus of permitted variation within a small range of values. For instance, in some languages, verbs come before their objects (thus 'I eat fish'), but in other languages, verbs come after their objects (thus 'I fish eat'). A combination of set parameters then determines a language. The space of possible parameters 'define[s] the range of possible languages'.

The fate of the Chomskyan programme is of interest to philosophers for a number of reasons. One is that it might offer succour to rationalists in their dispute with empiricists.[2] Another is that the Chomskyan programme is, in effect, a programme for the study of language in general. It might, for instance, largely rule out the use of statistical methods in linguistics and the study of language (see NORVIG, 'On Chomsky and the Two Cultures of Statistical Learning' for a reconstruction of and response to such views). Taking such views further,

tions': cap. 2; HEINZ, 'Learning Long-Distance Phonotactics'; BERTOLO, 'A brief overview of learnability'; CLARK and LAPPIN, 'Acquisition'.

2 SCHOLZ et al. ('Philosophy of Linguistics': § 4.2.3) survey and contest such arguments.

we might hold that statistical methods for language (such as large language models) have 'achieved zero', as Chomsky has suggested in an interview (Machine Learning Street Talk, *The Ghost in the Machine—Noam Chomsky*), both in terms of their scientific fruitfulness as models of human language, and in terms of their own capabilities for the purpose e.g. of automating certain forms of reasoning and cognitive labour.

1.2 *Machine learning models as decoys.*

A rough description of modern machine learning techniques is this. A statistical learner is given many data relevant to the problem at hand (vision, language, and so on). It learns, for instance, to recognise objects, by observing statistical patterns in other examples where we have labelled objects. Moreover, once we have succeeded in inducing the system to learn the underlying capacity, we may have a model of that capacity useful for the purpose of cognitive science.

It is also 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 (and, perhaps, whether it even counts as knowledge in the first place). 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' that 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

3 *pace* Millière and Rathkopf ('Anthropocentric bias in language model evaluation').

come to know what we do, even in an idealised learning environment. Learning is *too hard*, even in a régime idealised to the benefit of the learner (*ibid.:* Theorem 2). They conclude that any AI system that purports to have human-level capabilities is, therefore, at best a 'decoy'. 'Too hard' is explicated in terms of asymptotic analysis. Whether or not their reading of 'too hard' is correct is therefore of interest.

1.3 *Rationality.*

Homo economicus, notoriously, maximises expected utility. In some environments, that is very hard, and uses far more computational resources than would be reasonable. This ought to worry anybody hoping to found explanations of human behaviour on the predicted behaviour of *homo economicus*.[4] The same applies to many other models of human behaviour in philosophy, the social sciences, and so on.

1.4 *The actual scope of the thesis.*

I do not propose to explore all these examples. I shall advance a general conclusion about the use and abuse of asymptotic analysis in cognitive science, philosophy and so on generally. But the examples I examine will be limited specifically to the case of language: thus, the asymptotics of language. I happen to think that my conclusions straightforwardly apply to these other fields as well, but I leave that beyond the scope of this thesis.

2 What counts as 'too hard' in three dialogues.

The central argument of this thesis concerns how philosophers should think about when tasks are 'too hard'. I then apply that answer to some of the arguments above. (Roughly, my conclusion is that they are far less decisive than traditionally thought.) In this section, I informally expound my views about that question, in three dialogues. In the first dialogue, I give a simple example of how considerations about what is 'too hard'

4 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.

might inform our views generally. In the next two dialogues, I explain two more specific views about how we should make such arguments.

2.1 *A first dialogue: an Austin Seven.*

Consider the following style of argument. Alice and Bob are in the other place in 1958. They see the following scene.

ALICE Oh look, an Austin Seven! What an amusing prank!

BOB That can't be a car. Nobody could possibly have carried a car all the way up to the roof—certainly not without their being noticed.

ALICE It looks very convincing to me.

BOB Well, how exactly do you think they got whatever it is on the roof up there? A real Austin Seven is very heavy and surely some rather elaborate apparatus would have been required. And they would have been stopped.

Bob here seeks to constrain a theory of *what* is on the roof by *how* it could be lifted onto the roof. And he has posed a reasonable question: how exactly could a *bona fide* Austin Seven have been lifted onto the roof? Moreover, if he is right that there is no good answer to that question, it is reasonable for him to infer that it must be something else (e.g. a very sophisticated model).

On the other hand, it so happens that he was wrong. (If memory serves, the tabs took the engine out.) The argument that lifting an Austin Seven onto the roof of Senate House was indeed reasonable, and ought to have worried Alice, but it was *defeasible* evidence against the theory that a real Austin Seven was on the roof.

This thesis is concerned with arguments in this spirit, though they are much more complicated. Two psychologists will help to make some of the niceties to be explored in this thesis clearer.

2.2 *A second dialogue: two psychologists.*

A little formalism is required to make the following dialogue intelligible. Consider the following propositions.

P Roses are red.

Q Violets are blue.

R Sugar is sweet.

Now, consider the following claims.

- P is true and Q is true.
- P is true or R is true.

These two claims are *satisfiable*. There is a way of assigning truth values to P, Q and R such that both claims, taken together, come out true.

Consider now another claim.

- P is true, and if P is true then Q is true, and if P is true then Q is not true.

This claim is *unsatisfiable*: there is no way of assigning truth values to P and Q such that the overall sentence comes out true. The problem is that we have to set P to be true, and then we have to set Q to be true and not to be true.[5]

We can introduce symbols for 'connectives' like 'and', 'or', 'if…then…' and 'not', namely $\wedge$, $\vee$, $\rightarrow$ and $\neg$. We can then construct formulæ such as

- $P \wedge Q$,
- $P \vee R$ and
- $P \wedge (P \rightarrow Q) \wedge (P \rightarrow \neg Q)$.

These are called *boolean formulæ*. The problem of satisfiability is to, given a boolean formula, decide whether it is satisfiable.

Computer scientists generally think that the problem of satisfiability is quite hard. (I delay a precise explanation of that

5 I am assuming here, of course, the law of non-contradiction, *pace* Priest ('The Logic of Paradox').

to § 10.5.) Let us take that as given. Suppose now two psychologists are discussing human reasoning.

CAROL I wonder how it is that humans assess the plausibility of claims other people make. Here is my theory. [We call it T.] On hearing a series of claims, humans formalise the claim as boolean formulæ. Then they test them for satisfiability.

DAVID There is a problem. It has been convincingly argued that the time it takes to determine whether a boolean formula is satisfiable increases exponentially as a function of its length.[6] That is to say: there is some constant k, such that every time we add a new symbol to the boolean formula, the time to assess its satisfiability is multipled by k. That suggests that satisfiability is too hard a problem for humans to solve. So T is implausible.

CAROL But that is an absurd constraint. Even exponential functions often have small starting values. For example, $2^3 = 8$. So, for a start, to rule out T, you need to show that the input sizes really are big enough to reach a tractability constraint. Moreover, there are many different exponential functions. Some are much more forgiving than others. There are some algorithms that take, for instance, about 1.1^n time. That is, every time we increase the size of the input by 1, the time taken to solve the problem increases by a factor of 1.1. This is much more forgiving than e.g. 2^n. So you need to account for the *base* of the exponent. Finally, exponential functions also can contain additive and multiplicative constants. Something like $1.1^n + 10^{100}$ is much larger than 1.5^n for most reasonable values of n. On the other hand, $\frac{1.3^n}{10^{100}}$ is much smaller than 1.1^n for quite a long time. So even once you account for the input size, you also need to account for these constants, before you rule out my theory.

DAVID The very purpose of asymptotic analysis is to disregard these constraints. Our knowledge of the precise computational minutiæ of the human brain is limited; unlike the case of an electronic computer, we didn't design it. The problem with fo-

6 This is not actually true, but for the purpose of simplifying the exposition, we shall ignore that nicety. Satisfiability for formulæ in 3-CNF are thought to have this property. There are efficient algorithms for so-called 'disjunctive normal form' as a function of their length. The problem is that DNF is very inefficient; that is, we can generally write a logically equivalent formula in CNF that is much shorter.

cusing on the difference between constants is that results about them will vary from one computational device to another. The utility of results about, for instance, exponential hardness is that they do not depend on the computational minutiæ of the human brain. They apply to all plausible computational systems.[7]

CAROL I should distinguish two points. First, there is a point about *runtimes*. That is: is a process that takes e.g. 1.1^n seconds to be treated as intractable in the same way as one that takes 2^n seconds? The considerations you adduce are not, I think, responsive to this point. Second, there is a point about the robustness of results over different computational systems. I agree that it would in principle be desirable to be able to theorise about the tractability of problems with no regard for the minutiæ of the computational systems that solve them. But that desirability does not mean that it turns out we can tenably theorise that way. That is something that must be demonstrated *a posteriori*.

DAVID Your point is, in principle, well taken. But, in practice, on actually existing comptuational systems, there aren't that many small bases—most are at least e.g. 1.5 or 2. Moreover, there are in practice hardly ever multiplicative constants that significantly decrease the time required (e.g. $\frac{1.1^n}{10^{100}}$). Therefore, in practice, unless the inputs are unusually and uninterestingly small, exponential increases in difficulty are enough to rule out a theory.

CAROL The standard you propose will often be correct. But that does not mean it has the dialectical effect desired. In particular, it is simply not true that the inputs have to be *uninterestingly* small. As you admit, there are bases as small as 1.1 or 1.3. By way of illustration, suppose that the human brain has only 1000 microseconds in which to discharge a task, and that the task takes 1.1^n microseconds on an input of size n. That leaves room for inputs of up to size $n = 72$. Working memory is (to a first approximation) bounded by about 7 ± 2 items (READ et al., 'Working memory'), but each such item or 'chunk' may correspond to several symbols of the formalised claim, so n may plausibly fall in the tens—inside the feasible region for a base of 1.1, but far outside it for a base of 2. So there are in-

7 See § 12.1 and § 11.

teresting and plausible theories that can posit that we take exponential time to perform certain tasks on relevantly large inputs. The details, therefore, matter: multiplicative and additive constants, and the base.

My argument is, in effect, that of Carol.

2.3 *A third dialogue: the burden of proof.*

A final question concerns the burden of proof. Suppose that the conversation reaches the following point.

- David has shown that, for instance, the exponential base really is 2 according to Carol's theory. Moreover, he has shown that in practice n becomes quite large (e.g. 100).
- Carol has, on the other hand, significant independent evidence in favour of her theory: perhaps, for instance, it fits observed patterns of human reasoning—response times, characteristic errors—better than any rival explanation does.

We seem to be at an impasse. On the one hand, there is good reason to subscribe to Carol's theory. On the other hand, David seems to show that it attributes to us something implausibly hard. Carol might respond thus.

CAROL It seems to me that your hardness results concern the 'worst case' among instances of any given size. An algorithm to solve satisfiability might in fact only increase the time required exponentially on a very small fraction of inputs. Maybe in practice these are the ones that arise in cases relevant to my theory (human reasoning). Your tractability constraint needs to apply to the cases of human reasoning that actually arise, not to hypothetically possible boolean formulæ.

DAVID Your theory attributes to its subject the solution of the satisfiability problem in general. You are therefore committed to the solution of satisfiability *in general*. If you think that only a restricted set of formulæ arise in human reasoning, you should specify a new theory. That new theory should specify which boolean formulæ arise. If the new theory with a restricted set of inputs also displays exponential difficulty, then it can also be ruled out. But in any case that would only be a defence of a *revised* theory rather than the original.

CAROL It is true that I am committed to the *existence* of a restricted domain of boolean formulæ relevant to human reasoning. However, it is enough that such a domain *exists* to suggest that a revised version of my theory is true and does not contradict resource limitations. That much I concede. The question, however, is whether I am also obliged to *construct* or explicitly state what that restricted domain is. I do not think that there are any *a priori* grounds to make such a demand generally. On the contrary, the question of whether it is reasonable to accept that a domain exists with certain properties, even though we cannot explicitly construct it, should be judged by standard naturalistic or scientific criteria. The independent evidence I have adduced in favour of a satisfiability account of reasoning should then be taken, further, as evidence of the existence of this domain.

Here, again, I side with Carol. It is reasonable to suggest that there *exists* a restricted domain without *constructing* it. Moreover, I shall argue that this is a naturalistically acceptable practice: it is the situation in computer science.

3 Analytic table of contents.

- Chapter II: What rôle should resource constraints and considerations of feasibility and tractability play in philosophy and allied disciplines? I expound my preferred view here. (This corresponds to most of § 2.2.)
- Chapter III: Are measures of tractability derived from asymptotic analysis suitable for those purposes? Not, I claim, as ordinarily used. (This corresponds to most of § 2.3).
- Chapter IV: My general pessimism about asymptotic analysis applies to an argument given from tractability for the compositionality of meaning.
- Chapter V: Ditto for nativist arguments for universal grammar from asymptotic analysis (and, perhaps surprisingly, some empiricst arguments seeking to respond to them.)
- Chapter VI: This is all, perhaps, a little pessimistic. Can asymptotic analysis play any useful rôle, in philosophy generally or the study of language? Yes, I conjecture, on a concluding speculative note. Asymptotic analysis can do more than help us to characterise when theories are tractable. We can also use asympototic analysis to work out tradeoffs between resource usage and outcomes. These are potentially more empirically testable and fruitful than tractability-based arguments.

II Tractability in philosophy.

The broad purpose of this thesis is to examine whether and how asymptotic analysis should inform the study of language. The purpose of asymptotic analysis is to measure the difficulty of problems and instances of problems. Based on this measure of difficulty, we can classify problems or instances thereof as tractable or intractable. The ultimate claim of this thesis is that asymptotic analysis has been misapplied in drawing the boundary between tractability and intractability. In particular, I shall distinguish

- tractability as it is studied in asymptotic analysis in computer science, and
- tractability as we might define it to make a dialectical contribution in cognitive science and philosophy,

and claim that there is a mismatch between the two that requires more careful handling than has been hitherto standard. But before I characterise the mismatch, I need to offer a view on the second point: what exactly is the place of tractability in philosophy and the study of language and cognition more generally?

§ 4 I offer a few intuitive examples of how tractability might inform philosophy, with a few comparisons with possibility-based arguments.

§ 5 I characterise various ways of making tractability constraints more precise.

§ 6 I expound the 'tractable cognition thesis', the most well-known application of tractability constraints to philosophy and cognitive science.

§ 7 I expound (and rebut) two arguments against a place for *any* tractability constraint whatsoever.

§ 8 I accept that some forms of 'competence' are not subject to a tractability constraint, but others are.

4 Tractability and possibility.

'Tractability' is a technical term, approximately synonymous with 'practicable' or 'feasible'. Its usage is largely restricted to computer and cognitive science, and mathematics. (Thus engineers do not usually call problems in civil engineering 'tractable' so much as e.g. 'feasible'.) At any rate, I shall use the terms synonymously.

All three terms are somewhat imprecise. We might worry that there is a risk of equivocation in making arguments using these concepts. So why theorise in terms of tractability at all?

The place of tractability in philosophical argument is analogous to that of the various species of possibility. Philosophers have argued, for instance, that

- p-zombies are possible, so physicalism in the philosophy of mind is false (CHALMERS, 'The two-dimensional argument against materialism'),
- the completion of Hilbert's programme, at least in naïve form, is impeded by Gödel's incompleteness theorems (GÖDEL, 'Formally undecidable propositions') and the concomitant impossibility of carrying out much arithmetic in any self-verifying logical system (ZACH, 'Hilbert's Program': §§ 1.4, 4), and
- the nature of a moral dilemma is that it is impossible to avoid either wrongful action or wrongful inaction (MCCONNELL, 'Moral Dilemmas': § 2).

Nomological and metaphysical possibility are rather liberal. Tractability is less so. An intuitive example: it is logically (and probably nomologically) possible for me to walk to Vladivostok, but it is not feasible for me to do so. On the other hand, it is both feasible and possible for me to walk to the Cherwell. (We might think of tractability as possibility relative to resource constraints.) Tractability, therefore, might allow us to make different arguments. A simple example: LEVESQUE ('Complexity': § 6) asks whether classical logic is a good model of human reasoning. He answers in the negative: reasoning in classical logic is intractable. Moreover, in modifying classical logic to make it tractable, we shall also find a logical formalism that is more psychologically realistic.

A few more quotidian examples illustrate both the commonsensical and scientific respectability of reasoning in terms of tractability. Consider the following argument.

4.1. • It's not feasible for a human to walk more than ten kilometres on an empty stomach.

- Socrates has been fasting the whole week, was seen in a field twenty kilometres away from the agora yesterday, but is now in the agora.
- Socrates is a human.
- Therefore, Socrates did not walk the whole way.

4.1 is a perfectly valid and sound argument. This argument has a negative conclusion (that Socrates didn't walk the whole way.) But we might also advance valid arguments with positive conclusions.

4.2.
- …
- Socrates either walked or was transported in a horse-drawn carriage.
- Therefore, Socrates took a horse-drawn carriage.

4.1 and 4.2 are deductive. But we can sometimes offer e.g. abductive arguments.

4.3.
- It is very hot.
- People who walk quickly when it is hot sweat.
- Jowett made it from the Radcliffe Camera to Oxford station in ten minutes.
- Jowett generally dislikes being sweaty.
- Therefore, Jowett arrived at the station by some means other than walking.

What all these arguments have in common is some view about the practical limits to walking, or, equivalently, the feasibility of walking at various speeds.

5 Some variants of tractability.

Tractability, like possibility, admits many precisifications. In this section, I more precisely characterise those precisifications, to clarify others' and my own arguments.

The working understanding of tractability employed in this thesis is in terms of *resource constraints*. Tasks use up resources.

- Someone walking uses energy. They also require the instantaneous generation of force in their muscles to accelerate.

- A system multiplying numbers will need space to store the numbers themselves and to record intermediate workings.
- In solving a statistical problem, resources might include samples from which to make inferences.

Resource constraints then might include calorific intake, memory available, and the maximum size of a sample that can practically be obtained.

We will now make this more precise by stating how resource constraints can arise.

5.1 *Tractability for whom?*

There are at least two places where considerations of tractability arise.

- ***Theorist-side*** **tractability.** Sometimes we use theories to generate predictions, find solutions given boundary conditions, and so on. Thus we use aerodynamics to predict the trajectories of aeroplanes and epidemiology to predict the spread of disease. The generation of these predictions and solutions is amenable to analysis in terms of tractability. For instance, the chaotic behaviour of the atmosphere makes generating predictions far into the future about the weather difficult. This is *theorist*-side difficulty because it is difficulty for the theorist. A theory is *theorist-side tractable* if prediction and solution are tractable for that theory relative to the resources of a theorist.
- ***Subject-side*** **tractability.** By the subject of a theory, I simply mean what the theory concerns: e.g. the weather, animals, stars and so on. Sometimes, the subjects of a theory can't meaningfully be said to act or do anything. Volcanoes don't act; nor does the sun. Theories purely about volcanoes or stars therefore do not have as their subjects entities capable of action. However, sometimes the subjects of a theory can and do act. A theory about elephants or humans concerns beings that communicate, eat and walk around. A theory about agents may posit that those agents act in certain ways. Those actions, in turn, may be easy or hard, and tractable or intractable. A theory is *subject-side tractable* if the actions it attributes to its subjects are tractable relative to the resources available to the subject.

5.2 *Tractability of what?*

We can attribute tractability to individual tasks or actions, such as my walk to the Radcliffe Camera this morning, or my walk to the train station yesterday. There are various features of these scenarios. The weather was reasonable. The roads were open. I am able-bodied. There were no sinkholes on the way.

- ***Parameters.*** Consider now the set of all the factors that might change resource usage in a token of a task (such as walking or sorting a list). We call these the parameters of the problem, $P_1, P_2, \ldots$. Say that these parameters then can take values $v_1 \in V_1, v_2 \in V_2, \ldots$
- ***Parameter spaces.*** For any given problem, there is then a parameter space comprising all the different combinations of values of relevant parameters $V_1 \times V_2 \times \ldots$. Parameter space for walking includes the weather, gradients, the length of a walk and features of the terrain.
- ***Parameter subspaces.*** If we fix a certain value for a given parameter $v_k \in V_k$, we obtain a parameter subspace $V_1 \times V_2 \times \ldots \times \{v_k\} \times \ldots$. For instance, we might say that the weather is cloudless, windless and 20C; then there is a parameter subspace in which e.g. the gradient, distance and terrain vary, but not the weather. Since very many factors are potentially relevant but are not in practice worth considering, scientific theory usually happens in a parameter subspace.
- ***Regions* and *instance classes.*** Parameter spaces can be carved up into regions or classes by specifying constraints on parameters. For instance, there is a region of walks in which the gradient never exceeds 10%.
- ***Instances.*** When we specify *all* the parameters in a space, we have an instance of a problem $\langle v_1, v_2, \ldots, \rangle \in V_1 \times V_2 \times \ldots$. One example is my walk this morning. When we work in a restricted parameter subspace, an instance specifies only the parameters included in the subspace. Consider the subspace given only by the distance of a walk. An instance is then simply specified by a distance.

We may attribute tractability to instances, regions/classes and problems. Thus:

- (an instance) walking ten kilometres on a flat surface is feasible;
- (a class) walking up a 200% gradient is infeasible; and

- (a problem) walking in general is feasible.

The first claim is perfectly well-defined. However, the second and third are not obviously well-defined. Intuitively, it really is infeasible to walk up a 200% gradient. But it might be possible to walk such a gradient on a very short staircase. Should we retreat only to attributions of tractability to instances?

I suggest not. We should seek to evaluate these claims by working out a more precise region/class to which they relate. Suppose I am especially literal-minded, and I say ‘one can always walk in London’. Someone could reasonably point out that the intended significance of this claim is that someone of ordinary fitness could walk from any point in London to any other point. It is then easy to see why one might think this claim false (consider a walk from Croydon to Epping). Therefore, in the context, we establish a more precise claim about a precisely defined region, and then evaluate the tractability of the instances within it.

On this view, instances are the primary bearers of (in)tractability. Derivatively, as a function of context, classes and problems as a whole can be called tractable or intractable.

Several variants of tractability at the level of a problem as a whole are of interest.

- ***Actual tractability***. Consider a theory that makes claims about the energy requirements of walking. One natural way of assessing the theory is to consider cases of walks we actually make. That is, we examine the adequacy of the theory against e.g. my afternoon walk rather than a hypothetical walk from here to Vladivostok. The *actual* instances of a problem are those instances that actually arise within the scope of a theory. Consider a theory about the walking habits of people in Britain: within the scope of the theory are e.g. a don’s morning walk from Park Town to a central Oxford college (one can dream), but not a hypothetical walk I take from here to Vladivostok. Actual tractability is then a matter of the subject-side tractability of the actual instances according to a theory. For instance, if a theory suggests that every next step on a walk has double the energetic requirement of the previous one, and the first step requires a hundred kilocalories, it will classify all but the shortest walks as intractable. It will therefore fail even the task of actual tractability. There are actual walks of e.g. a hundred steps. That would require approximately 100×2^{100} kilocalories of energy, which is not a plausible energy intake.

- ***Observable tractability.*** Sometimes, we don't observe all the actual cases, but only some subset of them. Say a problem is *observably tractable* just in case the *observed* instances are tractable.
- ***Explicit tractability.*** Suppose, on the basis of observable tractability, I argue a problem is actually tractable. However, I admit that I do not in fact know all of the actual cases that have arisen. Perhaps I have a dataset including all walks from all people with a fitness tracking app, but not all walks in general. With the cases of which I am aware, I can construct an argument that all of those are tractable. However, someone might ask on what grounds I claim that the dataset I have is representative of all the actual cases. One way to offer such an argument is to explicitly construct a region of parameter space and claim that *all* actual cases (not just the observed ones) are contained within it. If every actual instance in parameter space is then tractable, we say a problem is *explicitly tractable.*

  I should point out that explicit tractability is not just a property of the subject of a theory but also imposes an epistemological constraint on the theorist. I introduce explicit tractability because some think that the burden of proof it imposes on the theorist is in fact a reasonable one. I shall therefore argue that explicit tractability is in fact too strong on methodological grounds.
- ***Total tractability.*** An extremely restrictive demand is that *every* instance across parameter space should be tractable. This is very unlikely to be satisfied even by intuitively tractable problems such as arithmetic; there are numbers that are too large to add up.

Any true theory about a process that uses resources will trivially satisfy all actual tractability and explicit tractability. That is simply to assert the near-tautology that any process that is actually carried out with a certain level of resource usage will satisfy any true constraints on resource usage, in all actual and observed cases. Moreover, some *false* theories will also satisfy these constraints; but that is not a surprise and should not put us off: it would be very surprising if considerations from resource usage and tractability alone could carve up theoretical space in a way that makes immediately clear which theory is true. However, *some* false theories will fail to satisfy actual and explicit tractability, including, perhaps, some otherwise desirable theories. That can arise in two ways. First, they might suggest we carry out some process in a way that would use too many resources. Second, they might suggest that we carry out a process that we couldn't carry out at all.

I shall claim that a slight strengthening of actual tractability is appropriate in combination with asymptotic analysis, but that explicit tractability is too strong. This strengthening I call nomologically robust tractability. However, this requires some development of formalism from asymptotic analysis, so I postpone this argument to § 14.5. However, the remainder of the argument in this chapter does not require an elaboration of nomologically robust tractability, and can be presented in terms of the foregoing notions.

6 The tractable cognition thesis.

How, if at all, is subject-side tractability a *desideratum* in philosophy? van ROOIJ et al. (*Cognition*) offer the most systematic and well-known defence of a tractability constraint in philosophy, following earlier work by van ROOIJ ('Tractable').

The constraint is formulated in terms of the three levels of description of MARR (*Vision:* 22 ff). I first explicate the three levels (§ 6.1) and then the thesis (§ 6.2).

6.1 Marr's three levels.

The three levels apply to all information-processing systems. We can ask a number of questions about the system. At the 'top level' we may specify an input-output mapping for the system, the purpose of that mapping, and why the mapping is appropriate and adequate. Marr here gives the example of a cash register in a shop. He offers an account of the input-output mapping, its purpose and the adequacy of the mapping simultaneously. The purpose of the cash register is to offer an appropriate sum to pay given a basket of goods purchased. We can then reason as follows.

- Someone who buys nothing should not need to pay anything.

- Someone who buys nothing in addition to some other things should pay the price only of those other things.

Therefore, prices contain a ‘zero’ element as we define zero.

- Changing the order in which goods are purchased should not change the total one pays.

This amounts to commutativity.

- Putting items into separate piles and paying for the piles separately should not change the total price.

This amounts to associativity.

- Someone who buys an item and returns it for a refund should not pay anything.

This amounts to the existence of inverses.

This tells us that the cash register operates over an abelian group. The obvious one is the *integers under addition*; this then is input-output mapping appropriate to the task of adding prices, since addition is the operation given by the axioms for zero, commutativity, associativity and inverses. Moreover, *addition* is a well-defined input–output mapping, and so we have specified the mapping. Finally, we have given the purpose of the machine (generating total prices).

Marr calls this an ‘abstract computational theory’ of the device, or the *computational level* of description.

At a second level, we must *represent* these inputs and outputs (in this case, prices). For instance, we might represent these as moveable sliders on an abacus, or as transistors in an electronic computer. Given this representation, we must have an *algorithm* that mechanically converts the input representations into the correct output representations (relative to the mapping at the computational level).

Finally, at the third level, there is a physical implementation of the representation and the algorithm. For instance, a cash register might be implemented out of transistors. The visual system is made up e.g. of neurons.

6.2 *The tractable cognition thesis.*

We begin with what I call the Marr–van Rooij *tractability constraint* (van Rooij et al., *Cognition*: 14).

6.1. Suppose that some theory posits, at the computational level, that a system effects an input–output mapping M. The theory therefore is committed to the *existence* of a tractable means by which to implement that mapping M at the second (algorithmic) level.

A tractability constraint on cognition yields the following *tractable cognition thesis* (**TCT**) (van ROOIJ, 'Tractable': 946).

6.2. [T]he set of functions describing *possible* cognitive capacities is a subset of the set of *tractable* functions.

In both theses, however, we have simply asked whether an input–output mapping as a whole is tractable. This raises the questions posed in § 5.2. Do we require that the input–output mapping should be observably, actually or explicitly tractable?

According to van ROOIJ et al. (*Cognition:* § 9.5), only *explicit* tractability is sufficient. Consider the following dialogue.

ALICE Everybody walks in London.

BOB But walking from Epping to Croydon would be infeasible! Surely someone would have to take the train.

ALICE What does that matter? I've only observed people walking at most two miles. Clearly walking two miles is feasible. Why should I worry about walks from Epping to Croydon?

BOB You should revise your theory then. You should explicitly specify which walks you think people actually undertake in London. Only then is your theory that everybody walks in London reasonable.

Alice here has offered a theory that is *observably* tractable. Bob has taken no explicit view as to the actual tractability of the theory. Rather, he is suggesting that the burden of proof is with Alice. If Alice wishes to defend the subject-side tractability of her theory, she must explicitly formulate a theory with a restriction in parameter space that guarantees tractability on the instances within the restricted parameter region it concerns: thus, explicit tractability. That is what van ROOIJ et al. (*ibid.:* § 9.5) demand.[8]

I will later offer an argument in favour of actual over explicit tractability (§ 14). In terms of the dialogue above, I claim that the burden of proof is not wholly with Alice. However, for now, I shall examine some cases where tractability

8 It also seems that she demands total tractability, at least read literally.

constraints have been argued to be misplaced, and assess the merits of these arguments.

7 In partial defence of tractability.

I now expound and rebut two arguments against any tractability constraint whatsoever. The first is derived from one way of reading Marr, which is of interest given the Marrian formulation of the tractable cognition thesis (I argue that it is both exegetically and theoretically wrong). The second is a defence of the programme of rational analysis against tractability constraints that applies more generally..

7.1 *A Marrian argument against all tractability constraints.*

It is possible to read Marr in such a way that he would deny any tractability constraint whatsoever. Marr (*Vision:* 28) offers an argument against the criticisms of Chomsky offered by Winograd ('What Does it Mean to Understand Language?'). According to Marr, Winograd argues that Chomsky's theory 'cannot be made to run on a computer'. However, Marr seems to regard this criticism as misplaced: 'finding algorithms by which Chomsky's theory may be implemented is a completely different endeavor from formulating the theory itself'.

Whether or not Marr actually opposed tractability constraints, the argument attributed to him fails by his own lights.

Marr (*Vision:* 27) argues that the understanding of perception 'depends more upon the computational problems that have to be solved than upon the particular hardware in which their solutions are implemented'. It is true that Marr is clear that, methodologically, one ought to start with a *computational* problem. But Marr also accepts the constraint that *some* hardware must implement *some* solution to that problem. Therefore, if it can be demonstrated that *no* such solution is in the offing, a theory can be ruled out. Moreover, if it can be demonstrated that a problem is intractable, at least to the point where any such solution would be unreasonably resource-heavy, that would, by Marr's lights, be strong if defeasible evidence against the theory. Second, Marr (*ibid.:* 29) praises Chomsky and Lasnik ('Filters and Control') for offering a theory that might explain certain '*ad hoc* restrictions' in the input–output level of their theory about language in terms of '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 (if not necessity) for a computational-level theory to be consistent with *in-principle limitations* on the computational power of *any* system. Finally, MARR (*Vision:* 36–7) rejects views of vision as a 'completely invariant shape description from an image…in one step'. He reasons that such a task is 'almost certainly impossible' by a geometric argument. Again, the implication is that there is a constraint from the second (algorithmic) level to the first (computational) level.

Therefore, Marr's views are compatible with tractability constraints. Rather, Marr supplements the thesis. Considerations at the algorithmic level legitimately can *rule out* theories at the computational level. But they do not definitively *rule in* such theories either. And it is usually wise to begin at the computational level and proceed 'top-down', according to MARR (*ibid.:* 25). We should first examine 'the nature of the problem being solved', and only then 'examine the mechanism (and the hardware) in which it is embodied'. But this methodological maxim is compatible with tractability constraints of various forms.

7.2 The rational analysts against tractability constraints.

A second argument can be derived from the defence CHATER and OAKSFORD ('Rational analysis') offer of the programme of rational analysis.

> According to the programme of rational analysis, rationality, in an intuitive sense, appears to be at the heart of the explanation of human behavior, whether from the perspective of social science or everyday life (*ibid.:* 93).

Tractability offers a challenge to this programme that CHATER and OAKSFORD (*ibid.:* § 2.3) seek to surmount. The programme of rational analysis suggests agents often 'deriv[e] the optimal behavior function'. But optimisation is often intractable (on at least one reading of 'intractable'). It seems that this leads to a contradiction with cognitive systems' limits.

We can distinguish a number of responses on behalf of the rational analyst.

- The measure of tractability used is wrong.
- The measure is right, but is applied incorrectly, so that optimising is actually tractable.
- Optimising generally may be intractable. But it might be tractable in a more restricted context where the space of choices is much smaller. Or it might be possible at least to approximate optimisation in a way that is good enough.
- Optimising can be carried out by much larger systems (e.g. some of the optimisation might be carried out evolutionarily) so is easier than thought.
- It doesn't matter whether optimising (or approximating optimisation) is tractable at all.

Much of this thesis follows the first argument. (That is not out of sympathy, specifically, with the programme of rational analysis, though rational analysts could follow such a move.) However, Chater and Oaksford (*ibid.:* 110) seem to offer something like the last defence. If they succeed, it might seem that tractability constraints in philosophy have rather poor prospects. I shall now reject that last defence.

Chater and Oaksford (*ibid.*) do not explicitly distinguish the responses above. I will not respond to the other suggestions, since they simply suggest a refinement of how we apply tractability constraints, rather than suggesting we should ignore tractability altogether. However, it seems that one of the arguments to which they are committed amounts to the fourth point. I therefore propose to reconstruct that argument and reject it.

van Rooij et al. ('"As if'-explanations') systematically survey and reject the other responses as variants of the claim that cognisers behave 'as if' they optimise without being subject to a tractability constraint. I generally agree with their view, and propose only to isolate what I take to be an independent argument that requires a response.

Chater and Oaksford ('Rational analysis': 110) observe that birds manage to fly even though aerodynamic calculations may well be computationally intractable. The general suggestion is that an 'optimal behavior function is an explanatory tool, not part of the agent's cognitive equipment'.

One reconstruction of the Chater–Oaksford argument is that it simply amounts to the distinction between subject- and theorist-side difficulty. It is true that theorist-side intractability need not entail that there is any subject-side intractability. That is to say that, for instance, the theory of aerodynamics is not deficient simply because it does not allow us to tractably make predictions over long periods of time with the desired accuracy due to the effects of chaos. However, intractability results for optimisation are usually subject-side rather than simply theorist-side. For instance, a theory that predicts we maximise utility both posits that the *subject maximises utility* (subject-side) and then requires us to work out how the subject could maximise utility to make predictions (theory-side).

At this point, we ought to ask *how* subject-side optimisation can be explanatory if it is not 'part of the agent's cognitive equipment'. There is here a somewhat confused suggestion that agents (or birds) need only 'use successful algorithms', without 'mak[ing] the calculations that would show that these algorithms are successful'. It is of course true that a bird need not know enough aerodynamics to demonstrate that they know how to fly to fly. But it is hard to see how an agent or bird could 'use successful algorithms' without implementing them. Chater and Oaksford could retreat to the penultimate defence (that it's not just the bird but the evolutionary system doing the optimisation). That of course is a legitimate move, but it invites the application of a tractability constraint on the bird's 'cognitive equipment' and the evolutionary system in which they and their ancestors live. Unless we attribute optimisation to *something*, it is very hard to see how optimisation could be explanatory.

A qualification is in order. Some system's optimality is explanatory even though there is no computational system working through search space. Such explanations have been offered in respect of soap bubbles and the path of light. More generally, there are many equilibrium or variational explanations in science, which appeal to optimality. The error, I think, in this argument is that these putative counterexamples cannot plausibly be characterised as computational systems. We are therefore entitled to restrict tractability constraints to systems that *can* be characterised as computational systems, in which case no counterexample arises.

We might, admittedly, ask whether it is legitimate to distinguish these supposedly non-computational systems, which poses notorious difficulties (Piccinini, 'Computation in Physical Systems'). Since I do not propose to defend a novel theory of when computation is physically implemented, it is not possible for me simply to disregard this worry. I propose to defend three weaker these.

- There is a distinction between *bona fide* computational physical systems and non-computational physical systems.
- Cognitive systems fall on the computational side.
- Tractability constraints apply to the *bona fide* computational physical systems.

I shall not offer any detailed justification of the first claim.[9] Most cognitive scientists will accept at least some qualified version of the second claim (I would include, for instance, many dynamical systems under this rubric). At any rate, since Chater and Oaksford ('Rational analysis') offer the other defences (e.g. via approximation) in a computational idiom, it would be inconsistent for them to resort to this claim.

But what sort of tractability constraint does this subsection defend? I think it establishes the legitimacy of an actual tractability constraint. But it is not clear that it establishes the legitimacy of an explicit tractability constraint. Explicit tractability certainly is explanatorily useful. But it is not explanatorily *essential*. It is one thing to suggest that an algorithm plays a useful explanatory rôle if its implementation (even in the relatively easy cases) has nothing to do with the 'cognitive equipment' of an agent—I have dismissed this suggestion as quite implausible. It is another to suggest that the cognitive equipment of the agent is able to tractably deal with the actual cases, although we aren't quite sure exactly which cases really are actual, or which other cases the agent could handle. That said, this subsection does not *rule out* explicit tractability constraints either. The question will have to be settled later (§ 14).

9 For a recent optimistic defence of a so-called 'robust mapping' account of computation, see Anderson and Piccinini (*The Physical Signature of Computation*). For a general survey, see (Piccinini, 'Computation in Physical Systems').

8 Competence, performance and tractability.

van ROOIJ et al. ('Intractability and the use of heuristics in psychological explanations': 480 n 14) raises the suggestion that the distinction between competence and performance might obviate the applicability of tractability constraints. (She ultimately rejects the suggestion.) *Prima facie*, this is the view of CHOMSKY (*Aspects:* 3 ff), who suggests that competence is 'unaffected by such grammatically irrelevant conditions as memory limitations'. It is also, according to FRIXIONE ('Tractable Competence': 379, 383) the received wisdom.

This, I argue, is partly right. It depends, unsurprisingly, on how we understand competence. *Pace* Frixione and to a certain extent van Rooij, there is a perfectly reasonable sense in which theories of competence should not be subject to tractability constraints. But *pace* the received wisdom they attack, there are other reasonable senses of competence that should be subject to them.

But just what is the competence–performance distinction (CPD)? There is relatively little controversy over the proper interpretation of *performance*. CHOMSKY (*Aspects:* 4) defined it as 'the actual use of language in concrete situations'. But we can equally speak of performance as the actual use of other systems such as mathematical knowledge and reasoning (FRIXIONE, 'Tractable Competence': 379–80; DUPRE, 'Competence/performance distinction': 246).

Competence, however, is characterised as perfect knowledge of language by an 'ideal speaker-listener who is unaffected by such grammatically irrelevant conditions as memory limitations, distractions, shifts of attention and interest, and errors (random or characteristic) in applying his knowledge of the language in actual performance'. That, at least, is what Chomsky writes, but it is perhaps a little difficult to interpret.

I shall distinguish four senses of 'competence':

- *sensu* Marr,
- as internally inert cause,
- as internally dynamic cause and
- as idealisation.

In this, I partly follow Dupre (‘Competence/performance distinction’), though I disregard the exegetical question of what Chomsky (*Aspects*) really meant.

8.1 Competence *sensu* Marr.

Some authors appear to suggest that competence is simply Marr’s computational level, including van Rooij et al. (‘Intractability and the use of heuristics in psychological explanations’: 480 n 14), Frixione (‘Tractable Competence’: 381), and Marr (*Vision:* 28 ff). According to Dupre (‘Competence/performance distinction’: 248–9), this is exegetically wrong. The distinction he draws is of interest. Competence, according to Dupre, stands in a relation of cause and effect to performance. On the other hand, Marr’s levels of description are ‘“vertical”: they distinguish different, more fine-grained, ways of describing one and the same system’. For the reasons in § 7.1 and § 7.2, I think that competence *sensu* Marr should indeed be subject to tractability constraints.

8.2 Competence as internally inert *cause.*

The preferred view of Dupre (*ibid.:* 247) is that, as above, the relationship between competence and performance is the same as the relationship between cause and effect. In more detail, the CPD is just the distinction between a ‘potentially explanatory subsystem’ and ‘the downstream, observable behaviour on which inferences to the properties of this system are to be based’. But what kind of cause is competence?

Chomsky (*Aspects:* 4) suggests it is an ‘underlying system of rules’. Now, in general, a system of rules cannot itself be tractable or intractable. Tractability is, I have suggested, a matter of resource usage. But a system of rules is, in itself, internally *causally inert*. The elements of a system of rules do not stand in causal relations to other elements. For instance, consider an axiomatisation of a set theory. One axiom will not somehow ‘cause’ another axiom. Since resource usage is a matter of causal interaction, a system of rules cannot, in itself, use resources.

The application of rules, however, does use resources. It is there that it is natural to apply a tractability constraint. That said, relationship of a system of rules to empirical observation need not be particularly close or exhaustive. For instance, I know perfectly well how a Turing machine works, and can in some cases even work out their workings on paper by hand.[10] Clearly there is a fact of the matter about the model of computation about which I am thinking if e.g. I am writing down the standard definition of a one-tape Turing machine. But there is no reason to apply a tractability constraint to the object of my knowledge. What there is, however, is reason to apply a tractability constraint to the cases where I *apply* the rules. It is probably intractable for me to apply the rules of Turing machines on paper in order to find the smallest Mersenne prime of 10000 or more digits.

It might be objected at this point that it is a misnomer to call abstracta like systems of rules causes. Only *realised* systems of rules (e.g. realised biologically or physically) could count as causes. I think that what Dupre means (and what I mean) by 'cause' here is looser than that. To say that a system of rules is a *cause*, as I mean it, is to say simply that there are causes that *involve* the system. So my knowledge of how Turing machines work is biologically instantiated, and specific aspects thereof are *bona fide* causes. In this loose sense I have suggested internally causally inert systems of rules can be spoken of as causes, but that is all.

There is, however, a way of applying a tractability constraint to a combination of a hypothesis according to which some agent *(1)* applies some system of rules S *(2)* to some set of inputs I. Suppose we know that there is *some* system of rules S, but, for instance, that S might be S_1 or S_2. Suppose, moreover, that there is some $i \in I$ such that the application of S_1 to i is intractable, but the application of S_2 would not be intractable. And suppose, finally, that we know that S really is applied to i (without approximation or some other shortcut, and without failing). Then we ought to say that $S = S_2$. But this is not to say that there is anything wrong with S_1 *per se*, or that somehow it fails to respect a tractability constraint. Only the conjoined hypothesis that S is applied to some particular i can be held to a tractability constraint. There is nothing wrong with S_1 or S_2 *per se*, whatever one's view of the metaphysics of abstract system of rules.

10 Thanks to my anonymised supervisor for pointing out this example.

8.3 *Competence as* internally dynamic *cause.*

There are, however, other sorts of causes, to which a tractability constraint can more straightforwardly be applied; these causes also count as a 'potentially explanatory unobserved causal subsystem[s]'. The difference is when that causal subsystem *itself* has internal causal dynamics and therefore can use resources. As a criterion: a posited subsystem is internally *dynamic* just in case the theory attributes to it a course of state transitions—a doing, and not merely a content; otherwise it is internally inert. I shall argue that so far as this form of competence is concerned, tractability constraints are relevant.

Here is a somewhat contrived example illustrating how a tractability constraint might naturally apply directly to competence in this scenario. Suppose the interior of the skull were completely unobservable: we could not insert neuroëlectrical probes or conduct brain scans. But suppose we were also to theorise that, within the skull, some sort of computational system were present, tasked with arithmetic. Supposing we could, in fact, justify the claim that some sort of system tasked with arithmetic is indeed present, we could then naturally apply any tractability constraint on computational systems to that arithmetic subsystem, even though we couldn't observe it. That is because a subsystem actually carrying out arithmetic is capable of using resources. The same applies to less contrived Chomskyan attributions of competence: Chomsky does not think we e.g. have a perfect parser given the grammar of natural language that is implemented. If he were to, he would face tractability constraints. Instead, he faces a tractability constraint relative to *performance*, since we have an *imperfect* parser of sorts that bears some systematic relationship to his preferred grammars.

8.4 *Competence as idealisation.*

Finally, one reading of competence focuses on Chomsky's remark that competence is the knowledge of an '*ideal*' speaker (*ibid.:* 3) the suggestion then is that competence is a matter of idealisation away from resource constraints e.g. on memory.

This raises a more general question: what is the connexion between idealisation and tractability constraints more generally?

It is useful here to introduce a little philosophy of science. An idealised model of a system 'deliberate[ly] simplifi[es] or distort[s]…something complicated with the objective of making it more tractable [theorist-side] or understandable' (Frigg and Hartmann, 'Models in Science': § 1). Examples include '[f]rictionless planes, point masses, completely isolated systems, omniscient and fully rational agents, and markets in perfect equilibrium'.

Philosophers of science generally hold that there are two principal sorts of idealisation (*ibid.:* § 1). *Galilean* idealisations have the purpose of ensuring theorist-side tractability through simplification. So, for instance, it is easier to model classical mechanics in terms of point masses. *Aristotelian* idealisation is a matter of discarding properties that are considered, for one reason or another, irrelevant. For instance, suppose we are building a fairly humble bridge over the Thames. It is reasonable to disregard relativistic effects in building the bridge and focus only on classical effects in deciding e.g. whether it is sufficiently strong.

Some cases of idealisation will not affect considerations of subject-side tractability. For instance, on occasion, there is no question of subject-side tractability in the first place. Theorising about God, perhaps, falls in this category. More prosaically, so does theorising about the weather. Moreover, on occasion, an idealisation will have very little effect on tractability considerations either way. Consider again the walking example; at no point have we taken into account the relativistic mechanics of acceleration (which would appear to be an Aristotelian idealisation). Again, there is no reason to think that this radically changes the instances of the problem of walking we classify as (subject-side) tractable, or the regions of parameter space about which we are interested in theorising.

Suppose we have good reason to think that the doings of a theory's subject really are resource-bounded. Is it then legitimate to offer an idealised theory of the subject that ignores those bounds? I suggest that it isn't.

Let us examine Galilean and Aristotelian idealisations in turn. A Galilean idealisation is justified 'pragmatically' (Weisberg, 'Idealization': 641) it allows a theorist to avoid being 'stuck'. However, the legitimacy of the idealisation is that, in principle, it ought to be possible to 'de-idealize'. So a theory that is idealised in the Galilean sense comes with a promissory note: that the distortion can eventually be reversed. If, therefore, the distortion is at the level of ignoring a tractability constraint, the promissory note is that, eventually, some argument for tractability will be offered. Of course, no general rule as to when such promissory notes ought to be accepted can be given. In some cases, it may be that a related programme of research suggests demonstration of tractability is very likely; in other cases, we might have reason to think it is very unlikely.

Aristotelian idealisations, however, pose a somewhat more interesting problem. WEISBERG (*ibid.:* 642) describing *minimal* idealisations (he uses the term synonymously with 'Aristotelian'), describes them as 'contain[ing] only those factors that *make a difference* to the occurrence and essential character of the phenomenon in question'.

Is it legitimate, then, to ignore tractability on Aristotelian grounds? Not, I suggest, in a way that would change the dialectical situation. In some cases, tractability constraints don't and shouldn't significantly alter the trajectory we take in exploring theoretical space. Perhaps we are discussing theories of cognition that suggest we add up small numbers, and that's it. There is no suggestion we do anything more complex or involved. Our choice between these theories, then, has little to do with tractability: we aren't using tractability to rule out any of these theories. Everything (subject-side) in sight is very easy. So idealising away tractability constraints here is very much legitimate on Aristotelian grounds, because it does not 'make a difference to the occurrence and essential character' of the phenomena under study.

In other cases, perhaps tractability constraints pose a serious concern for some theory T_1, and point us towards some theory T_2. In this case, the question of tractability does significantly alter the 'essential character' of the phenomenon in question, if T_1 and T_2 meaningfully differ. Therefore, idealising away from tractability would not be legitimate here, precisely because doing so would make a significant difference dialectically.

The dichotomy therefore applies to any idealised Chomsky-inspired theory concerning speakers of a language. If it is a Galilean idealisation of performance, the purpose is to ensure theorist-side tractability. It therefore contains the usual promissory note that a de-idealised account of performance will respect tractability constraints. (We can read Chomskyan accounts of *performance* as attempting to do just that.) As an Aristotelian idealisation, it is legitimate insofar as the memory limitations don't change the 'ocurrence and essential of strings parsed. But, in that case, those memory limitations aren't dialectically decisive in the way that adherents of asymptotic arguments would hope; they wouldn't actually rule out large areas in theoretical space.

For now, I am satisfied by the conclusion that idealisation does not generally offer any reason to deprecate tractability constraints in any dialectically decisive way. I later return to how idealisation might in fact allow us to choose between e.g. actual, explicit and various related constraints, but this requires the development of asymptotic analysis (§ 14.5).

9 Recapitulation and conclusion.

I have offered a partial defence of tractability constraints in philosophical theory.

§ 5 Tractability is in the first instance a property of instances of a problem. However, we can also make attributions of tractability to classes of those instances in parameter space meaningful.

§ 6 It is legitimate, following van Rooij, to impose a tractability constraint on theories at Marr's computational level.

§ 8 It is also potentially legitimate to impose tractability constraints on internally causally inert systems, such as systems of rules, *in a context where they are applied*. Idealisation does not significantly change the picture.

We can now turn to the question of whether asymptotic analysis offers a suitable characterisation of tractability for these purposes.

III Asymptotic tractability and computationalism.

Consider the following computationalist claims about cognition.

- **Minimal computationalism:** 'core mental processes (e.g., reasoning, decision-making, and problem solving)'[11] are computations that could be carried out on a Turing machine.
- **Asymptotic computationalism:** **minimal computationalism** is true, and those computations are amenable to asymptotic analysis, e.g. it makes correct predictions.[12]

Both these views, I submit, are correct. A final view, however, is that we can asymptotically bound tractability; that is to say, we can, in terms of asymptotic analysis, offer an upper bound on what is tractable. Many of my fellow-travellers (i.e. asymptotic computationalists) think we can use such upper bounds to advance rather radical arguments about language and cognition (§ 1).

I claim that (most of) my fellow asymptotic computationalists are wrong, because they have applied the wrong tractability constraints: they demand the wrong kind of growth over too much of parameter space.

The dialectical purpose of this chapter is to explain why and how they are wrong in general. In later chapters, I then explain how that general story applies in specific instances involving language acquisition and processing. But I also wish to expound the formalism of asymptotic analysis used in these arguments in a philosophically motivated way.

§ 10 I expound the formalism of asymptotic analysis.

§ 11 I construct what I take to be the strongest argument for **asymptotic computationalism.**

§ 12 I expound the standard view relating asymptotic measures of tractability to cognitive science and philosophy.

11 I have borrowed this term from Rescorla ('The Computational Theory of Mind': § 2). However, the relationship between **minimal computationalism** and the 'computational theory of mind' therein expounded is not straightforward. For instance, **minimal computationalism** does not straightforwardly entail the *representational* theory of mind (*ibid.:* § 3.1).

12 It is a little hard to state properly what I mean here without actually introducing asymptotic analysis, but, to a first approximation, what I mean is that if asymptotic analysis predicts that resource usage should grow as a function of the size of a problem at a certain rate, and that growth is reflected in a physical system implementing that computation, then asymptotic analysis applies.

§ 13, § 14 I explain why this standard view is wrong, even accepting **asymptotic computationalism** and the strongest arguments for it.

10 Asymptotic analysis.

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

10.1. *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 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.

10.2. *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.

A fairly straightforward argument can be given, perhaps surprisingly, that two operations is not enough. Each operation either swaps two numbers or leaves them in place. So for arbitrary inputs a fixed sequence of two operations, only four possible rearrangements are possible. But the arrangements of a list $\langle a, b, c\rangle$ required to sort it could be any of the following:

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. (Three operations do suffice: compare-and-swap the first and second numbers, then the second and third, then the first and second again.)

10.3. *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.

10.4. *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 and return to the start?

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

10.2 *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. So-called *merge sort* involves repeatedly merging smaller sublists until the entire list is sorted, and some implementations require $O(n)$ 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.

10.3 *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) $n \log n$ comparisons. The travelling salesman problem and 3-SAT are thought to take exponential time (on any reasonable view of the operations available).[13]

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

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

$$\begin{aligned}\Theta(g(n)) = \{f(n) :\ & \text{there exist positive constants} \\ & c_1, c_2 \text{ and } n_0 \text{ such that, for all } n \geqslant n_0, \\ & 0 \leqslant c_1 \cdot g(n) \leqslant f(n) \leqslant c_2 \cdot g(n)\}\end{aligned}$$

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

Proof. For all $n \geqslant 1$,

$$0 \leqslant 1 \cdot n \leqslant 2n \leqslant 2 \cdot n$$ □

10.8. *Example.* $n^3 \notin \Theta\left(n^2\right)$.

Proof. Suppose otherwise. Then $n^3 \leqslant c_2 n^2$ for all $n \geqslant n_0$, but this is contradictory if $n > c_2$. □

10.9. *Example* (Katajainen and Träff, 'Analysis of merge-sort'). List-sorting takes $\Theta(n \log n)$ comparisons.

We will also use the following notation.

13 Strictly, this is the result of the exponential time hypothesis (which implies $P \neq \mathrm{NP}$); $P \neq \mathrm{NP}$ alone yields only a superpolynomial lower bound.

10.10. *Definition* (Upper bounds). Given a non-negative $g : \mathbb{R} \to \mathbb{R}$,

$$\begin{aligned} O(g(n)) = \{f(n) :{} & \text{there exist positive constants} \\ & c \text{ and } n_0 \text{ such that, for all } n \geqslant n_0, \\ & 0 \leqslant f(n) \leqslant c \cdot g(n)\} \end{aligned}$$

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.

In the next subsection, I explain the model of computation that is usually taken to answer these questions (the Turing machine model). The justification of that view is the **invariance** thesis, expounded in § 11.

10.4 *The Turing machine model.*

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

10.11. *Definition* (Arora and Barak, *Complexity*: § 1.1, informal.). Suppose that $f : \{0,1\}^* \to \{0,1\}$, that is, it takes as input a finite string of 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 $x \in \{0,1\}^*$; 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,

10.12. *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 $\rhd$, 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. $\delta : Q \times \Gamma^k \to Q \times \Gamma^{k-1} \times \{L, S, R\}^k$, where $k \geqslant 2$ is the number of tapes, is a *transition function* that describes the rules M uses from step to step. (The input tape is read-only, hence $k-1$.)

We initialise the machine in q_{start}. The machine contains $k \geqslant 2$ 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 machine is set to read, on the working tapes, some particular cell, containing the start symbol $\rhd$; the remainder of the cells are blank. The input tape contains the input, followed by the blank symbol on the remainder of its cells. 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 $\sigma_2', \dots, \sigma_k'$ and change state to q'. The heads are moved according to z. If the machine transitions into q_{halt} then it has *halted*.

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

Now, suppose $T : \mathbb{N} \to \mathbb{N}$, 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.*

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

10.5 *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, to explain why, if philosophers (or anybody else) 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 satisfiability 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 (about the same proiblem) 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. Trivially, P $\subseteq$ NP, but we do not know whether the inclusion is strict.

10.14. *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 $p : \mathbb{N} \to \mathbb{N}$. We say that $\mathcal{L} \in$ NP iff there is a polynomial-time verifier for $\mathcal{L}$.

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

We don't have a proof either way. But there is some reason to think this conjecture is true.

10.16. *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,

10.17. *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 $f : \{0,1\}^* \to \{0,1\}^*$
- 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 $\mathcal{L} \leqslant_p \mathcal{L}'$.

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

The Cook–Levin theorem then underwrites the assumption that there are many NP-complete problems.

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

10.19. *Theorem* (Cook, 'Complexity', Levin, 'Универсальные задачи перебора'). SAT is NP-hard (indeed, NP-complete).

By a further polynomial-time reduction, 3-SAT (the restriction of SAT to formulæ in 3-CNF) is also NP-complete (Karp, 'Reducibility among Combinatorial Problems').

10.6 *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.

10.20. *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 $e \in E$.

10.21. *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 (*ibid.*). However, we have an algorithm that runs in time $O\left(1.2738^k + kn\right)$ 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:

10.22. *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.*).

11 Asymptotic computationalism.

The dialectical structure of my argument concerning the correct view of asymptotic tractability is as follows.

- I construct what I take to be the strongest argument for **asymptotic computationalism**.
- I characterise the standard way of combining asymptotic analysis and tractability constraints in terms of parameter space.
- I then explain why the argument for asymptotic computationalism does not deliver that conclusion, and why it is not independently motivated.

In this section, I characterise what I take to be the strongest argument for **asymptotic computationalism**.

In § 10.3, I observed that results from asymptotic analysis are, in the first instance, proved relative to a model of computation. Time complexity is defined in terms of the number of fundamental operations required to solve a problem. Therefore, models of computation with different fundamental operations may generate different time complexity results for the same problem.

Many of the features of the model above seem quite arbitrary. For instance—

1. does it matter whether Γ contains only 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 accessed 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.

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'.[14]

14 I should like to address here one worry that might arise, which is that there is no reasonable notion of physicalism that does not collapse into naturalism (STOLJAR, 'Physicalism': § 5.4). In short, I think the argument can then be run again in terms of naturalistically respectable systems, i.e. systems that we empirically observe.

Chomsky and some others have argued that there is no bovious way of distinguishing what is 'physical' on purely conceptual or *a priori* grounds. Thus e.g. the Newtonian view of gravitation is supposedly physically respectable and yet did not conform to early modern notions of what a body should be.

I am quite sympathetic to this view, but, for simplicity, I have formulated the argument above in terms of physical rather than naturalistic reasonableness. See i.a. CHOMSKY (*New Horizons in the Study of Language and Mind:* 84 ff).

The invariance thesis suggests that all physically reasonable models of computation agree (within a certain bound) with each other in their complexity-theoretic classifications of the difficulty of problems. In particular, they agree on which problems take polynomial time.

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. it follows 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.

11.1. *Conjecture* (**Invariance**). 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.

A considerable number of authors who apply asymptotic analysis to human cognition do not explicitly justify the assumption that the invariance thesis applies to human cognition and simply assume that polynomial time in the standard class is of significance in the study of human cognition, especially in game theory.[15] 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 question '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.

15 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).

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).[16]

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.

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

16 Their gloss of how a quantum computer works is a little simplistic, but see also Aaronson, 'Physical reality'.

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.[17]

Invariance, I have argued, suggests that it is scientifically respectable to characterise problems as either polynomial-time or not, based on the study of a problem on a Turing machine. Moreover, **invariance** properly applies to cognition. Therefore, **asymptotic computationalism** follows (assuming **minimal computationalism**).

Since polynomial time therefore seems to be a scientifically respectable categorisation of problems, we might then wonder whether it is a good standard for the attribution of tractability to problems. I wish at this stage to make two observations. First, this conclusion does not follow from **asymptotic computationalism**. Second, it does not follow from the argument I have given for **asymptotic computationalism**. All we know is that we are likely to be able to correctly determine whether a problem takes polynomial time on cognitive hardware based on whether it does on a Turing machine; but it does not follow that there is anything interesting about polynomial time. Consider another (ludicrous) scenario. Suppose it were to turn out that all instances of a problem that take between 37 and 39 minutes on one reasonable model of computation were to take that same length of time on any other reasonable model of computation. That would be suggestive that the time between 37 and 39 minutes is of scientific interest. But it would not follow that taking fewer than 39 minutes is the mark of tractability. It could be scientifically interesting for all sorts of reasons, and tractability is only one of them. I now turn to

17 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*).

the question of whether polynomial time is of interest because it demarcates tractability.

12 Asymptotic tractability.

12.1 The standard formulations.

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

> *12.1.* *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, any reasonable model of computation will yield the same class of feasibly computable functions.

This then allows us to formulate the following.

> *12.2.* **P-cognition thesis** (van Rooij, 'Tractable': § 3): 'tractable' in the tractable cognition thesis is given by a *problem-level* attribution of tractability, determined by polynomial time in the standard class.

van Rooij (*ibid.:* § 5), however, slightly liberalises **P-cognition thesis** to a fixed-parameter constraint.

> *12.3.* **FPT-cognition thesis** (*ibid.:* § 5): 'tractable' in the tractable cognition thesis is given by a *problem-level* attribution of tractability, determined by fixed-parameter tractability in the standard class, where the fixed parameter is in practice small.

Let me state how this constraint then works in a little more detail.

- **A theory under examination.** The object of the constraint is to examine whether we should rule out some theory T concerning cognitive capacities.
- **A problem.** T will suggest that we solve some problem P. I will use the concrete example of a theory of reasoning that

suggests that we solve 3-SAT in order to evaluate the coherence of arguments other people make.[18]

- **A parameter space.** Given *P*, there is then some associated parameter space of instances of the problem. In this case, that is the set of formulæ in 3-CNF.
- **A parameter subspace.** We then construct a parameter subspace. In the case of the **P-cognition thesis**, we attend only to the 'size' parameter (e.g. the number of literals or the length of a formula), ignoring all other structure. In the case of **FPT-cognition thesis**, we attend only to size and the chosen fixed parameter. A simple example might be the number of clauses containing negated literals.[19]
- **The *worst* case.** Given a value for the size (and, optionally, a value for the fixed parameter), we find the instance that uses up the most resources to solve with that size and fixed parameter. This allows us to construct the worst-case parameter subspace.
- **Polynomial/fixed parameter constraints.** We then require that, as a function of size, resource usage grows no more than polynomially. If there is a fixed parameter, we require that, fixing an arbitrary value for the fixed parameter, resource usage grows no more than polynomially as a function of size.

There are, however, two other approaches we might take.

- **The *average* case.** Given some view about the distribution of cases, we take the *expected* resource consumption over that distribution.[20] We then apply the same polynomial constraint, but, instead, to the average rather than worst case.
- **The *worst actual* case.** Instead of taking the worst case from *all* of parameter space, we take the worst actual instance, and then apply the same polynomial constraint.

We can decompose the approaches above into two elements.

- ***Choice of case.*** The first is a choice of *case* (worst, average, worst actual).

18 I don't have any specific target theory in mind here—it is just an illustration; I use satisfiability for simplicity.

19 To my knowledge, this alone does not actually lead to any interesting parametrised complexity results. I am only offering this as a simple example of a fixed parameter.

20 See e.g. GOLDREICH, 'Average Case Complexity, Revisited'

- ***Choice of growth constraint.*** The second is a choice of asymptotic constraint to impose on those cases (in this case, polynomial).

I dispute both, in reverse order.

13 Moore's law and the Cobham–Edmonds thesis.

In this section, I dispute the choice of a polynomial growth constraint in the case of humans (i.e. the application of the Cobham–Edmonds thesis to human cognition.)

The justification for our accepting the Cobham–Edmonds thesis commonly found is, at its root, a crude inductive generalisation.[22] After all, the Cobham–Edmonds thesis might seem to admit obvious counterexamples. For instance, an algorithm that takes $\Theta\left(n^{2^{100}}\right)$ 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, and we might justify it on the basis that, in fact, it is indeed most of the time correct. Thus: 'it works'. And, indeed, that seems to be the case. For instance, there are surprisingly few 'galactic' algorithms, i.e. algorithms with polynomial asymptotic runtime that in practice are unusable.[21] 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 are of any interest (see also Dean, 'Complexity': n 10). We can therefore provisionally take the Cobham–Edmonds thesis to be a (defeasible) heuristic in the sense above.

The problem with such a crude justification is that it leaves someone who wishes to resist the application of the Cobham–Edmonds thesis to human beings in a rather difficult position. Either they seemingly take the anti-naturalist position of denying what is commonly accepted in computer science, or they abandon their initial position. But the situation is no more edifying for the proponent of the application of the Cobham–Edmonds thesis to human beings. To simply declare that 'it works', on the basis of an inductive generalisation largely de-

21 Helfgott, *Isomorphismes de graphes en temps quasi-polynomial (d'après Babai et Luks, Weisfeiler-Leman…)*; Lipton and Regan, 'David Johnson'

rived from computer engineering instead of cognitive science, appears to run the risk of overgeneralising.

My response to this situation is as follows. First, I shall attempt to explain *why* 'it works'—but for electronic computers only. Second, I then note that the explanation doesn't apply to human beings. This improves the situation in several ways. First, it offers a better justification of the Cobham–Edmonds thesis in computer engineering. Second, it then allows us to deny its application to human cognition, without abandoning minimal coimputationalism or a vulgar anti-naturalist denial of Cobham–Edmonds in general.

What, then, is my improvement on the justification that 'it works'? 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 too large may come within the scope of our hardware. Thus we find that computers have improved by many orders of magnitude since the first vacuüm-tube efforts of the mid-twentieth century (MOORE, 'Cramming'). It follows 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.

Although the connexion between Moore's law and the Cobham–Edmonds thesis, in my view, is fairly obvious, it seems that it is not noted in many standard surveys.[22] That might explain why in the literature it is an unusual position to be as thoroughgoingly computationalist as I am inclined to be

22 This is true of the following: DEAN, 'Complexity': § 2.2 and n 10; ARORA and BARAK, *Complexity:* § 1.6; GOLDREICH, 'Average Case Complexity, Revisited': § 1.2.3.5; DU and KO, *Theory of computational complexity:* § 1.4; PAPADIMITRIOU and YANNAKAKIS, 'Complexity as bounded rationality': 6–7; AARONSON, *P ?= NP:* § 3..

whilst rejecting the use of asympototic analysis in the study of cognition. Indeed, the only paper of which I am aware that has explicitly linked the two is a paper in intellectual property law (Chin, 'Abstraction and equivalence': 205–6).

> [A]s the processing speed of available computers increases exponentially over time…it is polynomial-time algorithms, and only polynomial-time algorithms, that are capable of harnessing these improvements to solve exponentially larger problem instances.

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').

The difference with humans 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 bound growth in working memory 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.

There is an independent feature of polynomial time that makes it an attractive class about which to theorise. Suppose we use one polynomial-time algorithm as part of another (we can call this composition). What is the runtime of the resulting algorithm? We don't, of course, know specifically; but we do know that it is polynomial. In other words, polynomial time 'is closed under the operations of explicit transformation, composition and limited recursion' (Cobham, 'Difficulty': 25). Closure, like model-invariance, is suggestive that a complexity class is of interest; but it does not straightforwardly tell us that it is interesting because it demarcates tractability.

At the relatively small scales to which we are limited, as creatures not subject to Moore's law, then, the difference between polynomial and superpolynomial resource bounds may not matter very much. On small inputs, even exponential bounds are not prohibitive. This much everybody accepts. van Rooij et al. (*Cognition:* § 9.9), for instance, say that it follows straightforwardly from a fixed-parameter constraint.

I am concerned here by *how* small is 'small enough' for the purposes of the **FPT-cognition thesis**. My central point is that 'small enough' as given by asympototic analysis might be *big* enough to include cognitively interesting inputs, contrary to the spirit (if not the letter) of e.g. this remark fron van Rooij et al. (*ibid.:* § 9.9) but they may, in fact, endorse this point (though they do not seem to do so explicitly).

> [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. On these grounds, we can rule out any defence via the fptc.

13.1. *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.

13.2. *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.

Unfortunately for philosophers, the culture in theoretical computer science tends to undervalue results about specific constants and bases on specific models of computation. Usually, it is considered more interesting to show that a problem is NP-hard than to show that, for instance, on some neurologically plausible model of computation, a problem takes $O(1.3^n)$ as opposed to $O(2^n)$ time. It would be foolish for philosophers to quarrel with theoretical computer scientists; there are perfectly good mathematical justifications for theoretical computer scientists' choice of interest. The burden instead lies with philosophers interested in aysymptotic arguments to provide the results themselves—but, since asymptotic analysis is not a straightforward field of research, it would be polyannish to expect immediate progress in adopting that strategy.

In particular, in asymptotic analysis, results with specific constants usually are the wrong sort. We can distinguish *constructive* proofs that a certain algorithm on a certain machine class runs in a certain time from *hardness* results that show that *no* algorithm can run faster than with a certain runtime. Asymptotic arguments, insofar as they are employed to rule out large areas of hteoretical space, are only useful if they are *hardness* results. But hardness results are usually proved with respect to the standard class; few are interested in the specific constants that come out on e.g. one-tape Turing machines or random-access machines.

14 Naturalism and cases.

Above, I offered three ways we might impose a tractability constraint: in terms of the worst, average and worst actual case. Which is right?

I am now going to mount an argument against worst-case tractability constraints as an *a priori* standard. The argument has two premisses.

1. Any philosopher or cognitive scientist who employs asymptotic analysis must be committed not only to deductively proved results in theoretical computer science, but also stan-

dardly accepted practices and conjectures in the field. Call this view *computational naturalism*.

2. One such practice is that the worst case isn't always the right case with which to think about tractability.

I shall justify these premisses in order. I then respond to several defences that could be offered of the worst case, and explain why the average case is not the right replacement for it. I finally argue that we may be justified in slightly strengthening actual tractability to what I call idealised counterfactual tractability.

14.1 Computational naturalism.

First, why ought anybody appealing to asympototic analysis to accept computational naturalism? After all, philosophers manage to make use e.g. of mathematical techniques routinely, without necessarily accepting the philosophical commitments of mathematicians. The situation in asymptotic analysis is, however, somewhat perplexing. Most of the field relies on unproven but widely believed conjectures. Here is one practitioner's view (Williams, 'Some Estimated Likelihoods for Computational Complexity': § 1).

> Complexity theorists generally have many intuitions about what is 'obviously' true. Everywhere we look, for every new complexity class that turns up, there's another conjectured lower bound separation, another evidently intractable problem, another apparent hardness with which we must learn to cope. We are surrounded by spectacular consequences of all these obviously true things, a sharp coherent world-view with a wonderfully broad theory of hardness and cryptography available to us, but—*gosh, it's so annoying!*—we don't have a clue about how we might prove any of these obviously true things. But we try anyway.

Moreover, there is a large body of results that, roughly, 'demonstrate strongly how the popular and intuitive ways that many theorems were proved in the past are fundamentally too weak to prove the lower bounds of the future'.

The consequence of this is that to make any meaningful progress in complexity theory, one must rely on one of these 'obviously true' conjectures. The force of these conjectures is the same as that of a theorem: on their intended interpretation, there is no inexactitude and there are no exceptions whatsoever. These *conjectures* are therefore different from *heuristics*.

A conjecture with a counterexample is just false, but an heuristic with a counterexample is not much worse off for it. Moreover, most philosophical work relies on this. So it is standard to assume, for instance, that P ≠ NP, or the exponential time hypothesis (to a first approximation, that NP-hard problems take exponential time, not just superpolynomial time), and so on.

The justification for this assumption from the point of view of philosophy is simply naturalism about computer science. It is not, therefore, particularly extraordinary. This is the same spirit that animates physicalism, indispensability arguments for mathematical platonism, some variants of moral naturalism, and so on (Papineau, 'Naturalism': § 1).

14.2 *The worst case in practice.*

Given what I have called computational naturalism, what follows for worst-case analysis? I now offer two examples where worst-case intractability has not especially moved computer scientists.

14.1. *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 $Ax \leqslant b$ and $c^T x$ is maximum,
- decide that $\{x \in \mathbb{R}^n : Ax \leqslant b\}$ is empty, or
- decide that for all $\alpha \in \mathbb{R}$ there is $x \in \mathbb{R}^n$ such that $Ax \leqslant b$ 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 were '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 superpolynomial 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 practical use of SAT solvers in industrial applications (Marques-Silva, 'Applications'). SAT is NP-complete, as we saw above. Moreover, many natural subclasses of cases in fact *must* take exponential time for resolution-based solvers, the kind actually in use (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 superpolynomial 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 approch 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).

I want at this stage to anticipate one possible objection. How much should we make of the examples given (linear programming and SAT)? Are they really representative of the field? Perhaps van Rooij could reject my argument for the second premiss on the grounds that worst-case complexity is very often the right measure.

I agree that worst-case complexity often is the right measure. My purpose here is to demonstrate that we should not assume that it is always the right measure, and that possibility should not be neglected. For this, the two examples I have given are sufficient. Satisfiability is a very general problem in terms of which other problems are often encoded. Moreover, when we encode other problems in SAT, we often find that the resulting runtime is practical and useful for industrial purposes. This suggests that the various other problems encoded also are in the same boat as SAT; some interesting but as yet unknown region of parameter space is tractable and does not contain pathologically hard instances. Hence the practicability of linear programming and SAT is therefore suggestive of a more general phenomenon amongst NP-hard problems, namely that there may be a significant subclass of practical relevance that differs from the worst case. Although linear programming is not NP-hard, similar considerations apply (i.e. we often encode problems in it and use the exponential-time simplex algorithm).

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 instance, 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.

14.3 *Some defences of the worst case rejected.*

I should like now to assess three possible responses van Rooij et al. (*ibid.*) could offer in defence of worst-case tractability.

The first objection is as follows; it is in fact the reply van Rooij et al. (*ibid.:* § 9.5) give to what they call 'the average-case objection', to which I return in § 14.4; but it equally amounts to a rejection of *any* departure from the worst-case framework.

> One possible objection may be that the worst-case input may never happen in practice and that, therefore, the worst-case complexity measure overestimates the real complexity of the computational-level theory. This objection becomes self-contradictory, however, once we accept that the hypothesized input domain is part of the computational-level theory…: The cognitive scientist postulating a computational-level theory $F : I \to O$ is, by definition, assuming that every input $i \in I$ can happen in practice.
>
> If there were to exist any input $i \in I$ for[sic] which the cognitive theorist believed it could not or does not happen in practice, then he or she should instead postulate a different computational-level theory $F' : I' \to O$, where input domain $I' \subset I$ is restricted so as to exclude those inputs not assumed to occur in practice.

Taken literally, this objection imposes an implausibly stringent standard of tractability, and contradicts any prospect of meaningful progress with asymptotic analysis in philosophy or cognitive science. Suppose that a computational-level theory is committed to assuming that 'every input $i \in I$ can happen in practice'. Then van Rooij et al. (*ibid.*) are effectively demanding total tractability.

This is because hard resource constraints do not actually grow with the size of the problem. If a human's working memory is of a certain size, it does not matter whether they are trying to prove Fermat's last theorem or work out what to cook for lunch. Their working memory does not expand simply because Fermat's last theorem is more interesting or sophisticated. Now, given some hard resource constraint, total tractability would rule out any algorithm with e.g. linear or even logarithmic complexity. The reason is simple: for any resource constraint (e.g. 100 hours of computation time), there is *some* input that eventually is large enough to exceed that resource constraint, assuming the resources used grow logarithmically or linearly. In fact, assuming there is some bound on resources, what this requires is *constant* complexity (that is $O(1)$). Thus we obtain a *constant-time tractability*: only theories that postulate constant-time input–output mappings are tractable. But very few algorithms run in constant time. If, 'by definition', this is what Marr's computational level means, the computational level cannot tenably be combined with a tractability constraint. Thus, literally construed, a dilemma arises for van Rooij et al. (*ibid.*). Either the input domain I is genuinely unbounded, in which case the constant-time absurdity just rehearsed follows; or I is bounded—as the domain of any human capacity presumably is—in which case asymptotic analysis is strictly inapplicable to it, and the **P-cognition thesis** itself must be read as an idealisation, whose legitimacy falls to be assessed by the criteria of § 8.4.

Now, second, we might suppose then that 'by definition' a computational theory has a commitment to the tractability of the worst case. No argument is forthcoming for this view, but even if it is, this requirement can be circumvented. We can translate talk of *actual* tractability of theories into talk of *worst-case* tractability in a systematic fashion as follows.

- We begin with a theory T postulating an input–output mapping $F : I \to O$. It is claimed that T is actually tractable.
- This is equivalent to claiming that there exists some T' that postulates some $F' : I' \to F[I']$ such that every actual case is contained in I' and F' is worst-case tractable.

Therefore, even if computational-level theories are somehow definitionally held to worst-case tractability standards, it is perfectly possible to *express* claims of actual tractability; there is no 'self-contradict[ion]', *pace* van Rooij et al. (*ibid.*).

We might raise a worry about trivialisation. There are finitely many *actual* instances of the exercise of any given human cognitive capacity, and they are bounded in size. There is therefore a maximum time that any finite-time algorithm would take on any of them. This would therefore appear to yield constant-time complexity. The fallacy in this line of thinking is as follows. Actual tractability does not constrain growth, but rather the subject-side resource usage on given instances that a theory predicts. Results about growth might suggest that actual tractability is unlikely to be met, but are only useful insofar as they do so.

This, I think, leads us to a more substantive and interesting question, which amounts to the distinction between explicit and actual tractability. Here, it seems that van Rooij is on the side of explicit rather than actual tractability. We grant, therefore, that it is possible that a theory can be (1) true, (2) worst-case intractable, but (3) actually tractable. Suppose, however, (4) we don't actually have a clear view about *what* the actual cases are, or any obvious general description of them. Is it then (5) rational to *believe* or support the theory?

One defence of worst-case tractability would then be that (4) makes (5) irrational. The burden of proof is on the theorist to explicitly show tractability, rather than simply to show its possibility and hope for the best.

My answer to this challenge is threefold. First, I think it is true that the absence of a clear characterisation of the actual cases indeed places dialectical pressure on a theorist. That pressure need not, of course, be decisive. There may be clear pressure the other way. Perhaps T_1 does very well on tractability constraints but does not integrate very well with other theories, and T_2 is only observably tractable and we can only guess whether it is actually tractable. Then this sort of dialectical pressure might push us to choose T_2.

Second, however, I don't think that the pressure is particularly great, at least in the general case. Here, I appeal again to computational naturalism. (1)–(5) apply perfectly well to the industrial application of SAT solvers. Does that mean that computer engineers and scientists who use SAT-solvers are acting irrationally? I don't think so. Naturalism suggests that philosophers who subscribe to (1)–(5) are not, therefore, *ipso facto* irrational either.

Third, it is often the case that there are very many other dialectically relevant factors in theory choice. Therefore, computational naturalism also suggests that we should be fairly cautious in ruling out theories on tractability grounds, unless they are observably intractable theories; mere lack of explicit tractability will be fairly indecisive in many cases.

14.4 *What about the average case?*

One alternative to analysis of the worst case is analysis of the average case: given some distribution over the instances of a problem, we require that *expected* resource consumption, rather than worst-case resource consumption, grow no more than polynomially with size. I am of the view that average-case analysis *can* be extensionally correct, but that it is the wrong standard to adopt. I offer three arguments.[23]

First, average-case tractability requirements could be extensionally wrong if they allow us to neglect important instances of problems that arise in practice. There are two ways this can happen. The first is if average-case analysis is defined as expected cost. If so, there could be actual cases of importance that nevertheless don't influence the average expected cost much. The intractability of a theory relative to those cases would then be ignored by the average-case analysis. The second is if we follow a 'typical case' approach to average-case analysis, by which we select some representative instances from problem space, and then examine asymptotic performance on those (GOLDREICH, *Computational Complexity*: § 10.2.1.1). If so, we might simply select the wrong ones.

In response to this argument, the proponent of average-case complexity could point out that if we select the *correct* distribution, the force of these counterarguments is blunted. In particular, what makes an actual case that arises important? Presumably it is its frequency. But, if a case has high frequency, it will affect the average more, and it ought to be included as one of the representative typical cases.

In turn, the opponent of average-case complexity has two further responses. One is that it is not obviously good enough that the cases on which the theory is wrong are infrequent. Suppose, for instance, a theory of language acquisition were to make incorrect predictions about a well-studied language spoken only by a community of a few thousand people. It would not, I submit, be any more theoretically reasonable to

23 One might object that I appealed to some sort of average-case analysis in the case of the *smoothed* analysis of the simplex algorithm (§ 14.2). However, smoothed analysis is something of an 'hybrid of the worst-case and average-case analysis of algorithms' (SPIELMAN and TENG, 'Smoothed analysis of algorithms': 296). That is, we take the worst case from parameter space and only then slightly perturb it. Smoothed analysis is not, therefore, particularly distributionally committed. Indeed, peturbation naturally yields instances we might easily otherwise have observed, consideration of which I advocate in § 14.5.

disregard the failure of this theory of language acquisition in this small community than it would vis-à-vis English.

Another is that it is not obvious what the distribution *is* in all cases. For instance, it is actually quite difficult to construct a model of the distribution of utterances heard by a child in the language-learning task (CLARK and LAPPIN, *Nativism*: §§ 6.3–4). Therefore, even if average-case complexity is correct, it is not a useful approach, because we are not in a good position to come up with useful constraints on the relevant distributions.

Finally, average-case analysis both faces formal difficulties and attracts somewhat less interest amongst theoretical computer scientists than worst-case complexity. This is not, of course, a watertight argument against the possibility of a philosophically useful programme of average-case study of NP-hard problems (for instance), but it is an argument against assuming that one exists without either (1) *qua* philosophers dedicating significant effort to it in the same way we e.g. prove novel results about modal logics or (2) convincing theoretical computer scientists to change their interests.

14.5 Counterfactually robust tractability.

In the concluding section of this chapter, I shall defend a slight strengthening of actual tractability constraints. Like explicit tractability, this standard is epistemologically/methodologically motivated, but it has significantly different implications.

Actual tractability is not always straightforward to empirically verify, because there are many actual but unobserved cases. One possibility is that we could simply use observable tractability as a proxy for it. Sometimes, that is fair enough: we might think that we have observed a fairly representative sample of problem instances. But sometimes it isn't.

I therefore propose an alternative standard: *idealised counterfactual tractability*. The procedure I propose is as follows. We attempt to *idealise* from the observable instances of a problem to a broader set of *counterfactually possible* instances we might easily otherwise have observed. We then use this as a proxy for the actually arising cases, and so we use idealised counterfactual tractability as a proxy for actual tractability. The worlds considered under this criterion of counterfactual possibility must, at least, be nomologically possible. They should, moreover, respect the same resource constraints that actual agents face.

Two points are in order. First, evidently, idealised counterfactual tractability might include cases that did not actually arise. In this sense, it is stronger.

The second is that the extent to which idealised counterfactual tractability is stronger than actual tractability is not very great. This is because of general considerations around the legitimacy of idealisations; here, we return to the considerations I indicated I would return to in § 8.4.

It's not straightforwardly clear to me whether we should classify this form of idealisation as Aristotelian or Galilean, so I propose simply to examine the theoretical consequences of either view. If counterfactual idealisation is Galilean, as I argued, it is accountable to a *de*-idealisation constraint, including with respect to tractability. That is to say: if we were to somehow learn of which cases *actually* arose, we should not come to a different verdict tractability-wise.

The case of Aristotelian idealisation, however, is more interesting. I want to sketch one reason we might think that idealised counterfactual tractability could legitimately go relatively further than actual tractability. Take the example of language acquisition. Suppose that the only actually existing language were English, but that humans were otherwise identical in the way they acquire language. Clearly, in this situation considered as actual, it would have been counterfactually possible to learn Arabic or Chinese. (The actual world is a witness to that.) But that would not have been an actual instance in this hypothetical. Would it be legitimate for an idealisation to include Arabic or Chinese, even if they were much harder? I suggest so. The scientifically relevant properties of our acquisition of language (or other objects of study) include not just *actual* evidence but also what would have happened in other situations (Williamson, 'Modal science': §§ 2–7). This applies just as much to cognition as it does to dynamical systems in physics. In other words, the aim then is something like *nomologically robust tractability*, and idealised counterfactual tractability is then a proxy for that standard, given the state of the art in science at a given time.

Let me now discharge the promissory note of § 5 with a working definition. A theory T satisfies *nomological robust tractability* just in case, in every nomologically possible situation in which the capacity T describes operates—within some specified range of conditions, e.g. holding fixed human physiology and the general character of the environments in which the capacity is exercised—the instances that arise are solvable within the agent's actual resource bounds. Idealised counterfactual tractability then corresponds to the attempts of actually existing working scientists to determine nomological robust tractability: given the state of science at a time, idealising from the observed instances is how we estimate whether nomologically robust tractability holds. General conisderations from the philosophy of idealisation can help us to avoid either strengthenings that lead towards total tractability or weakenings that lead to actual tractability.

15 Recapitulation.

I have offered two principal arguments about the application of asymptotic measures of tractability.

§ 13 Exponential bases, additive and multiplicative constants, and realistic input sizes matter in imposing subject-side tractability constraints, even though asymptotic analysis ignores or obscures them.

§ 14 Worst-case tractability is too demanding as an *a priori* default; the case studies of the following chapters argue that the default in fact fails in the target literature. Actual tractability may (on occasion) be too weak. Average-case tractability is likely too weak as well. Explicit tractability tries to be a useful strengthening, but is also misguided, because it contradicts the spirit of computational naturalism. Idealised counterfactual tractability, however, is just right.

Now I turn to examination of some cases where tractability-based arguments from asymptotic analysis have been offered. I claim that they do not respect the foregoing principles, and, therefore, that they are unsuccessful.

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 (§ 16). Pagin claims to fill a lacuna in the argument with asymptotic analysis (§ 17). I reject the argument for several reasons; some have nothing to do with asymptotic analysis, but some do, and they offer a useful illustration of the points I make in § 13 and § 14.

16 Compositionality from (in principle) 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 '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.

16.1. 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.

16.2. 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'.[24] 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*.

Pagin therefore attempts to motivate a further constraint. We begin with a first principle.

16.3. *Definition* (*COG*). 'Other things equal, the greater the computaional complexity of an interpretation task, the greater the cognitive difficulty of the interreptation' (*ibid.:* 553).

A computability constraint therefore ignores two important factors. First, there are bounds (attention, memory, lifespan and the number of neurons, for instance) on human cognitive capacities. Accordingly, there are bounds on the difficulty of interpretation tasks humans are able to complete. These bounds can be exceeded by merely computable semantics. Thus

16.4. *Definition* (*TRC*). Any plausible semantics fo rnatural language is computationally tractable.

Pagin's revised abduction therefore has the following form. The *explicandum* is our comprehension of novel sentences. The *explicantes* are restricted to tractable means by which we could

24 Heck, 'Does Davidson Argue for Compositionality?': 2; Malpas, 'Donald Davidson': § 3.1; Pagin and Westerståhl, 'Compositionality II': § 1.1

learn their meanings, by TRC. Pagin then attempts to demonstrate that tractability entails compositionality by the contrapositive: to a first approximation, noncompositionality entails intractability. (He will require a certain kind of background syntactic complexity for the result to go through.)

17 Compositionality from tractability.

In this section, I reconstruct Pagin's formal framework and results.

We begin with *grammatical terms*, given by a partial algebra $\langle \mathrm{GT}_L, \mathrm{AT}_L, \Sigma_L \rangle$ of grammatical terms, atomic terms, and finitely many partial syntactic operations (*ibid.:* definition 1). GT_L is the closure of AT_L under Σ_L. A *semantic function* is some function $\mu : \mathrm{GT}_L \to M$ into an arbitrary set of meanings.

μ is *standard compositional* (definition 3) just in case, for each n-ary $\alpha \in \Sigma_L$, there is some meaning operation $r_\alpha : M^n \to M$ such that, whenever $\alpha(t_1, \ldots, t_n)$ is defined,

$$\mu(\alpha(t_1, \ldots, t_n)) = r_\alpha(\mu(t_1), \ldots, \mu(t_n)).$$

We then slightly liberalise this. Instead of a single semantic function, take a finite family $\mu_1, \ldots, \mu_j$ with a designated starting member; a complex term may be evaluated by one member of the family while its parts are evaluated by others, the choice of evaluating function being fixed by the syntactic operation and the argument position (the meaning operations being correspondingly indexed $r_{\alpha,i}$). A family of this kind is *general compositional* (definitions 4–5). General compositionality supposedly accommodates certain putative counterexamples to standard compositionality, but I leave this question beyond the scope of the thesis. In any case, Pagin's results concern general compositionality.

By contrast, a *recursive* meaning function need only satisfy

$$\mu(\alpha(t_1, \ldots, t_n)) = r_\alpha(\mu(t_1), \ldots, \mu(t_n), \boldsymbol{t_1}, \ldots, \boldsymbol{t_n}).$$

The bold arguments $(t_\prime, \ldots, t_n)$ show that the meaning operation may depend on the syntactic structure of the constituents directly, instead of via their meaning. Consequently, synonymous constituents may make different contributions to the meaning of an expression overall. Nevertheless, recursive meaning functions are still computable, by definition. Therefore, restrictions to recursive functions do not motivate compositionality.

We now offer a model of the cost of a computation, in order to derive a measure of time complexity. Pagin proposes to compute the meaning of a term in the object language by term rewriting into a canonical meaning representation in the metalanguage, in the form of a normal form (i.e. one to which no further rule applies). Davidson's reference clauses for 'the father of' illustrate the idea (*ibid.:* 567): with the rules $\mu(a) \to$ Annette and $\mu(F(v)) \to \boldsymbol{F}(\mu(v))$—where F is the object-language operator and $\boldsymbol{F}$ its metalanguage analogue—we have the four-step derivation

$$\mu(F(F(F(a)))) \to \boldsymbol{F}(\mu(F(F(a)))) \to \ldots \to \boldsymbol{F}(\boldsymbol{F}(\boldsymbol{F}(\text{Annette}))).$$

Each rule application counts as one step for the purposes of time complexity, and the size of a term is its number of symbol occurrences. The *input complexity* $C^i(k)$ of a rule system is then the largest number of steps needed to normalise an input of size k; the *output complexity* $C^o(k)$ is the corresponding quantity relative to the size k of the *output*, the canonical meaning representation.

Pagin insists on measuring tractability in terms of both input and output complexity. The derivation of the meaning of an object-language expression might have a low *output* complexity simply because the normal form giving its meaning is inflated, or *vice versa* (*ibid.:* 565–566). Tractability then is given by bounding both input complexity and output complexcity accordingly (*ibid.:* 565).

Pagin's positive result concerns rule systems in 'direct' form, in which every rule rewrites $\mu(\alpha(v_1, \ldots, v_n))$ in a single step to a metalanguage operation applied to $\mu(v_1), \ldots, \mu(v_n)$—the rewrite-system image of (general) compositionality. Every application of such a rule contributes at least one symbol of the eventual output and introduces nothing that later needs to be eliminated, so a canonical output of size k is reached in at most k steps: complexity is *linear*, which within this framework is the minimum available; such systems are 'maximally time efficient' (*ibid.:* 571).

Pagin's argument that noncompositionality entails intractability is by the generalisation of the following exmaple. *Godzilla* contains two atoms, α and β. It also contains a binary operation δ, which is freely applicable: the grammatical terms are just the finite binary trees over the atoms. The metalanguage has an atom l, unary operators m and h, and a binary operator O. The object-language atoms are synonymous: $\mu(\alpha) \to l$ and $\mu(\beta) \to l$. However, whether m or h appears in a given position in teh meaning of a complex terms depends on whether α or β occurs in the corresponding position. Since the atoms don't differ in meaning, this dependence violates compositionality (though not recursivity). Gozilla (*ibid.:* § 8.3.3) contains, *inter alia*, the following rules.

$$\mu(\delta(u, v)) \to f(\mu(u), \mu(v), uv).$$

$$f(x, y, u, \delta(v_1, v_2)) \to O(f(x, y, u, v_1), f(x, y, u, v_2)).$$

Pagin calculates that rewriting a term with the μ-argument $\delta(\delta(\alpha, \beta), \delta(\beta, \alpha))$ into a term whose μ-arguments are $\delta(\alpha, \beta)$ and $\delta(\beta, \alpha)$ takes ten steps where a direct system takes one, and leaves four copies of each of the two embedded μ-terms where a direct system leaves one of each. The work remaining is therefore multiplied at each step, and Pagin proves that the input complexity is factorial: in the worst case, interpreting a term containing k atomic symbols takes at least $(k-1)!$ rewrite steps (proposition 7).

The generalisation of the argument is somewhat tricky (*ibid.:* § 9). Pagin himself gives an example of a noncompositional semantics that is tractable. The explosion is derived from an operation of arity at least two that is freely embeddable. But even then, compositionality is not sufficient to yield tractability.

Assembled, the argument is this.

1. Other things equal, the greater the computational complexity of an interpretation task, the greater its cognitive difficulty ((COG)).
2. Speakers reliably work out the meanings of novel sentences in real time with bounded resources; so, given (COG), any plausible semantics for a natural language is tractable (TRC) and, other things equal, minimally complex ((MIN)).
3. With combinatorially unrestricted syntax, a noncompositional recursive semantics has at least factorial input complexity, and is therefore intractable (Godzilla).
4. General compositional semantics in direct form attain linear—minimal—complexity, and only general compositional

semantics attain minimal complexity (here Pagin describes 'direct' systems).

5. Natural-language syntax permits free embedding of binary operations, so the tractability of its semantics must be secured in a syntax-independent way.

6. Therefore general compositionality is a necessary feature of the best available explanation of real-time comprehension; by inference to the best explanation, natural-language semantics is general compositional.

The objections of the next section are indexed to this argument. The first questions the construal of the interpretation task presupposed by the second premiss; the second questions whether TRC can be applied to a semantics at all; the third grants both and attacks the worst-case reading embodied in the third and fifth premisses.

18 Three arguments against Pagin.

In this section, I offer two preliminary arguments against Pagin that do not depend on the asymptotic machinery of Chapter III, and then a third, which applies the considerations of § 13 and § 14.

18.1 Modelling meaning.

Let us distinguish two tasks:

1. to establish the meaning of an expression in a manner fit for systematic enquiries into language of the kind carried out in semantic theories and linguistics; and

2. to establish the meaning of an expression in a manner sufficient to communicate, say what one means, understand others, and so on.

It may be that the two tasks are the same. But it may not be. The task by which Pagin is concerned is to determine the meaning of an expression in the object language as a canonical expression of the metalanguage. This may indeed be how we should think about meaning for the purposes of linguistics. But it does not follow that it is necessary for the purpose of an ordinary competent speaker of a language.

I should like to sketch one way in which this view might, in fact, be misconceived. To a first approximation, according to conceptual rôle semantics, 'the meaning or propositional content of an expression or attitude is determined by the role it plays in a person's language or in her cognition' (WHITING, *Conceptual Role Semantics*).

That role, in particular, is given by the extent to which a speaker 'is…prepared to make certain inferential transitions', such as from 'x is a male sibling' to 'x is a brother'.

Conceptual rôle semantics offers a fairly straightforward delineation between the two tasks above. It seems to be sufficient to attribute knowledge of language in the second sense that the expressions of a language should have the appropriate inferential rôles. But that only requires that a speaker should be disposed to make the appropriate transitions in the right circumstances.

To be so disposed does not require that the speaker should know or even have an internal state reflecting all the possible dispositions that might arise. To demand that would be a little like demanding of someone who *could* walk anywhere within ten kilometres that they should *actually* walk to all those locations. Indeed, it is quite possible that someone could not in fact walk to *all* those locations but only some subset of them. And it is also quite possible that a competent speaker could be disposed to make all the right linguistic transitions, and even, when asked, articulate those transitions, but could not make all of them at the same time, due to their finitude.

It therefore is conceivable that Pagin has inappropriately subsumed the second task (that of a speaker's establishing a meaning of an expression) under the first (that of the linguist or semantic theorist studying language).

18.2 *Semantic competence.*

Tractability constraints apply to realised cognitive systems with internal causal–computational dynamics. They don't, however, apply to internally causally inert static systems of rules (§ 8). We might think that the semantics for a fixed language belongs to the latter. Thus the semantics would fix meanings of expressions (perhaps even in some metalanguage). But it would not tell us how *we* map expressions to their meanings. On this view, TRC misapplies a tractability predicate: what must be tractable, if anything, is the processing that deploys the semantics, not the semantics itself. In particular, there need not be any guarantee that, in general, we can map expressions to their meanings as we would if our performance were to perfectly match our competence. It might turn out that competence is noncompostional, and, accepting Pagin's conditional argument *arguendo*, perfect performance would be intractable. However, actual performance could still be tractable because it is restricted only to relatively easy cases. This mirrors the Chomskyan distinction between *unacceptable* and *ungrammatical* sentences (CHOMSKY, *Aspects*: 10–11). One response Pagin could offer is that the competence–performance gap is not sufficiently large for this response to go through. That, in effect, is the objection I address in the next subsection.

18.3 Which cases? How big are they?

Suppose that neither of the foregoing arguments holds, so that some tractability constraint does apply to a semantics via its use. Pagin's argument nevertheless plausibly falls foul of the difficulties I outlined in § 13 and § 14.

It is useful first to examine cases where compositionality is allegedly violated. (If there were no live proposals for noncompositional theories of meaning, Pagin's argument would be superfluous.) 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. (Note that we might in fact take δ in Pagin's Godzilla example to combine a subject and a sentential complement, which would correspond to proposiotional attitude reports.) Pagin would reject such semantics on the grounds that it is intractable. How might its supporters respond?

1. A simple response might be that inputs are of bounded depth: nobody asserts, or needs to interpret, attitude reports nested fifty deep, so an explosion out along the depth axis is irrelevant to the actual use of language. Combined with a relatively forgiving superpolynomial bound (§ 13), we might think superpolynomial complexity is not tremendously prohibitive. On this response, it is no surprise that we find sentences such as the following difficult to understand (or even to parse):

(3) 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)).

This response, however, does not quite work. Bounds on depth alone do not straightforwardly correspond to patterns of difficulty. For instance, the following sentence is perfectly comprehensible.

(4) 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, yet it is far easier than (3), whose embedding is only three deep. A depth bound predicts uniform ease below the bound and failure above it; what we find instead is ease at depth nine and difficulty at depth three.

2. A better response, I submit, is that the difficulty of instances tracks their *structure*, and not (only) their size (§ 14). (3) is hard because it is a *centre*-embedding: each embedded clause interrupts the clause containing it, so all the interrupted dependencies must be held open at once. (4) is easy because it is an *edge*-embedding (that is, a peripheral, left- or right-branching, embedding): each clause can be closed off as the next begins.

This, of course, is what the literature in psychology and linguistics predicts (Blaubergs and Braine, 'Short-term memory limitations on decoding self-embedded sentences'; Miller and Isard, 'Free recall of self-embedded english sentences'; Foss and Cairns, 'Some effects of memory limitation upon sentence comprehension and recall'; Miller and Chomsky, 'Finitary Models of Language Users': § 2.1).

In short, the pair of examples exhibits a double dissociation between size and difficulty: (4) is easy at embedding depth nine, while (3) is hard at depth three. The structure of an instance class can radically change the difficulty of the problems to which its instances are inputs, even where the size (the number of embeddings, or the length of a term) does not. Thus the objection of § 14 arises: there is an interesting, structurally characterised subspace of instances that may well be tractable even though the worst case isn't.

According to Pagin, noncompositionality generally leads to exponential blowup when binary operations are freely embeddable. Cases like (4) confine complex embeddings only to one argument place, and put an atom in the other, which is precisely the structure of another example Pagin gives of a noncompositional recursive semantics that remains tractable (Pagin, 'Compositionality, computability, and complexity': 581). Therefore, a hypothetically (worst-case) intractable semantics of propositional attitude reports would appear perhaps even to be confirmed by the relative difficulty of examples like (3) and (4).

What follows is a difficulty for Pagin's assumption that tractability must be secured syntax-independently. If this is a constraint on what actually or counterfactually arises, TRC can be satisfied by noncompositional semantics, so the abduction fails. If it is a constraint over combinatorially unrestricted instance space, TRC is unmotivated, because it's not at all established that we do in fact quickly comprehend arbitrary sentences (that Chomsky would call grammatical but unacceptable), or are capable of comprehending them at all. Either way, the argument is unsound.

For these reasons, I submit that Pagin's asymptotic argument does not establish that meaning is compositional. Other considerations will have to guide us.

V Asymptotics and language acquisition.

Nativists, to a first approximation, argue that we have innate knowledge of language in the form of universal grammar. Some nativists appeal to asymptotic analysis. In this chapter, I am partly concerned to argue that they are wrong. But I also claim that some of the debate that has followed their arguments is also misdirected; the correct response to these nativist arguments, in effect, is given in Chapter III, in rejecting the naïve application of polynomial worst-case constraints. In § 19, I outline the underlying formalism and offer some *a priori* criticisms of various learning-theoretic régimes. In § 20, I examine the alleged significance of learning-theoretic results, and state where I think the misunderstanding is.

19 Asymptotic and PAC-learnability.

In this section, we shall be concerned with certain arguments from *learning theory*—a field that makes central use 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 the reasons suggested in § 13 and § 14.

19.1 Formalism: PAC-learnability.

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

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

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

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

19.4. *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.

19.5. *Definition* (representations). A *representation class* over X is a pair $\langle \sigma, C \rangle$ such that $C \subseteq (\{0,1\} \cup \mathbb{R})^*$ and $\sigma : C \to 2^X$. 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).

19.6. *Definition* (examples). The *positive examples* $\mathrm{pos}(c) = \sigma(c)$ of a concept $c \in C$ are those that fall under the concept. The *negative examples* $\mathrm{neg}(c) = X - \sigma(c)$ 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 $x \in \mathrm{pos}(c)$ and 0 otherwise.

We will sometimes 'parametrise' representation classes and instance spaces.

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

19.8. *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 a 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.

19.9. *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 $\mathrm{pos}(c)$.

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

19.10. *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* $S = \langle x_1, b_1 \rangle, \ldots, \langle x_m, b_m \rangle$ is a finite sequence of labelled examples.

19.11. *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.

19.12. *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.

19.13. *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 be the one for ‘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 fixed but arbitrary probability distributions. D_C^+ is a fixed but arbitrary probability distribution over pos(c), and D_C^- over 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: the learner has access to a single oracle drawing labelled examples of the target representation from one distribution D over X, rather than separate positive and negative oracles, and error is measured relative to that single distribution.

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

When $\mathrm{err}(h; c, D) \leqslant \varepsilon$, h is ε-good; otherwise, it is ε-bad.

We can now define learnability.

19.15. *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 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 $\mathrm{err}(h; c, D) \leqslant \varepsilon$.

C is *polynomially learnable* if it is learnable by some polynomially evaluable H.

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

By *asymptotic learning theory*, I simply mean the study of the asymptotic growth of sample, time or space complexity with respect to any or all of the parameters above. We might, for instance, impose an imposition on the distribution, or similar; I still count that as asymptotic.[25]

19.2 *Justifying PAC-learnability.*

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 mistakes by the number of sentences uttered by them.

Let us now examine a few elements of the PAC-learnability régime and their justification.

1. **Polynomial runtime.** The learning task must be completed in polynomial time. I have little more to say at this point, and simply refer to Chapter III (in particular § 13 and § 14). Standard textbooks on learning theory largely follow the rest of the literature in their justification of a polynomial constraint (Valiant, *Probably approximately correct:* § 3.4; Clark

25 I shall ignore identification in the limit (Gold, ‘Identification’). It requires separate treatment.

and LAPPIN, *Nativism:* § 7.1) or leave it implicit (KEARNS and VAZIRANI, *An Introduction to Computational Learning Theory*).

2. **Polynomial sample complexity.** The learning task must also be completed (with respect to the success and error parameters) with polynomial sample complexity. One way of intuitively motivating this is that to draw a sample is not costless, in that it takes a certain period of time and/or quantum of computational effort. I don't object to this argument, but the same arguments about polynomial runtime also apply to sample complexity.

3. **Distribution freedom.** The PAC régime is *distribution-free*; that is to say, it requires learning over an arbitrary distribution. This incurs precisely the same problem as worst-case analysis: sometimes, all 'reasonable' distributions admit tractable learning, and there are pathologically bad distributions that do not arise in practice. Thus we have results where distribution-free learning is intractable but not e.g. learning over a uniform distribution (MANSOUR, 'An O(nlog log n) learning algorithm for DNF under the uniform distribution').

A more promising method is to attempt to construct an idealised but realistic distribution and assess learnability relative to that (CLARK and LAPPIN, *Nativism*: § 6.1 ff).

What justification for distribution-freedom has been offered? VALIANT (*Probably approximately correct:* 76) offers only a promissory note: '[t]he bounds obtained for worst-case distributions often provide useful guidance on how many useful examples to use in practice'.

de WOLF ('Applications': § 2.4.4) offers a somewhat different argument: it may be that distribution-free learning is intractable but a fixed and realistic distribution will not yield intractability; but, if so, that vindicates nativism: 'a child must have a certain bias which directs the way it acquires language'. If this is a form of nativism, it is really a very weak form of nativism. The fixing of the distribution is a property of the environment rather than the child, and most empiricists are generally happy to concede that the *environment* might only be fruitfully modelled by a limited family of distributions. Moreover, it is not inductive bias as ordinarily meant—it says nothing about the hypotheses the child privileges.

4. **Arbitrarily small error and success parameters.** 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. (*Linguistics will thrive*), 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.

The near-universal success of first-language acquisition may make it plausible that δ should at least be *small*. But, as noted at the opening of this section, it is a further step—and an unjustified one—to require that δ be *arbitrarily* small at merely polynomial cost. And it is even less obvious that ε should be considered to become arbitrarily small relative to the target language (e.g. that of one's parents).

In response to this argument, it might be pointed out that setting arbitrarily small error and success parameters yields the same *extensional* definition of learnability as PAC-learnability, by 'boosting'. The idea is that an algorithm that successfully reduces the error and success parameters to a fairly achievable constant can then be *repeated* (within a polynomial bound) in order to yield an arbitrarily small error with an arbitrarily high probability of success (Kearns and

Valiant, 'Limitations': cap. 4; Freund and Schapire, 'A desicion-theoretic generalization of on-line learning and an application to boosting'). I therefore submit that any learning régime with arbitrarily small error and success parameters can only be justified if boosting is available. The standard justification for boosting is that the samples are drawn *independently*. I now turn to the justification of that view.

5. **Independence.** The task of the learner in the PAC-régime is to learn from an arbitrary *but fixed* distribution. This, however, does not appear *prima facie* to characterise linguistic stimuli (Clark and Lappin, *Nativism:* § 5.2.1; Chater and Vitányi, ''Ideal learning' of natural language': § 6.6). Clark and Lappin (*Nativism:* § 5.2.1) nevertheless argue that it is reasonable to assume independence as an idealisation, because almost all computational studies assume it. The justification of this view is, in effect, that the independence assumption underwrites in effect only the application of various formulations of the *law of large numbers*, which, in effect, show that inferences from large samples are increasingly accurate as the size of the samples increases. In a more generalised 'ergodic' régime in which, to a first approximation, the sample proceeds through sample space freely, such laws of large numbers apply, and, therefore, analogues of PAC-learnability results often be proved. In particular, it is possible to use *boosting* algorithms in a liberalised ergodic régime (Lozano et al., 'Convergence and Consistency of Regularized Boosting With Weakly Dependent Observations').

What, therefore, ought we to expect from the PAC-learnability literature vis à vis natural language? Perhaps surprisingly, there is a case to be made that the assumption that error or success parameters can be made arbitrarily small is in fact, extensionally, harmless. This is because it can be justified whenever boosting is avilable, and boosting appears to be available in a liberalised erdogic régime that therefore may not be unduly unrealistic so far as natural language is concerned. On the other hand, the standard criticisms about polynomial time apply, and distribution-freedom seems potentially unwarranted.

20 Alleged implications of learnability results.

I will begin with a fairly representative result, according to which the *regular languages* are not PAC-learnable. Since the regular languages are at the bottom of the Chomsky hierarchy and hardly sophisticated enough to include natural languages, their unlearnability is often regarded as suggestive. (I agree that their unlearnability is indeed suggestive, but not of UG, as some have thought.) I then briefly outline some considerations that have arisen in the debate over the interpretation of these results. An interesting feature of this debate is that it largely disregards the considerations of Chapter III: it naïvely assumes that superpolynomial complexity is prohibitive and adopts a worst-case measure of tractability.

20.1 *A representative result: the PAC-unlearnability of the regular languages.*

Kearns and Valiant ('Limitations': § 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.

20.1. *Definition.* Given a polynomial p, let ADFA_n^p denote the class of acyclic deterministic finite automata of size at most $p(n)$ accepting only strings of length n, and put $\mathrm{ADFA}^p = \cup_{n\geqslant 1} \mathrm{ADFA}_n^p$.

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

20.2. *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: for some polynomial p, ADFA^p is not efficiently polynomially learnable (using any hypothesis class).

20.2 *Alleged consequences.*

What follows? NOWAK et al. ('Aspects') make two claims. First, the non-PAC-learnability of the *regular* languages entails that we must restrict hypothesis space. (This much, nobody disputes.) Second, the appropriate restriction yields UG.

> 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.

Here, UG is glossed as a 'collection of grammars', in the same way as e.g. the context-free languages yield a class of grammars.

Some authors do not even indicate precisely which learning-theoretic results they have in mind when making structurally similar arguments. YANG ('The Great Number Crunch': 223) write that '[o]verwhelmingly, learnability results in ·· the PAC framework[] are negative', and infer that 'learning is not possible unless the hypothesis space is tightly constrained by prior knowledge, which can be broadly identified as Universal Grammar'. *Chez* Yang, 'universal grammar' is associated with 'grammatical theories that postulate a finite range of variation, such as the Principles and Parameter framework, Optimality Theory, and others'; what remains is then the empirical question of how exactly we might decide between these proposals on empirical grounds. KODNER et al. (*Linguistics will thrive:* 5) similarly argue in favour of a 'rich' UG, on the grounds that learning theory shows that 'it is not possible to learn all computable concepts from arbitrary data presentations representative of them, with feasible amounts of computational resources'. At any rate, I propose to examine the prospects of an argument from the cryptographic hardness of learning the regular languages.

20.3 *The existing debate.*

The literature on the interpretation of these results has, I think, identified three principal problems with these arguments, other than those I outline in Chapter III.

1. **Negative results don't force universal grammar.** One style of argument runs thus (*ibid.:* § 2). Learning theory 'clearly and firmly support[s]' the poverty of the stimulus argument (here we can take this to include the PAC-unlearnability of the regular languages, though that is a slight abuse of the term 'stimulus', since the hardness assumption is given by computational rather than sample complexity.) 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]'. Learning from a plausibly sized sample requires that we should constrain the hypothesis space. Therefore, if 'small' language models ever are successfully created, they too will encode 'non-trivial structural priors facilitating language acquisition and processing'. These 'biases, principles, and limitations' are 'some form of Universal Grammar'.

All these claims can be disputed, but let us examine the last. It seems to me that it is entirely erroneous on any reasonable view of what UG is. We cannot, for instance, identify anything comparable to merge, or the elements of a principles and parameters framework for language. If we were to discover that language models could only be successfully trained on corpora of a certain size with a certain ratio of novels, blogs and newspaper articles, that would certainly be interesting, but it would not remotely resemble the kind of enquiry found in generative linguistics. In any case, it is not even responsive to the immediate target of the argument, namely, PIANTADOSI ('Language models'). He centrlaly claims that language models are perfectly good theories of language, and should substitute Chomskyan theories about grammar, competence and performance, and so on; he does so by cataloguing 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). PIANTADOSI (*ibid.:* 38 ff) claims that 'their underlying architecture for learning is relatively unconstrained'. This is not to say that there is no difference, for instance, between transformers and other architectures; but the fact that transformers have generally performed better than many other architectures hardly amounts to a 'rich' universal grammar.

2. **Positive and negative results do not straightforwardly follow the nativist–empiricist divide.** If positive learnability results were generally only to be derivable for nativist theories of language, that could (perhaps) be relatively convincing evidence for the nativist (modulo the other issues mentioned). But it is not, in fact, clear that nativists have much of an asymptotic advantage over empiricists. A common view of the supposedly advantages of nativist theories is expressed by YANG ('The Great Number Crunch': 223), who writes that

> if there is a finite number of hypotheses, then learnability is in principle ensured.

There are a number of technical problems with this claim. First, it is not true so far as PAC-learnability is concerned. Second, there is a distinction between the space of *hypotheses* and the space of processes that generate the underlying distribution over which learning is to be achieved. PAC-learnability is a stronger constraint on the second than the first. In particular, the so-called 'VC dimension' of the *hypothesis space* can be infinite even if the underlying learnable class must have finite VC dimension. Third, there is some evidence that UG-based models of language do not ensure learnability with respect to the *computational* (time) complexity of learning, even if they ensure that sample complexity is reasonable (CLARK and LAPPIN, *Nativism*: § 10.1).

3. **Distribution-freedom can be overcome.** Most hardness results assume distribution-freedom. The usual justification for this is that we have few, if any, grounds to offer a more restricted characterisation of the evidence. However, CLARK and LAPPIN (*ibid.:* cap 6) claim to offer a plausible model of language that accounts for 'indirect negative evidence' (that is, evidence of ungrammaticality through low probability) that is a relatively plausible account of natural language.

20.4 *A critical view of the use of asymptotic analysis.*

I don't propose to settle any of these the debates above. What is important, however, is that they generally involve acceptance of the Cobham–Edmonds thesis. If we were to be sceptical of the Cobham–Edmonds thesis, the dialectical pressure that hardness results would place on us to choose between theoretical *desiderata* would be immensely lesssened, and so the salience of many of these considerations would accordingly diminish. That, I submit, is the case. Most of the criticisms stated in Chapter III are not addressed in this literature.

One interesting response, however, is given by Clark and Lappin (*ibid.:* § 7.4). They note the parallel with SAT, and the risk of 'completely unsatisfactor[ily]...saying that we have algorithms that work when they work, and don't otherwise'. Their first response is to advocate the study of *parametrised* complexity, which I also partly address in Chapter III. But this is mostly a promissory note. Their second response is to note the learnability of 'various subclasses of regular, context-free and context-sensitive languages' by 'applying distributional criteria to identify congruence classes in the dataset'. The problem, however, is that these classes remain largely linguistically uninteresting, and so it is only conjectured that 'we will eventually be able to devise distributionally driven algorithms that can provably learn classes which include the set of natural languages'.

20.5 *Recapitulation.*

We have seen that PAC-learnability is neither obviously justifiable in terms of the plausiblity of its assumptions nor empirically fruitful. There is a general paucity of *positive* PAC-learnability results that correspond to clearly articulated restrictions on the hypothesis space that correspond in turn to natural language. It seems that we are indeed reduced to 'saying that we have algorithms that work when they work, and don't othewrise'. Clark and Lappin do present some positive (slightly relaxed) 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. If we think of PAC-learnability and its various polynomially bounded relaxations 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. A rather negative prognosis for the asymptotics of language acquisition follows.

VI Prolegomena to a future asymptotics of language.

21 Recapitulation.

I have offered the following general criticisms of asymptotic analysis as a standard of tractability useful in philosophy.

§ 13 Polynomial constraints are misleading when assessing tractability in human cognition—and this view is not only perfectly consistent with computational naturalism but arguably motivated by it.

§ 14 A focus on the worst case is also misleading in asymptotic analysis.

This yields the outline of a different programme of tractability-based argument in philosophy, which pays heed *inter alia* to the specific details of models of computation, the possibilities of parametrised complexity theory, and the inconclusiveness of seeming failures to make any progress in parametrised complexity theory in deciding whether problems might in practice be tractable. But this will require much mathematical progress. I therefore wish to conclude with another suggestion, which requires much less new mathematics, and instead might allow us to make progress mostly experimentally.

22 Distributional asymptotic adequacy.

We can think of tractability as a specific form of empirical adequacy. A theory according to which someone walked a distance they could only feasibly have travelled by taking a car is an empirically inadequate theory; from such resource constraints, we can generally probe theories for empirical adequacy. The various forms of tractability offered (actual tractability, observable tractability and so on) attempt to secure a standard of empirical adequacy that is both theoretically informative in the sense that it is nontrivial and rules out many live theoretical candidates, and not too strong, in the sense that it does not rule out true theories. Here I sketch one standard of empirical adequacy that I suggest may be more fruitful. This is an essentially speculative proposal, but I shall sketch how it might find empirical support.

We shall make the following suppositions.

1. We are theorising about the use of a particular resource to solve a particular problem (e.g. the time taken in working out the meaning of an expression).
2. It is possible to *quantify* the use of this resource, as we standardly have done.
3. We are interested in the relationship between resource usage and a particular parameter (let us say it is the size of an expression for concreteness.)er of instances of the problem, and can observe directly resource usage and the value of the parameter. (Perhaps with a brain scan we can work out how long it takes to establish the meaning of an expression by monitoring neurological activity.) Consider the following cases.
4. There seems to be a linear, logarithmic or quadratic relationship between resource usage and the parameter.
5. Resource usage very rapidly climbs, plausibly exponentially, as the parameter increases.

Both of these, I think, are suggestive of an *asymptotic constraint* on theories as they apply to these actual cases. Take the first case. Suppose we are given two theories T_1 and T_2. T_1 suggests $O(n)$ resource usage as a function of the parameter's value n. T_2 suggests $O(2^n)$ resource usage. It seems that T_1 is preferable. Similarly, consider the second case, in which resource usage very rapidly climbs; in this case, T_2 is preferable. (Of course, this is all a matter of somewhat approximate curve-fitting, but that hardly distinguishes this task from other areas of natural science.)

The generalisation of these considerations leads to the following standard: *distributional asymptotic adequacy*. We can hold theories to a (defeasible) constraint that the asymptotic predictions they make about actual cases' resource usage should match the observed distribution. (We can then extend this standard counterfactually, as in § 14.5.)

The very rough intuition is that superpolynomial complexity should, in general, lead to relatively sharp dropoffs in performance,[26] whereas polynomial complexity—at least of the typical, low-degree kind—permits 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 poten-

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

tially difficult cases in a reasonable period of time, accuracy, and so on.

23 A speculative application to language.

Some very long sentences are easy to parse and understand.

(5) Cows aren't green or red or brown or politicians or plumbers…

Other long sentences are more difficult to parse and understand, such as (3).

Note, for instance, that (5) can be extended to a fairly great extent without loss of comprehensibility, whereas, even if (3) is intelligible, the following—obtained from (3) by adding two further centre-embeddings—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 (5) may eventually decline as the sentence is extended. But centre-embeddings like (3) seem to decline in comprehensibility very quickly.

There seems here to be a rapid shift in the comprehensibility of centre-embeddings somewhere between depth two and four (compare (3) with (6)). On the other hand, there is no such rapid shift in the comprehensibility of sentences like (5).

What happens in cases like (3) 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. These measures trade off against one another, and asymptotic analysis can accordingly be used to chart predicted *tradeoffs between resource usage and outcomes*—how quickly accuracy or fidelity to literal meaning must be surrendered as resources cease to suffice—and to hold theories to the tradeoff curves they predict.

There is an 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 20 °C to 25 °C does not produce any *dramatic* change. On the other hand, water takes up much more space when it is 101 °C than when it is 99 °C, 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. It is stated, in the first instance, for a single cognizer confronting instances of varying size:

1. wide range in the sizes of successful cases (e.g. in the length or embedding depth of the sentences understood),
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. There is also an across-people corollary: if different people succeed on very different sizes (e.g. vocabularies of ten thousand and of a hundred thousand words), difficulty grows superpolynomially in size, and resources vary little from person to person, the same contradiction follows—though across-people comparisons carry a caveat, since they confound differences in difficulty with individual differences in resources. The illustration that follows is of this corollary form. 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 as long as the previous one, uniformly across 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 suppose that speakers make full use of the learning time available to them, then, on this doubling hypothesis, speakers with a vocabulary of a hundred thousand words must have had astronomically more learning time available than those with ten thousand—which is absurd. 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 superpolynomial 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 predicts, for asymptotic reasons, a much narrower distribution of outcomes against resources used than is actually observed. If I am right, tractability is only half the story. In a slogan: we should be concerned not only by what is too *hard*, but also by what is too *easy*.

References

Aaronson, Scott, ***P ?= 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.

Abe, Naoki, **'Feasible learnability of formal grammars and the theory of natural language acquisition'**, *Proceedings of the 12th conference on Computational linguistics - Volume 1*, COLING '88, USA: Association for Computational Linguistics, 22 August 1988, pp 1–6.

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.

Anderson, Neal G and Gualtiero Piccinini, ***The Physical Signature of Computation****: The Physical Signature of Computation: A Robust Mapping Account*, Oxford University Press, 1 June 2024.

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.

Bertolo, Stefano, **'A brief overview of learnability'**, *Language Acquisition and Learnability*, edited by Bertolo, Stefano, Cambridge: Cambridge University Press, 2001, pp 1–14.

Blaubergs, Maija S. and Martin D. Braine, **'Short-term memory limitations on decoding self-embedded sentences'**, *Journal of Experimental Psychology* 102.4 (1974), pp 745–748.

Borges, Jorge Luis, **'La Biblioteca de Babel'**, *Ficciones*, Madrid: Alianza Editorial, 1987, pp 89–100.

Chalmers, David J., **'The two-dimensional argument against materialism'**, *The Character of Consciousness*, edited by Chalmers, David J., Oxford University Press, 7 October 2010, p 0.

Chater, Nick and Mike Oaksford, ‘**Rational analysis**’: ‘The Rational Analysis Of Mind And Behavior’, *Synthese* 122.1 (1 February 2000), pp 93–131.

Chater, Nick and Paul Vitányi, ‘‘**Ideal learning’ of natural language**’: ‘‘Ideal learning’ of natural language: Positive results about learning from positive evidence’, *Journal of Mathematical Psychology* 51.3 (1 June 2007), pp 135–163.

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.

Chin, Andrew, ‘**Abstraction and equivalence**’: ‘On Abstraction and Equivalence in Software Patent Doctrine: A Response to Bessen, Meurer and Klemens’, *Journal of Intellectual Property Law* 16.2 (8 September 2016), p 197.

Chomsky, Noam, ***Aspects****: Aspects of the Theory of Syntax*, 50th edition, The MIT Press, 1965, JSTOR: j.ctt17kk81z.

Chomsky, Noam, ***New Horizons in the Study of Language and Mind***, 1st edition, Cambridge University Press, 13 April 2000.

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.

Clark, Alexander and Shalom Lappin, ‘**Acquisition**’: ‘Complexity in Language Acquisition’, *Topics in Cognitive Science* 5.1 (2013), pp 89–110.

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.

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: Oxford University Press, 27 September 2001, pp 3–16.

Dean, Walter, **'Complexity'**: 'Computational Complexity Theory', *The Stanford Encyclopedia of Philosophy*, edited by 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.

Du, Dingzhu and Ker-I. Ko, ***Theory of computational complexity***, Second edition, Hoboken, New Jersey: John Wiley & Sons, 2014, 1.

Dupre, Gabe, **'Innateness and Language'**, *The Stanford Encyclopedia of Philosophy*, edited by Zalta, Edward N. and Uri Nodelman, Fall 2026, Metaphysics Research Lab, Stanford University, 2026.

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.

Foss, Donald J. and Helen S. Cairns, **'Some effects of memory limitation upon sentence comprehension and recall'**, *Journal of Verbal Learning and Verbal Behavior* 9.5 (1 October 1970), pp 541–547.

Freund, Yoav and Robert E. Schapire, **'A desicion-theoretic generalization of on-line learning and an application to boosting'**, *Computational Learning Theory*, edited by Vitányi, Paul, Berlin, Heidelberg: Springer, 1995, pp 23–37.

Frigg, Roman and Stephan Hartmann, **'Models in Science'**, *The Stanford Encyclopedia of Philosophy*, edited by Zalta, Edward N. and Uri Nodelman, Winter 2025, Metaphysics Research Lab, Stanford University, 2025.

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.

Gold, E Mark, **'Identification'**: 'Language identification in the limit', *Information and Control* 10.5 (1967), pp 447–474.

Goldreich, Oded, **'Average Case Complexity, Revisited'**, *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, edited by Goldreich, Oded, volume 6650, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp 422–450.

Goldreich, Oded, ***Computational Complexity***: *Computational Complexity: A Conceptual Perspective*, Cambridge: Cambridge University Press, 2008.

Gödel, Kurt, **'Formally undecidable propositions'**: 'On formally undecidable propositions of *Principia mathematica* and related systems, I', *From Frege to Gödel: A Source Book in Mathematical Knowledge, 1879–1931*, translated by Bauer-Mengelberg, Stefan, reprint, Source books in the history of the sciences, Cambridge (Mass.) London: Harvard University Press, 1967, pp 596–617.

Hansen, Thomas Dueholm, Haim Kaplan, Or Zamir, and Uri Zwick, **'SAT'**: 'Faster k-SAT algorithms using biased-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).

Heinz, Jeffrey, **'Learning Long-Distance Phonotactics'**, *Linguistic Inquiry* 41.4 (2010), pp 623–661, JSTOR: 40926398.

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 J. and Umesh Vazirani, ***An Introduction to Computational Learning Theory***, The MIT Press, 15 August 1994.

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 Century: 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.

Levesque, Hector J., '**Complexity**': 'Logic and the Complexity of Reasoning', *Journal of Philosophical Logic* 17.4 (1988), pp 355–389, JSTOR: 30227050.

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.

Lozano, Aurélie C., Sanjeev R. Kulkarni, and Robert E. Schapire, **'Convergence and Consistency of Regularized Boosting With Weakly Dependent Observations'**, *IEEE Transactions on Information Theory* 60.1 (January 2014), pp 651–660.

Machine Learning Street Talk, ***The Ghost in the Machine —Noam Chomsky***, 10 July 2022.

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.

Mansour, Yishay, **'An O(nlog log n) learning algorithm for DNF under the uniform distribution'**, *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, New York, NY, USA: Association for Computing Machinery, 1 July 1992, pp 53–61.

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.

McConnell, Terrance, **'Moral Dilemmas'**, *The Stanford Encyclopedia of Philosophy*, edited by Zalta, Edward N. and Uri Nodelman, Spring 2024, Metaphysics Research Lab, Stanford University, 2024.

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, Statistical, Nonlinear, and Soft Matter Physics* 80 (2 Pt 1 August 2009), p 021912, pmid: 19792156.

Miller, George A. and Noam Chomsky, **'Finitary Models of Language Users'**, *Handbook of Mathematical Psychology*, edited by Luce, D., John Wiley & Sons., 1963, pp 2–419.

Miller, George A. and Stephen Isard, **'Free recall of self-embedded english sentences'**, *Information and Control* 7.3 (1 September 1964), pp 292–303.

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).

Norvig, Peter, **'On Chomsky and the Two Cultures of Statistical Learning'**, *Berechenbarkeit der Welt? Philosophie und Wissenschaft im Zeitalter von Big Data*, edited by Pietsch, Wolfgang, Jörg Wernecke, and Maximilian Ott, Wiesbaden: Springer Fachmedien, 2017, pp 61–83.

Nowak, Martin A., Natalia L. Komarova, and Partha Niyogi, '**Evolution**': 'Evolution of Universal Grammar', *Science* 291.5501 (5 January 2001), pp 114–118.

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 Żegleń, 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.

Papineau, David, **'Naturalism'**, *The Stanford Encyclopedia of Philosophy*, edited by Zalta, Edward N. and Uri Nodelman, Fall 2023, Metaphysics Research Lab, Stanford University, 2023.

Piantadosi, S.T., '**Language models**': 'Modern language models refute Chomsky's approach to language', *From fieldwork 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.

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.

Priest, Graham, **'The Logic of Paradox'**, *Journal of Philosophical Logic* 8.1 (1979), pp 219–241, jstor: 30227165.

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.

RESCORLA, Michael, **‘The Computational Theory of Mind’**, *The Stanford Encyclopedia of Philosophy*, edited by ZALTA, Edward N. and Uri NODELMAN, Summer 2026, Metaphysics Research Lab, Stanford University, 2026.

RISTAD, Eric Sven, ***Game****: The language complexity game*, Artificial intelligence series, Cambridge (MA), London: MIT Press, 1993.

ROBERTS, Ian, **‘Introduction’**, *The Oxford Handbook of Universal Grammar*, edited by ROBERTS, Ian, Oxford University Press, 22 December 2016, p 0.

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, Johan KWISTHOUT, and Todd WAREHAM, ‘**‘As if’-explanations**’: ‘Rational analysis, intractability, and the prospects of ‘as if’-explanations’, *Synthese* 195.2 (2018), pp 491–510, JSTOR: 26748857.

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.

Scholz, Barbara C., Francis Jeffry Pelletier, Geoffrey K. Pullum, and Ryan Nefdt, **'Philosophy of Linguistics'**, *The Stanford Encyclopedia of Philosophy*, edited by Zalta, Edward N. and Uri Nodelman, Summer 2025, Metaphysics Research Lab, Stanford University, 2025.

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.

Stoljar, Daniel, **'Physicalism'**, *The Stanford Encyclopedia of Philosophy*, edited by Zalta, Edward N. and Uri Nodelman, Spring 2024, Metaphysics Research Lab, Stanford University, 2024.

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.

Valiant, Leslie, ***Probably approximately correct****: Probably approximately correct: nature's algorithms for learning and prospering in a complex world*, New York: Basic Books, 2013, 195.

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.

Whiting, Daniel, ***Conceptual Role Semantics***, *Internet Encyclopedia of Philosophy*, no date.

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 niaf011, PMID: 40342554.

WILLIAMS, R. Ryan, **'Some Estimated Likelihoods for Computational Complexity'**, *Computing and Software Science: State of the Art and Perspectives*, edited by STEFFEN, Bernhard and Gerhard WOEGINGER, Cham: Springer International Publishing, 2019, pp 9–26.

WILLIAMSON, Timothy, **'Modal science'**, *Canadian Journal of Philosophy* 46.4–5 (2 September 2016), pp 453–492.

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: Chomksyan 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.

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.

ZACH, Richard, **'Hilbert's Program'**, *The Stanford Encyclopedia of Philosophy*, edited by ZALTA, Edward N. and Uri NODELMAN, Winter 2023, Metaphysics Research Lab, Stanford University, 2023.

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.