Tractability and the computational mind

Szymanik, Jakub; Verbrugge, Rineke

Published in:
The Routledge Handbook of the Computational Mind

Tractability and the computational mind

Jakub Szymanik Rineke Verbrugge

January 27, 2018

1 Introduction

Computational complexity theory, or in other words, the theory of tractability and intractability, is defined in terms of limit behavior. A typical question of computational complexity theory is of the form: *As the size of the input increases, how do the running time and memory requirements of the algorithm change?* Therefore, computational complexity theory, among other things, investigates the scalability of computational problems and algorithms, i.e., it measures the rate of increase in computational resources required as a problem grows (see, e.g., Arora and Barak, 2009). The implicit assumption here is that the size of the problem is unbounded. For example, models can be of any finite size, formulas can contain any number of distinct variables, and so on.[1]

Example 1: Satisfiability The problem is to decide whether a given propositional formula is not a contradiction. Let φ be a propositional formula with $p_1, \ldots, p_n$ distinct variables. Let us use the well-known algorithm based on truth-tables to decide whether φ has a satisfying valuation. How big is the truth-table for φ? The formula has n distinct variables occurring in it and therefore the truth-table has 2^n rows. If $n = 10$, there are 1,024 rows; for $n = 20$, there are already 1,048,576 rows, and so on. In the worst case, to decide whether φ is satisfiable, we have to check all rows. Hence, in

[1] Size of an input formula is often defined as its length in terms of number of symbols and size of a model is often defined as the number of elements of its universe. However, different contexts may require different definitions of the sizes involved.

such a case, the time needed to find a solution is exponential with respect to the number of different propositional letters of the formula. A seminal result of computational complexity theory states that this is not a property of the truth-table method but of the inherent complexity of the satisfiability problem (Cook, 1971).

It is often claimed that, because complexity theory deals only with relative computational difficulty of problems (the question how the running time and memory requirements of problems increase *relative* to increasing input size), it does not have much to offer for cognitive and philosophical considerations. Simply put, the inputs we are dealing with in our everyday life are not of arbitrary size. In typical situations they are even relatively small, for example, a student in a typical logic exam may be asked to check whether a certain formula of length 18 is satisfiable. In fact, critics claim, computational complexity theory does not say how difficult it is to solve a given problem for a fixed size of the input. We disagree.

Even though in typical cognitive situations we are dealing with reasonably small inputs, we have no bounds on their size. Potentially, the inputs can grow without limit. Therefore, complexity theory's idealized assumption of unbounded input size is, like many idealized assumptions in the sciences, such as point-masses in Newtonian physics, both necessary in that it simplifies analysis of a complex world and convenient, because it results in characterizations of phenomena that balance simplicity of descriptions with empirical adequacy. As such, this idealization is justified on practical analytical grounds, whether it is true or not. Moreover, considering any input size, we avoid making arbitrary assumptions on what counts as a natural instance of the cognitive task. For instance, such a general approach works very well for syntax, where the assumption that sentences in a natural language can be of any length has led to to the development of generative grammar and mathematical linguistics (Chomsky, 1957).

Even though in each case of cognition we are dealing with inputs of a specific size, which in practice is limited, imposing *a priori* bounds on the input size would conflict with the generative nature of human cognition, that is, the idea that human minds can *in principle* yield cognitive outcomes such as percepts, beliefs, judgments, decisions, plans, and actions for an infinite number of situations. Moreover, in experimental practice, subjects usually do not realize the size of the model's universe in which they are supposed to solve some given problem, therefore, it often seems that they develop strategies (algorithms) for all possible sizes of universes. Still, we realize that many skeptics might not be satisfied with the above arguments. Therefore, in this chapter, after a brief introduction to computational complexity theory, we offer an overview of some interesting empirical evidence corroborating computational and logical predictions inspired by tractability considerations.

2 Theory of Tractability

Philosophers are still discussing the precise meaning of the term 'computation' (see e.g. Moschovakis, 2001); see also the chapter by Murray Shanahan on "Artificial Intelligence" in this volume. Intuitively, 'computation is a mechanical procedure that calculates something'. Formally speaking, the most widespread approach to capture computations is to define them in terms of Turing machines or an equivalent model. The widely believed Church-Turing Thesis states that *everything that ever might be mechanically calculated can be computed by a Turing machine* (see, e.g., Cooper, 2003, for an introduction to computability theory). It has been quickly observed not only that there exist uncomputable problems but even that for some computable problems, effective algorithms appear not to exist, in the sense that some computable problems need too much of computational resources, like time or memory, to get an answer. As explained in the introduction, computational complexity theory investigates the amount of resources required and the inherent difficulty of computational problems. Practically speaking, it categorizes computational problems via complexity classes (see, e.g., Arora and Barak, 2009, for an introduction to complexity theory).

This chapter focuses on the distinction between tractable and intractable problems. The standard way to characterize this distinction is by reference to the number of steps a minimal Turing machine needs to use to solve a problem with respect to the problem size. Informally, problems that can be computed in polynomial time are called tractable; the class of problems of this type is called PTIME. The class of intractable problems, including those problems that require exponential time to be solved on a deterministic Turing machine, are referred to as NP-hard problems.[2] Intuitively, a problem is NP-hard if there is no 'clever' algorithm for solving it, so NP-hard problems lead to combinatorial explosion.[3].

More formally, PTIME (or P) is the class of problems computable by deterministic Turing machines in polynomial time. NP-hard problems are problems that are at least as difficult as any of the problems belonging to the class of problems that can be computed by nondeterministic Turing machines in polynomial time (NPTIME problems). NP-complete problems are NP-hard problems belonging to NPTIME, hence they are the most difficult problems among the NPTIME problems. It is known that P=NP if any NPTIME-complete problem is PTIME computable. However, whether P=NP is a famous open question. Almost all computer scientists believe that the answer is negative (Gasarch, 2012). Many natural problems are computable in PTIME, for instance, calculating the greatest common divisor of two numbers or looking something up in a dictionary. We already encountered the NP-hard problem of satisfiability in Example 1.

Using the two fundamental complexity classes PTIME and NPTIME Edmonds (1965) and Cobham (1965) proposed that *the class of tractable problems is identical to the PTIME class.* The thesis is accepted by most computer scientists (see, e.g., Garey and Johnson, 1990). The version of the Cobham-Edmonds Thesis for cognitive science is known as the P-Cognition Thesis (Frixione, 2001): *human cognitive capacities are constrained by polynomial time computability.*

However, Ristad (1993) and Szymanik (2010) have shown that natural language contains constructions of which the semantics is essentially NP-complete and Levesque (1988) has recognized the computational complexity of general logical reasoning problems like satisfiability to be NP-complete. In other cognitive domains, Tsotsos (1990) has shown that visual search

[2]In the context of this chapter we assume, like many computer scientists, that the open question whether P=NP has a negative answer; see the next paragraph for a precise explanation.

[3]An example of combinatorial explosion happens when you try all valuations, that is, all combinations of truth values of the relevant propositional atoms, to find out whether a given formula is satisfiable (see Example 1).

is NP-complete. Therefore, the P-Cognition Thesis seems to be clearly too broad, excluding some non-polynomial problems with which the human mind can still deal. This does not need to mean that we should disregard these theories. Possibly the complexity of these theories may be just a result of over-specification, including certain instances of the problem that should be excluded from the realm of human cognitive capacity. To account for this problem, van Rooij (2008) proposed an alternative to the P-Cognition Thesis, the so-called the Fixed-parameter Tractability Thesis. The idea here is to not only consider the size of the input but also other input parameters; then, intractability of some problems may come from a parameter which is in practice usually very small no matter the size of the whole input. Such problems should still be tractable for the human mind (see, e.g., Downey and Fellows, 1998, for an introduction to parameterized complexity theory).[4] In other words, if inputs have a certain structure, the problem becomes easy to solve.

Example 2 In the Traveling Salesman Problem, the input is a set of cities, distances between them, a point of departure, and a maximum route length. The computational problem is to decide whether there exists a route that starts and ends in the point of departure, visits all cities exactly once, and does not exceed the maximum route length. This problem is also known to be NP-complete, thus intractable (Karp, 1972). However, when the cities are aligned exactly on a circle, then the problem is obviously trivial to solve. Actually, it is even enough that relatively many cities are aligned on a circle to make the problem easy. This property can be parameterized, intuitively, by counting the number of cities inside the circle and the number of cities on the circle. The Traveling Salesman Problem is fixed-parameter tractable (Deıneko et al., 2006), meaning that all the intractability of the problem can be contained within the parameter, that is, the more cities are not located

[4]Thus, an NP-hard problem may require time polynomial in the size of the input but exponential in a fixed parameter. For example, the satisfiability problem of Example 1 can be parameterized by the number of different atomic variables. A formula of size n with k atomic variables can be evaluated by checking all relevant valuations, which takes time of the order $n \times 2^k$. So for formulas that contain only three atomic variables, it appears to be tractable for people to check the satisfiability of quite long formulas.

on the circle, the more intractable the problem becomes.

2.1 Complexity distinctions and logical descriptions

Many complexity distinctions can also be captured in descriptive terms via logic. The idea here is simple: We use logical languages to describe a computational problem in some minimal way. The simpler the description, the easier the problem. In other words, descriptive complexity deals with the relationship between logical definability and computational complexity. We replace the classic computational complexity question with the descriptive complexity question, i.e., the question: *How difficult is it to describe the problem using some logic* (see, e.g. Immerman, 1998)? In terms of our two basic computational classes PTIME and NPTIME, it is important to note that every problem that can be described by first-order formula is computable in polynomial time. Moreover, the seminal result of descriptive complexity theory states that: a problem is definable in the existential fragment of second-order logic if and only if it belongs to NP (Fagin, 1974).

Summing up, the idea behind capturing complexity of cognitive problems boils down to having a formal, in some sense minimal, computational or logical description of the cognitive problem. The hope is that this description captures inherent, intrinsic, combinatorial properties of the problem that are independent from particular descriptions. Such structural properties should correlate with human behavior; we will give some examples in the following sections. Next, we study complexity of such descriptions or models that should then also capture inherent cognitive complexity and lead to some (experimental) predictions or explanatory insights about the cognitive capacity in question.

In the next few sections we give a personal selection of the outlined approach to study different aspects of cognition, namely Boolean categorization, semantic processing, and social reasoning. We will show several cases in which the P-Cognition Thesis fails but the Fixed-parameter Tractability Thesis comes to the rescue, while in one case of social reasoning, apparent levels of cognitive difficulty do not line up with levels of computational (standard or parameterized) complexity (see also van Rooij, 2008; Aaronson, 2013; Isaac et al., 2014, for more examples and discussion of complexity and the computational mind).

3 Boolean Categorization

From infancy onwards, we learn to categorize objects, for example, into animate and inanimate, and we continue to learn new categories well into adulthood, for example, distinguishing between ale and lager beers. In particular, many of our new concepts are built from the old ones using Boolean relations, such as 'and', 'or', and 'not'. For instance, we know that 'cousin' is a child of an uncle *or* aunt; 'beer' is an alcoholic beverage usually made from malted cereal grain *and* flavored with hops, *and* brewed by slow fermentation; in basketball, 'travel' is illegally moving the pivot foot *or* taking three or more steps without dribbling; 'depression' is a mood disorder characterized by persistent sadness *and* anxiety, *or* a feeling of hopelessness *and* pessimism, *or* .... Therefore, an important question in psychology is how people acquire concepts. In order to approach that big question one may try to consider the cognitive complexity of the process of acquisition, and ask: *Why are some concepts harder to acquire than others*?

Since the beginning of the experimental studies on categorization, it has been clear that some new concepts are easier to learn than others. For instance, intuitively, concepts depending on *and* are easier to learn than those depending on *or* (Bruner et al., 1956).[5] Shepard et al. (1961) have run a number of various experiments to study the acquisition of six different sorts of concepts based on three binary variables. Each concept was defined in a way to contain exactly four instances and four non-instances. In this way, the researchers created six novel concepts or categories. The experimental results showed a very robust complexity trend, which we describe below. Trying to explain Shepard and colleagues' finding, Feldman (2000) proposed to define their six concepts using Boolean propositional logic:

Concept I contains objects satisfying the following description: (not-a b c) or (not-a b not-c) or (not-a not-b c) or (not-a not-b not-c).

[5] One reason corresponds to the fact that $p \wedge q$ is true in just one of the four lines in a truth table while $p \vee q$ is true in three of them.

Concept II consists of the following instances: (a b c) or (a b not-c) or (not-a not-b c) or (not-a not-b not-c).

Concept III consists of the following instances: (a not-b c) or (not-a b not-c) or (not-a not-b c) or (not-a not-b not-c).

Concept IV consists of the following instances: (a not-b not-c) or (not-a b not-c) or (not-a not-b c) or (not-a not-b not-c).

Concept V consists of the following instances: (a b c) or (not-a b not-c) or (not-a not-b c) or (not-a not-b not-c).

Concept VI consists of the following instances: (a b not-c) or (a not-b c) or (not-a b c) or (not-a not-b not-c).

Shepard's complexity trend is: I $<$ II $<$ III, IV, V $<$ VI. This means that subjects made the least errors (and took the least time) before they learnt concepts of type I, while they made the most errors (and took the longest time) before they learnt concepts of type VI.

Feldman's main insight was to propose that the Boolean complexity of a concept can capture its cognitive difficulty in terms of errors made and time needed. Boolean complexity of the concept was defined as the length (expressed in terms of the number of literals, i.e., positive or negative variables) of the shortest Boolean formula logically equivalent to the concept.[6] For an easy example, the reader can check that the formula (a and b) or (a and not b) or (not a and b) can be succinctly and equivalently represented as (a or b), hence, its Boolean complexity is 2. Computing the shortest descriptions for the six concepts we get:

I := not a (complexity 1);

II := (a and b) or (not a and not b) (4);

III := (not a and not c) or (not b and c) (4);

IV := (not c or (not a and not b)) and (not a or not b) (5),

V := (not a and not (b and c)) or (a and (b and c)) (6);

[6] Interestingly, finding the shortest formula is intractable.

VI := (a and ((not b and c) or (b and not c))) or (not a and ((not b and not c) or (b and c))) (10).

Therefore, a prediction of cognitive difficulty based on Boolean complexity is consistent with Shepard's trend.

Based on this insight, Feldman also conducted experiments to provide a larger data set. He considered a generalization of the Shepard setting, an arbitrary Boolean concept defined by P positive examples over D binary features. For Shepard's six types, D was 3 and P was 4. Feldman (2000) experimentally studied 76 new Boolean concepts in a range of $D \in \{3, 4\}$ and $2 \leq P \leq 4$. Boolean complexity accounts for 50% of variance in the extended dataset. One can therefore conclude that minimal description length predicts the cognitive difficulty of learning a concept.

Feldman's paper does not deal with the border of tractability and intractability directly, however, it uses a logical description and the prototypical complexity problem of satisfiability to successfully capture and explain the difficulty of a very crucial cognitive problem. As such, it is a beautiful and inspiring example of applying logic in cognitive science.

Still, one may object to logical models of categorization on the grounds of the P-Cognition Thesis by asserting that a theory based on the NP-hard satisfiability problem cannot be computationally plausible. One way out here would be to invoke the strategy suggested by the Parameterized Complexity Thesis, that is, instead of rejecting the whole model, one can point out that it probably overgeneralizes the cognitive capacity. In order to come closer to the real cognitive capacity, the mental representation language can be restricted to some tractable subset of Boolean logic. Examples are restrictions to only positive formulas, only Horn clauses,[7] or only constraints on pairs of variables, possibly only in conjunctive normal form (2-CNF). Such a restriction would make the model fixed-parameter tractable (see Samer and Szeider (2009)), where the parameters would reflect the 'syntactic complexity' of the formula, corresponding to a mental representation.

The question of the underlying representation language for categorization

[7]A Horn clause is a disjunction of literals in which at most one of the disjuncts is a (positive) propositional variable.

may be asked more precisely within the use of machine learning techniques and Bayesian data analysis. Piantadosi et al. (2016), using Bayesian concept learning models, ask precisely this question: Which representational system is the most likely, given human responses? Surprisingly, their analysis suggests that intractable fragments are better representation languages than the tractable fragment in terms of Horn clauses. However, the verdict is not yet final, because the authors look at slightly different categorization problems than Feldman and they do not consider other viable tractable representations. This is good news, because clearly there are still many interesting research opportunities in explaining and predicting the complexity of human categorization (see also the Chapter by M. Colombo, "Learning and reasoning", this volume).

4 Semantic Processing

Probably the first serious applications of complexity theory beyond computer science are to be found in modern linguistics. Noam Chomsky started to view language from the computational perspective (Chomsky, 1965), proposing a computability hierarchy of various syntactic fragments of language. Chomsky's famous hierarchy of finite-state, context-free, context-sensitive, and recursive languages opened linguistics to many interactions with computer science and cognitive science. Let us remark at this point that Chomsky's approach is also asymptotic in nature. Even though the sentences we encounter are all of bounded length, he has assumed that they might be arbitrarily long. It was this assumption that directly led to the discovery of the computational model of language generation and the complexity hierarchy. This in turn was the source of the famous debate about how much computational resources are needed to describe grammars of natural languages (see, e.g., Barton et al., 1987). In other words, the computational approach provided us with methodological constraints on linguistic theories of syntax, as well as empirical predictions.

Already in the 1980s, researchers started to ask tractability questions about the grammatical formalism. Using computational complexity one could study whether a generative formalism is computationally plausible.

Computational complexity of parsing and recognition has become a major topic along with the development of computational linguistics (see, e.g., Barton et al., 1987; Pratt-Hartmann, 2008). To give a very quick summary, the results show that even for relatively simple grammatical frameworks, some problems are intractable. For example, regular and context-free languages have tractable parsing and recognition problems. Already somewhat more complex formalisms, such as Lambek grammars, Tree-Adjoining Grammars, Head-Driven Phrase Structure Grammar, and context-sensitive grammars, all give rise to intractable problems.

Interestingly, there is also a prominent descriptive complexity spin-off from the grammatical research. Instead of using machine models (such as finite state machines, push-down automata or Turing machines generating the language), one can specify grammars in terms of general constraints. A string (or a tree) is grammatical if and only if it satisfies the constraints. Naturally, such constraints can be expressed in logic, translating the problem of grammar complexity into the problem of *how much logic is needed to define the grammar*. This leads to some fundamental results: Büchi (1960) has shown that a language is definable by the so-called monadic fragment of second-order logic if and only if it is regular. McNaughton and Papert (1971) have proven that a set of strings is first-order definable if and only if it is star-free[8]. For readers who prefer modal logic to fragments of second-order logic, the following reformulation may seem more attractive: The temporal logic of strings captures star-free languages and propositional dynamic logic captures regular languages (see e.g. Moss and Tiede, 2006).

More recently, complexity thinking has been applied to some issues in natural language semantics. The earliest results come from a book by Ristad (1993). He provides a complexity analysis of various formal approaches to the problem of dependencies in discourse, such as finding the proper referents for "she" and "her" in the sentences "Yesterday Judit visited Susan for a game of chess. She beat her although she played black". Ristad's conclusion is that this problem is NP-complete and therefore all good formalisms accounting

[8]Regular *star-free* languages have Boolean operators like concatenation and union of two elements, but not the Kleene star (which allows any finite concatenation of elements of the language).

for it and correctly resolving anaphora, should be at least as complex, i.e. NP-complete, but not more complex to avoid over-generalizations. This very much resembles the argument from the Fixed-parameter Tractability Thesis, i.e., even though the general version of the problem is intractable, the hardness may come from a parameter which is usually very small in linguistic practice. However, as far as we know, to this day nobody has extended Ristad's analysis with new tools of parameterized complexity.[9]

One of the most prominent studies in semantic theory considers quantifier expressions such as 'most', 'some', or 'many'. Generalized quantifier theory has studied the descriptive complexity of quantifier meaning. It is well known, for instance, that the meaning of a quantifier like 'most' is not definable in first-order logic (even if we restrict ourselves to finite universes) (see Peters and Westerståhl, 2006, for a systematic overview). van Benthem (1986) started the work of characterizing quantifiers in terms of Chomsky's hierarchy and Szymanik (2010) draws a tractability/intractability border among natural language quantifier constructions. Interestingly, these distinctions turned out to predict cognitive difficulty of quantifier processing, as shown by Szymanik and Zajenkowski (2010) and Schlotterbeck and Bott (2013), respectively. Furthermore, computational complexity even has an impact on the distribution of lexical items in natural language, as recently observed by Szymanik and Thorne (2017) in the statistical analysis of English corpora. Obviously, there are still many open questions in this field, including both mathematical (e.g., parameterized analysis) and cognitive (e.g., modeling) questions (see Szymanik, 2016, for an overview).

Another prominent complexity problem in the semantic literature, pointed out already by Levesque (1988), is to account in computationally tractable way for the process of linguistic inference: does one given sentence follow from another given sentence?. This is important both for cognitive science considerations of how people reason and for natural language processing applications, like textual inference in search engines. The problem seems very hard, because inference in even such a simple formal system as Boolean propositional logic is already intractable. Hence, according to the

[9] Szymanik (2016) gives another defense for Ristad's thesis in terms of inferential meaning and indirect computability.

P-Cognition thesis, standard logical systems must outstrip human reasoning capacities. One approach here, known as the Natural Logic Program, is to work bottom-up, formalizing in a computationally minimal way various fragments of natural language (see, e.g., Icard III and Moss, 2014, for an overview). Building on this idea, Pratt-Hartmann (2004) has provided a complexity characterization of reasoning problems in different language fragments. For instance, the satisfiability problem of the syllogistic fragment[10] is in PTIME as opposed to the fragment containing relative clauses,[11] which is NP-complete. These results have already motivated empirical work, both in cognitive modeling (Zhai et al., 2015) and in linguistic analysis (Thorne, 2012).

5 Social Reasoning

In this section we review some interesting puzzles and games in which it is useful for people to reason about one another's knowledge, beliefs and intentions and we check the relevance of the P-Cognition Thesis and the Fixed-parameter Tractability Thesis for these tasks. The scope of possibilities turns out to be very varied.

5.1 Epistemic puzzles

The Muddy Children puzzle is a classical example of reasoning about other people's knowledge. Three children come home after playing outside. Their father says: "At least one of you has mud on your forehead." Then he asks the children: (**I**) "If you know for sure that you have mud on your forehead, please step forward". The children have common knowledge that their father never lies and that they are all sincere and perfect logical reasoners.[12] Each

[10] The syllogistic fragment of first-order logic only contains sentences corresponding to "All A's are B", "Some A's are B", "Some A's are not B" and "No A's are B", where A and B are unary predicates.

[11] In addition to the syllogistic fragment, this fragment also contains sentences such as "Every A who is not C is B".

[12] *Common knowledge* means informally that all the children know, all the children know that they all know, and so on, ad infinitum (for more explanation and a formal definition see, for example, van Ditmarsch et al. (2009)).

child can see the mud on the foreheads of the other two, but cannot see his or her own forehead. After the father asks (**I**) once, all three children remain standing in place. When their father asks the question a second time, however, all muddy children—in this case the two children b and c—respond by stepping forward, meaning that they know that they have mud on their foreheads. How could they have figured this out?

In epistemic logic, this puzzle is usually modeled using eight possible worlds, corresponding to the 2^3 different possible combinations of muddiness and cleanness of the three children a, b, c. For example, world (0,1,1) would correspond to child a being clean and children b and c being muddy. Two worlds are joined by an accessibility an edge for agent $i \in \{a, b, c\}$, if i cannot distinguish between the two worlds on the basis of his or her information. At successive states of the puzzle story, the agents' knowledge grows and more and more of these accessibility relations can be deleted, until the resulting possible worlds model only contains one world, namely the current real situation (0,1,1) with three accessibility relations to itself only, for each agent a, b, c (see for example van Ditmarsch et al. (2007); Fagin et al. (1995)).

For the above version of the muddy children puzzle with only three children, the solution based on possible worlds models as seen from the *global perspective* of the outside puzzle solver is elegant. However, it is hardly scalable, because a version of the puzzle with n children requires possible worlds models with 2^n worlds. Of course this is problematic in the context of the P-Cognition Thesis. It is intuitively implausible that actual human beings generate such exponential-sized models of all possible scenarios when they solve the muddy children puzzle for n children, because it is computationally intractable, requiring exponential time with respect to the number of agents. What is required here is a new *local perspective*, a study of reasoning about other agents' knowledge from the perspective of the agents involved. Such more local representations and epistemic logics are more cognitively plausible. Gierasimczuk and Szymanik (2011b,a), based on results from the theory of general quantifiers, propose to use local representations for several variants of the muddy children puzzle for n agents. Surprisingly, the total size of the initial model that they need is only $2n + 1$. As a result, the proposed representation is exponentially more succinct than the standard dynamic epistemic logic approach (see, for example, van Ditmarsch et al. (2007)) and their solution is tractable, bringing the puzzle into the realm of the P-Cognition Thesis. Dégremont et al. (2014) develop a general approach to local semantics for dynamic epistemic logics.

5.2 Theory of mind in story tasks

Theory of mind is the ability to reason about other people's mental states, such as their beliefs, knowledge, and intentions. Many studies appear to show that only when they are around four years of age do children learn that other people may have beliefs different than their own (but see the discussion on infants' precursors of theory of mind in M. McLeod and J. Mitchell's chapter "Computational approaches to social cognition", this volume).

Understanding that this ability can be applied recursively, such as in the second-order attribution "She doesn't know that I know that she wrote the anonymous Valentine card", starts even later, between five and seven years of age (Arslan et al., 2017a). Adults usually understand stories in which third-order theory of mind plays a role, but have difficulty answering questions at higher levels (Kinderman et al., 1998; Stiller and Dunbar, 2007). Theory of mind is often tested by asking experimental participants questions about stories starting from an initial situation, in which both the facts and the characters' first-order and second-order knowledge change a few times. The observer then needs to answer questions about a belief or knowledge statement of interest, such as "Does Ayla know that Murat knows that she moved the chocolate bar to the toy box?"

Recently, for the first time, this problem of dynamic knowledge update has been formalized on the basis of dynamic epistemic logic. Iris van de Pol et al. (2015) show that the problem, restricted to S5 models (in which the accessibility relations are reflexive, transitive and symmetric) with a salient "real world", is PSPACE complete.[13] This is a surprisingly high level of complexity, considering that P$\subseteq$ NP $\subseteq$ PSPACE and that most complexity theorists expect P and PSPACE to be different.[14] Even worse, it is not easy

[13]It turns out that this is the same complexity class that was independently found for a similar problem, namely model checking in epistemic planning (Bolander et al., 2015).

[14]The question P=?PSPACE remains a famous open problem, just like P=?NP.

to prove parameterized tractability results for the belief update problem. Only if both the maximum number of events per update *and* the number of updates remain small, so that the final updated model remains small as well, does the dynamic belief update problem become fixed-parameter tractable (van de Pol et al., 2015). Indeed it appears that in standard experimental false belief tasks, the number of actions and the number of events per action do stay small (Bolander, 2014), but that may not be the case for real-world belief update problems.

Perhaps most surprisingly, the order of theory of mind, corresponding to the modal depth of the formula that needs to be checked, does *not* play much of a role in the intractability of the dynamic belief update problem that participants in a false belief task need to solve: even for formulas of modal depth one, the problem remains intractable when the number of events per update or the number of updates grows large (van de Pol et al., 2015). This contrasts with the literature showing that, all children learn first-order false belief tasks much earlier than second-order ones, even if the number of events per update and the number of updates remain the same(Arslan et al., 2017b; Miller, 2009).

To move from first-order to second-order tasks, children have to keep separate the real situation (the chocolate in the toy box) both from Murat's belief about the chocolate and from Ayla's false belief about Murat's belief. Thus, they have to keep intermediate solutions in mind when asked a second-order question. It requires sophisticated complex memory skills to pass such a *serial processing bottleneck*. The serial processing bottleneck hypothesis (Verbrugge, 2009) is based on the finding that working memory acts as a bottleneck: people can only hold one chunk of information in working memory at a time Borst et al. (2010). This suggests that children need complex working memory strategies in order to process embedded beliefs in such a way that chunks of information can pass through the working memory bottleneck within that time threshold. It has indeed been shown that complex memory tasks (and not simple memory tasks) predict children's accuracy in second-order false belief tasks (Arslan et al., 2017a).

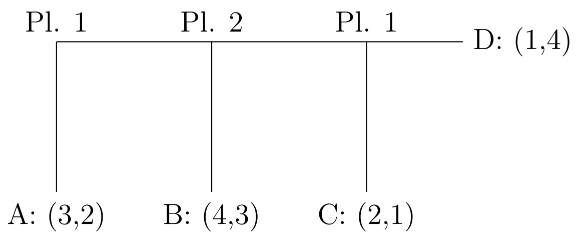

Figure 1: Game tree for a centipede-like game in which Player 1 chooses first; if she goes right, then Player 2 chooses, and if Player 2 also goes right, finally Player 1 chooses again. The pairs at leaves A, B, C, and D represent the payoffs of Player 1 and Player 2, respectively, if the game ends up there.

5.3 Strategic reasoning in dynamic games

Another surprise turns up when turning from how people apply theory of mind in false belief tasks to how they apply it in dynamic turn-taking games. It has been shown that children learn to apply second-order theory of mind in competitive dynamic games based on centipede-like game trees such as the one in Figure 1 only a few years after they have mastered second-order false belief tasks (Flobbe et al., 2008; Raijmakers et al., 2012) and even adults do not seem to be very good at such games (Hedden and Zhang, 2002). However, participants whose reasoning processes have been 'scaffolded' by an intuitive presentation of the game or by step-wise training, which progresses from simple decisions without any opponent, through games that require first-order reasoning ("the opponent plans to go right at the next trapdoor"), to games that require second-order reasoning ("the opponent thinks that I plan to go left at the last trapdoor") learn to play the game almost perfectly (Goodie et al., 2012; Meijering et al., 2011; Verbrugge et al., 2018).

Meijering et al. (2014) have constructed computational cognitive models to explain why people do not spontaneously start applying second-order theory of mind in competitive dynamic games. The model is based on the idea that theory of mind reasoning is effortful and that people first try out simpler strategies, only moving up a level of theory of mind when they lose too often, until they converge on a reasoning strategy that is "as simple as possible, as complex as necessary"[15]. Their model turns out to fit the data of both the adults and the children of the earlier experiments (Meijering et al., 2011; Verbrugge et al., 2018; Flobbe et al., 2008).

All the centipede-like turn-taking games used in the experiments are most efficiently solved using backward induction, an inductive algorithm defined on a game tree (Osborne and Rubinstein, 1994). The backward induction algorithm tells us which sequence of actions will be chosen by agents that want to maximize their own payoffs, assuming common knowledge of rationality. Szymanik (2013) showed that backward induction is PTIME-complete with respect to the size of the game tree. So contrary to expectation, solving dynamic games is tractable and does comply with the P-Cognition Thesis, whereas solving false belief tasks does not.

Even though backward induction is the most efficient strategy, an eye-tracking study suggests that it is not necessarily the strategy that participants really use in centipede-like games; they seem instead to favor a form of "forward reasoning plus backtracking" and they look a lot at the decision points, not only at the payoffs at the leaves (Meijering et al., 2012). This reasoning strategy corresponds to the adagium "first try something simple that has worked in other contexts in the past" and is akin to causal reasoning, as in: "If I choose right, then what will my opponent do at the next decision point? Oh, he will choose left, but that's bad for me, so I better go left now to prevent it." Szymanik et al. (2013); Bergwerff et al. (2014) have made computational models of the two reasoning strategies and have fit them to the participants' reaction time data, corroborating that the participants do use forward reasoning plus backtracking more than backward induction.

But which reasoning strategies do people actually use in specific games? Ghosh et al. (2014); Ghosh and Verbrugge (2018) formulated a logical strategy language that can be turned into computational cognitive models and they used these models to rule out certain reasoning strategies based on the predictions on choices and reaction times from the simulations of the models' "virtual strategizers" in several variations of dynamic games, including ones with non-rational opponents (such as Ghosh et al. (2017)). It would

[15] The authors here use 'complex' informally in the sense of cognitively difficult.

be interesting to apply these methods also to dynamic negotiation games, in which people also do not apply second-order theory of mind spontaneously, even though it leads to win-win solutions, but can be trained by a second-order theory of mind software opponent to do so (de Weerd et al., 2017), as well as to cooperative games such as the Tacit Communication Game, where theory of mind is very useful to coordinate on signals but it is not clear whether people really apply it, although at least first-order theory of mind appears to be likely (van Rooij et al., 2011; Blokpoel et al., 2012; de Weerd et al., 2015).

6 Conclusions

This book takes the computational perspective on cognition. In this chapter, we focus on the perspective of tractability: Is a specific cognitive task doable in general within a reasonable time and using reasonable memory space? A formal analysis of an information processing task that takes tractability seriously generates empirical predictions, which can be tested by experiments. Using computational cognitive models, for example, in a cognitive architecture such as SOAR or ACT-R, leads to very precise predictions, for example, about reaction times and sequences of places on the screen where participants will look at certain moments during an experiment. The outcomes of these experiments may later feed back into revisions of the formal theory and the computational cognitive model.

It turns out that the P-Cognition Thesis, stating that human cognitive capacities are constrained by polynomial time computability, provides a fruitful lens for assessing cognitive tasks. Some well-known cognitive tasks, however, appear to be NP-complete. In that case, sometimes a more relaxed thesis, the Fixed-parameter Tractability Thesis, comes to the rescue. Some NP-complete tasks turn out to be fixed-parameter tractable: The intractability comes from the problem's deterministic time complexity being exponential in some parameters that are usually very small in practice, no matter the size of the input as a whole. Such problems should still be tractable for the human mind, in contexts where the 'guilty' parameters can indeed be shown to be small. In other words, if the inputs have a helpful structure, the problem becomes tractable after all.

We have illustrated the cycle from complexity-theoretic analyses, through computational models and empirical results back to theory, by way of a number of specific examples in Boolean categorization, semantic processing in natural language, and social reasoning in story tasks, puzzles and games. Many well-known tasks indeed turn out to be solvable in PTIME or can be shown to be fixed-parameter tractable by choosing an appropriate representation. Sometimes intuitions about which tasks are easy and which tasks are hard do not perfectly match with the complexity-theoretic analysis. For those cognitive tasks, the perceived complexity for the human mind may come from the required complicated working memory strategies, in which intermediate solutions need to be stored and correctly applied. In any case, when studying the computational mind, knowledge about tractability and fixed-parameter tractability proves to be an essential element of the cognitive scientist's toolbox.

Acknowledgments The research of the first author leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. STG 716230 CoSaQ. The work of the second author on social cognition has been supported by Vici grant NWO-227-80-00.

References

Aaronson, S. (2013). Why philosophers should care about computational complexity. In *Computability: Turing, Gödel, Church, and Beyond*, pages 261–328. MIT Press.

Arora, S. and Barak, B. (2009). *Computational Complexity: A Modern Approach.* Cambridge University Press, New York, NY, USA, 1st edition.

Arslan, B., Hohenberger, A., and Verbrugge, R. (2017a). Syntactic recursion facilitates and working memory predicts recursive theory of mind. *PLoS ONE*, 12(1):e0169510.

Arslan, B., Taatgen, N., and Verbrugge, R. (2017b). Five-year-olds' systematic errors in second-order false belief tasks are due to first-order theory of mind strategy selection: A computational modeling study. *Frontiers in Psychology: Cognitive Science*, 8:275.

Barton, E. G., Berwick, R., and Ristad, E. S. (1987). *Computational Complexity and Natural Language*. Bradford Books. MIT Press.

van Benthem, J. (1986). *Essays in Logical Semantics*. Reidel.

Bergwerff, G., Meijering, B., Szymanik, J., Verbrugge, R., and Wierda, S. M. (2014). Computational and algorithmic models of strategies in turn-based games. In Bello, P. and others, editors, *Proceedings of the 36th Annual Conference of the Cognitive Science Society*, pages 1778–1783.

Blokpoel, M., van Kesteren, M., Stolk, A., Haselager, P., Toni, I., and Van Rooij, I. (2012). Recipient design in human communication: simple heuristics or perspective taking? *Frontiers in Human Neuroscience*, 6:253.

Bolander, T. (2014). Seeing is believing: Formalising false-belief tasks in dynamic epistemic logic. In *Proceedings ECSI*, pages 87–107.

Bolander, T., Jensen, M. H., Schwarzentruber, F., and Rennes, E. (2015). Complexity results in epistemic planning. In *IJCAI*, pages 2791–2797.

Borst, J. P., Taatgen, N. A., and van Rijn, H. (2010). The problem state: A cognitive bottleneck in multitasking. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 36(2):363.

Bruner, J. S., Goodnow, J. J., and Austin, G. A. (1956). *A Study of Thinking*. Science Editions, New York.

Büchi, J. (1960). Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 6:66–92.

Chomsky, N. (1957). *Syntactic Structures*. Mouton, The Hague.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press.

Cobham, A. (1965). The intrinsic computational difficulty of functions. In Bar-Hillel, Y., editor, *Proceedings of the 1964 Congress for Logic, Methodology, and the Philosophy of Science*, pages 24–30, Jerusalem. North-Holland Publishing.

Cook, S. A. (1971). The complexity of theorem-proving procedures. In *STOC '71: Proceedings of the third annual ACM symposium on Theory of Computing*, pages 151–158, New York, NY. ACM Press.

Cooper, B. S. (2003). *Computability Theory*. Math. Series. Chapman & Hall/CRC.

Dégremont, C., Kurzen, L., and Szymanik, J. (2014). Exploring the tractability border in epistemic tasks. *Synthese*, 191(3):371–408.

Deıneko, V. G., Hoffmann, M., Okamoto, Y., Woeginger, G. J., et al. (2006). The traveling salesman problem with few inner points. *Operations Research Letters*, 34(1):106–110.

van Ditmarsch, H., Eijck, J. v., and Verbrugge, R. (2009). Common knowledge and common belief. In Eijck, J. v. and Verbrugge, R., editors, *Discourses on Social Software*, pages 99 – 122. Amsterdam University Press.

van Ditmarsch, H., Van der Hoek, W., and Kooi, B. (2007). *Dynamic Epistemic Logic*. Springer, Dordrecht.

Downey, R. G. and Fellows, M. R. (1998). *Parameterized Complexity*. Monographs in Computer Science. Springer, New York, NY.

Edmonds, J. (1965). Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467.

Fagin, R. (1974). Generalized first-order spectra and polynomial time recognizable sets. In Karp, R., editor, *Complexity of Computation*, volume 7 of *SIAM—AMS Proceedings*, pages 43–73. American Mathematical Society.

Fagin, R., Halpern, J. Y., Moses, Y., and Vardi, M. Y. (1995). *Reasoning About Knowledge*. MIT Press, Cambridge, MA.

Feldman, J. (2000). Minimization of Boolean complexity in human concept learning. *Nature*, 407(6804):630–633.

Flobbe, L., Verbrugge, R., Hendriks, P., and Krämer, I. (2008). Children's application of theory of mind in reasoning and language. *Journal of Logic, Language and Information*, 17(4):417–442.

Frixione, M. (2001). Tractable competence. *Minds and Machines*, 11(3):379–397.

Garey, M. R. and Johnson, D. S. (1990). *Computers and Intractability. A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, NY.

Gasarch, W. I. (2012). Guest column: The second P =?NP poll. *SIGACT News*, 43(2):53–77.

Ghosh, S., Heifetz, A., Verbrugge, R., and de Weerd, H. (2017). What drives people's choices in turn-taking games, if not game-theoretic rationality? In Lang, J., editor, *Proceedings of the 16th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 265–284.

Ghosh, S., Meijering, B., and Verbrugge, R. (2014). Strategic reasoning: Building cognitive models from logical formulas. *Journal of Logic, Language and Information*, 23(1):1–29.

Ghosh, S. and Verbrugge, R. (2018). Studying strategies and types of players: Experiments, logics and cognitive models. *Synthese*.

Gierasimczuk, N. and Szymanik, J. (2011a). Invariance properties of quantifiers and multiagent information exchange. In Kanazawa, M., Kornai, A., Kracht, M., and Seki, H., editors, *Proceedings of 12th Meeting on Mathematics of Language*, volume 6878, pages 72–89, Berlin. Springer.

Gierasimczuk, N. and Szymanik, J. (2011b). A note on a generalization of the muddy children puzzle. In Apt, K. R., editor, *Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 257–264. ACM Digital Library.

Goodie, A. S., Doshi, P., and Young, D. L. (2012). Levels of theory-of-mind reasoning in competitive games. *Journal of Behavioral Decision Making*, 25(1):95–108.

Hedden, T. and Zhang, J. (2002). What do you think I think you think?: Strategic reasoning in matrix games. *Cognition*, 85(1):1–36.

Icard III, T. and Moss, L. (2014). Recent progress in monotonicity. *Linguistic Issues in Language Technology*, 9:167–194.

Immerman, N. (1998). *Descriptive Complexity.* Texts in Computer Science. Springer, New York, NY.

Isaac, A., Szymanik, J., and Verbrugge, R. (2014). Logic and complexity in cognitive science. In Baltag, A. and Smets, S., editors, *Johan van Benthem on Logic and Information Dynamics*, volume 5 of *Outstanding Contributions to Logic*, pages 787–824. Springer International Publishing.

Karp, R. M. (1972). Reducibility among combinatorial problems. In Miller, R. E. and Thatcher, J. W., editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press.

Kinderman, P., Dunbar, R., and Bentall, R. P. (1998). Theory-of-mind deficits and causal attributions. *British Journal of Psychology*, 89(2):191–204.

Levesque, H. J. (1988). Logic and the complexity of reasoning. *Journal of Philosophical Logic*, 17(4):355–389.

McNaughton, R. and Papert, S. A. (1971). *Counter-Free Automata.* M.I.T. Research Monograph no. 65. MIT Press.

Meijering, B., Taatgen, N. A., van Rijn, H., and Verbrugge, R. (2014). Modeling inference of mental states: As simple as possible, as complex as necessary. *Interaction Studies*, 15(3):455–477.

Meijering, B., Van Rijn, H., Taatgen, N., and Verbrugge, R. (2011). I do know what you think I think: Second-order theory of mind in strategic games is not that difficult. In *Proceedings of the 33rd Annual Meeting of the Cognitive Science Society*, Boston. Cognitive Science Society.

Meijering, B., van Rijn, H., Taatgen, N., and Verbrugge, R. (2012). What eye movements can tell about theory of mind in a strategic game. *PLoS ONE*, 7(9):e45961.

Miller, S. A. (2009). Children's understanding of second-order mental states. *Psychological Bulletin*, 135(5):749.

Moschovakis, Y. (2001). What is an algorithm? In Enquist, B. and Schmid, W., editors, *Mathematics Unlimited – 2001 and Beyond*, pages 919–936. Springer.

Moss, L. and Tiede, H. J. (2006). Applications of modal logic in linguistics. In Blackburn, P., van Benthem, J., and Wolter, F., editors, *Handbook of Modal Logic*, Studies in Logic and Practical Reasoning, pages 1031–1077. Elsevier Science.

Osborne, M. J. and Rubinstein, A. (1994). *A Course in Game Theory*. MIT Press.

Peters, S. and Westerståhl, D. (2006). *Quantifiers in Language and Logic*. Clarendon Press, Oxford.

Piantadosi, S. T., Tenenbaum, J. B., and Goodman, N. D. (2016). The logical primitives of thought: Empirical foundations for compositional cognitive models. *Psychological Review*, 123(4):392.

van de Pol, I., Szymanik, J., and van Rooij, I. (2015). Parametrized complexity results for a model of theory of mind based on dynamic epistemic logic. In Ramanujam, R., editor, *Proceedings of the 15th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 239–248.

Pratt-Hartmann, I. (2004). Fragments of language. *Journal of Logic, Language and Information*, 13(2):207–223.

Pratt-Hartmann, I. (2008). Computational complexity in natural language. In Clark, A., Fox, C., and Lappin, S., editors, *Computational Linguistics and Natural Language Processing Handbook*. Blackwell.

Raijmakers, M. E., Mandell, D. J., Es, S. E., and Counihan, M. (2012). Children's strategy use when playing strategic games. *Synthese*, 191(3):355–370.

Ristad, E. S. (1993). *The Language Complexity Game*. Artificial Intelligence. MIT Press.

van Rooij, I. (2008). The tractable cognition thesis. *Cognitive Science: A Multidisciplinary Journal*, 32(6):939–984.

van Rooij, I., Kwisthout, J., Blokpoel, M., Szymanik, J., Wareham, T., and Toni, I. (2011). Intentional communication: Computationally easy or difficult? *Frontiers in Human Neuroscience*, 5(0):1–18.

Samer, M. and Szeider, S. (2009). Fixed-parameter tractability. In Biere, A., Heule, M., and van Maaren, H., editors, *Handbook of Satisfiability*, pages 363–393. IOS press.

Schlotterbeck, F. and Bott, O. (2013). Easy solutions for a hard problem? The computational complexity of reciprocals with quantificational antecedents. *Journal of Logic, Language and Information*, 22(4):363–390.

Shepard, R. N., Hovland, C. I., and Jenkins, H. M. (1961). Learning and memorization of classifications. *Psychological Monographs: General and Applied*, 75(13):1.

Stiller, J. and Dunbar, R. I. (2007). Perspective-taking and memory capacity predict social network size. *Social Networks*, 29(1):93–104.

Szymanik, J. (2010). Computational complexity of polyadic lifts of generalized quantifiers in natural language. *Linguistics and Philosophy*, 33(3):215–250.

Szymanik, J. (2013). Backward induction is PTIME-complete. In Grossi, D., Roy, O., and Huang, H., editors, *Proceedings of the Fourth International Workshop on Logic, Rationality and Interaction, Lecture Notes in Computer Science*, volume 8196, pages 352–356. Springer, Heidelberg.

Szymanik, J. (2016). *Quantifiers and Cognition. Logical and Computational Perspectives*. Studies in Linguistics and Philosophy. Springer.

Szymanik, J., Meijering, B., and Verbrugge, R. (2013). Using intrinsic complexity of turn-taking games to predict participants' reaction times. In Knauff, M., Pauen, M., Sebanz, N., and Wachsmuth, I., editors, *Proceedings of the 35th Annual Conference of the Cognitive Science Society*, pages 1426–1432, Austin, TX. Cognitive Science Society.

Szymanik, J. and Thorne, C. (2017). Exploring the relation of semantic complexity and quantifier distribution in large corpora. *Language Sciences*, 60:80–93.

Szymanik, J. and Zajenkowski, M. (2010). Comprehension of simple quantifiers. Empirical evaluation of a computational model. *Cognitive Science: A Multidisciplinary Journal*, 34(3):521–532.

Thorne, C. (2012). Studying the distribution of fragments of English using deep semantic annotation. In Bunt, H., editor, *Proceedings of the ISA8 Workshop*. SIGSEM.

Tsotsos, J. (1990). Analyzing vision at the complexity level. *Behavioral and Brain Sciences*, 13(3):423–469.

Verbrugge, R. (2009). Logic and social cognition: The facts matter, and so do computational models. *Journal of Philosophical Logic*, 38(6):649–680.

Verbrugge, R., Meijering, B., Wierda, S., Van Rijn, H., and Taatgen, N. (2018). Stepwise training supports strategic second-order theory of mind in turn-taking games. *Judgment and Decision Making*.

de Weerd, Verbrugge, R., and Verheij, B. (2015). Higher-order theory of mind in the tacit communication game. *Biologically Inspired Cognitive Architectures*, 11:10–21.

de Weerd, H., Verbrugge, R., and Verheij, B. (2017). Negotiating with other minds: The role of recursive theory of mind in negotiation with incomplete information. *Autonomous Agents and Multi-Agent Systems*, 31:250–287.

Zhai, F., Szymanik, J., and Titov, I. (2015). Toward a probabilistic mental logic for the syllogistic fragment of natural language. In Brochhagen, T., Roelofsen, F., and Theiler, N., editors, *Proceedings of the 20th Amsterdam Colloquium*, pages 468–477.