

Truth Theories, Competence, and Semantic Computation*

Peter Pagin

March 18, 2012, 10:02

1. Davidson on Truth Theories

In this paper I shall focus on two claims that Davidson made regarding the significance of Truth Theories (henceforth T-theories). The first claim was that they can serve a very central *explanatory role*, which we can summarize as follows:

(ER) An adequate T-theory for a language *L* explains how it is possible for speakers of *L* to effectively determine the meaning of any meaningful expression of *L*.

For a T-theory to be *adequate* for a language *L* it must meet certain constraints on interpretation of speakers that are codified in the methodology of *Radical Interpretation* (cf. Davidson 1973; Davidson 1974). It is clear that Davidson thought that speakers of natural languages are interpretable by means of the methodology of radical interpretation, and hence that the claim (ER) is not trivially true because of problems of interpretation.

*Parts of this paper have been presented at the LOGOS Seminar, at the department of Logic, History, and Philosophy of Science, University of Barcelona, April 2010, and at CSMN, Department of Philosophy, History of Art and Ideas, Oslo, November 2010. I am grateful for comments from people attending these talks, especially Herman Cappelen, José Díez, Manuel García-Carpintero, Olav Gjelsvik, Kathrin Glüer, Max Kölbel, Josep Macià, Genoveva Martí, José Martínez, and Sven Rosenkranz.

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement no. FP7-238128. The work on this paper has also received funding from the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement n° 229 441 - CCC, and from my research fellowship at IEA-Paris, spring 2010.

In some places, Davidson discussed this aspect of T-theories in terms of *learnability* of languages, as in this early passage:

When we can regard the meaning of each sentence as a function of a finite number of features of the sentence, we have an insight not only into what there is to be learned; we also understand how an infinite aptitude can be encompassed by finite accomplishments. For suppose that a language lacks this feature; then no matter how many sentences a would-be speaker learns to produce and understand, there will remain others whose meaning are not given by the rules already mastered. It is natural to say that such a language is *unlearnable*. This argument depends, of course, on a number of empirical assumptions: for example, that we do not at some point suddenly acquire an ability to intuit the meanings of sentences on no rule at all; that each new item of vocabulary, or new grammatical rule, takes some finite time to be learned; that man is mortal (Davidson 1965, 8-9).

In some passages, he spoke of this feature in terms of meaning *determination*:

In this paper I have assumed that the speakers of a language can effectively determine the meaning or meanings of an arbitrary expression (if it has a meaning), and that it is the central task of a theory of meaning to show how this is possible. I have argued that a characterization of a truth predicate describes the required kind of structure, and provides a clear and testable criterion of an adequate semantics for a natural language (Davidson 1967, 35).

No doubt Davidson did not see much difference between these two ways of setting out the requirements on a semantic theory: a speaker does know a language just in case she has the ability to determine the meaning of an arbitrary meaningful expression in it, and hence learning a language is the same as acquiring this ability.¹

The idea is by now familiar. First, you can learn the meanings of the simple expressions (usually words) one by one, since they are finitely many. Second, you

¹Of course, because of human cognitive and biological limitations there is an upper bound to the size of expressions she can even parse, let alone interpret, but since the number of expressions up to that size is so large, speakers cannot anyway learn the expressions one by one, and so the requirement does not depend on the assumption that there are infinitely many meaningful expressions. This is pointed out e.g. in Grandy 1990. In what follows I shall not make the proviso explicit.

can again learn the semantic significance of the syntactic modes of composition, since they are again finitely many. By means of these two ingredients you can then work out, step by step, the meaning of any arbitrary grammatical and meaningful expression. Davidson required that a semantic theory show how this is possible, by showing how the meanings of complex expressions depend on the meanings of their parts and the mode of composition.

This requirement on semantic theories is often identified with the requirement that the semantic theory be *compositional*, i.e. such that meaning of *L* expressions according to the theory satisfies the principle of compositionality:²

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

Since the relation between the idea of compositionality and the idea of being able to work out the meanings of complex expressions will be important throughout the paper, it might be good to be clear about the discrepancy right away. When Davidson in the second quote requires that speakers be able to determine the meanings “effectively”, this is naturally interpreted as requiring the existence of a humanly feasible *method* such that for each meaningful expression of the language (under some analysis), this method delivers the meaning of the expression (under that analysis) after a finite number of steps. Since this method is abstract and operates on syntactic objects, it is computational. So it is natural to regard Davidson’s requirement as a requirement of *computability*. By Church’s Thesis, any computable function is *recursive*.³ Hence, it is most natural to interpret Davidson’s requirement as the requirement that semantics be *recursive*:

(PRS) The meaning of a complex expression is recursive function of its parts, the

²Davidson did not at the time use the terms ‘compositional’ or ‘compositionality’. They were introduced (with a slightly restricted application) in a talk in Oxford 1960 by Hilary Putnam (1975, 77), and in print, with a slightly different meaning by Jerry Fodor and Jerrold Katz (1964). Only occasionally did Davidson use these terms later.

³This in the first place concerns arithmetical functions, but the definition of recursive functions can be carried over to other domains defined inductively by formal operations. This is straightforward for primitive recursive functions, but does involve some additional complications when minimization is added. Cf. Pagin 2011.

meanings of its parts and its mode of composition.⁴

There are two crucial differences between (PC) and (PRS). Firstly, compositionality by itself does *not* require the *composition functions*, i.e. the functions from part meanings and modes of composition to meanings, to be recursive. Since the composition functions need not be computable in a compositional semantics, neither need the semantic function itself be.

Secondly, since the semantic function according to (PRS) takes the syntactic parts themselves as arguments, the compositional substitution rules need not hold. The meaning of a complex expression need not be a function of the meanings of the parts. This is so since two parts may have the same meaning, but substituting one for the other will change the meaning of the complex expression, since one argument now is different, the part that is the expression (or term) itself.

Of course, some semantic functions are both compositional and recursive. In light of Davidson's requirement it is clear that he would have wanted the semantics to be recursive over and above being compositional. It is not so clear why one would want semantics to be compositional over and above being recursive (we shall return to this question).

In connection with stating the computability requirement, Davidson also suggested that T-theories, i.e. semantic theories of the form of Tarskian truth definitions, meet it. This is correct. In fact, T-theories are both recursive and compositional.⁵ It might seem that this is enough for making true Davidson's claim (ER) about the explanatory power of T-theories. This question will occupy us further on.

The immediate concern, however, is with Davidson's *second* claim about the significance of T-theories, concerning their *psychological relevance*:

⁴In quotes below from Davidson 1986, he is explicit about recursiveness.

⁵It is sometimes claimed that T-theories, and the language of predicate logic itself, is not compositional, since e.g. the truth of a formula $\exists xA$ under an assignment f does not depend only on the truth of A , the immediate constituent, under f . The example indeed shows that the language is not compositional with respect to the semantic value *truth, given an assignment*. On the other hand, as emphasized by Janssen (1997), it is compositional with respect to the *set of satisfying assignments, or function from assignments to truth values*, as the semantic value. As Janssen points out, these are used in the cylindrical algebras developed by Henkin, Monk, and Tarski (1971).

(NPR) Only the *theorems* of a T-theory are relevant to its interpretational adequacy.

A T-theory provides a method for computing a T-theorem for each sentence of the object language L in question, i.e. a theorem of the form

(T) s is true in L iff p

where ' s ' is replaced by a name of a sentence of L and ' p ' by a sentence of the meta-language. This is a so-called T-sentence. A consequence of (NPR) is that the actual cognitive architecture or cognitive processes of a speaker or interpreter are irrelevant to the adequacy of the theory. The actual cognitive architecture need not correspond at all to the structure of the T-theory. The structure is needed for generating the theorems but only the theorems generated are relevant to its interpretational adequacy.

This view is pretty explicit in the following passages:

I have frequently argued that command of such a theory [a T-theory] would suffice for interpretation. Here however there is no reason to be concerned with the details of the theory that can adequately model the ability of an interpreter. All that matters in the present discussion is that the theory has a finite base and is recursive, and these are features on which most philosophers and linguists agree (Davidson 1986, 95).

In any case, claims about what would constitute a satisfactory theory are not, as I said, claims about the propositional knowledge of an interpreter, nor are they claims about the details of the inner workings of some part of the brain. They are rather claims about what must be said to give a satisfactory description of the competence of the interpreter. We cannot describe what an interpreter can do except by appeal to a recursive theory of a certain sort. It does not add anything to this thesis to say that if the theory correctly describes the computation of an interpreter, some mechanism in the interpreter must correspond to the theory (Davidson 1986, 96).

This theme of opposing the *competence* of the interpreter and the actual cognitive *performance* is introduced much earlier in Davidson's writings, even if less explicitly:

What could we know that would enable us to [interpret his words]? How could we come to know it? The first of these questions is not the same as the question what we *do* know that enables us to interpret the words of others. For there may easily be something we could know and don't, knowledge of which would suffice for interpretation, while on the other hand it is not altogether obvious that there is anything we actually know which plays an essential role in interpretation (Davidson 1973, 125).

Why does Davidson insist on the cognitive irrelevance of the structure of T-theories? It is a theme of Davidson 1973 that the interpretational adequacy of a T-theory can be settled by means of radical interpretation, applying the *Principle of Charity*. When we select the best T-theories by means of applying Charity, no other property of the T-theories are used than the set of T-sentences that are their theorems. Hence, applying Charity, and nothing else, for selecting T-theories entails that no other property matters to their interpretational adequacy. In particular, whether the cognitive architecture or cognitive processes of speaker or interpreter correspond to the structure of a T-theory is not relevant to its adequacy.

Now, however, the question arises whether Davidson's two main claims, (ER) and (NPR), mutually cohere. The structure of the T-theory is what enables the derivations of the T-sentences, and is what is meant to explain that ability of interpreters, but it is at the same time supposed to be irrelevant to the acceptability of the theory. Does this add up? In the next section I shall consider two negative answers from the literature.

2. Larson/Segal and Lepore/Ludwig

In their 1995 book, Richard Larson and Gabriel Segal give their version of truth-theoretic semantics, developing in detail ways to handle a number of expression types and construction types in English. A main motive for Larson and Segal (LS) is to combine truth theoretic semantics with Chomskyan syntactic theory, and in particular the *Principles and Parameters* framework from the 1980s and -90s.

Of greater relevance is that LS also take over Chomsky's views about the nature of syntactic theory, and apply it to semantics. On Chomsky's view, syntactic theory is a psychological theory about the human language faculty, and attempts to

describe the *grammatical knowledge* of the speaker. Correspondingly, LS say

We can see semantics as a theory of the knowledge that underlies our ability to make semantic judgments. Semantic theory addresses one part of our linguistic knowledge: *knowledge of meaning* (Larson and Segal 1995, 10, emphasis in the original).

In our view, these questions should be approached from a **cognitivist perspective**, according to which knowledge of language is knowledge of a body of (largely unconscious) rules and principles that assign representations and meaning to the physical forms of signs (be they phonetic, visual, or tactile). On this conception, an answer to [the question 'What do we know?'] would specify these rules and principles and show how they affect the required mapping from sign to structure and meaning (Larson and Segal 1995, 11, emphasis in the original).

The view of LS about the role of a T-theory is clearly very different from Davidson's own. The theory is supposed to describe what the language user *actually* knows, not only what knowledge would be sufficient for an interpreter if he had it. Moreover, it does not only concern the theorems, the T-sentences, but also the rules and principles that assign meanings to expressions. Hence, on this conception, it would be possible for a T-theory to be false even if all of its T-sentences are true, viz. if the principles used in the theory to derive the theorems are not the principles the language user knows.

Moreover, they connect this different perspective on T-theories with explanatory power:

The hypothesis that we know a set of compositional semantic rules and principles is a highly attractive one having a great deal of explanatory power. In particular, it accounts for three notable and closely related features of linguistic competence. [...]

Second, the hypothesis accounts for the obvious but important fact that *we can understand new sentences*, sentences that we have never come across before. This too is easily explained if we have a body of rules that allow us to infer the meanings of new sentences from prior knowledge of the mean-

ings of their parts and from knowledge of the semantic significance of their combination.

Third, the hypothesis accounts for the slightly less obvious but equally important fact that *we have the capacity to understand each of an infinitely large number of sentences* (Larson and Segal 1995, 11-12, emphasis in the original).

Although there is no direct polemic with Davidson at this point, it appears to be the view of LS that *only* the theory that a compositional semantics is *actually known* explains the ability to understand new sentences and the ability to learn an infinite language. It is easy to understand why one would think so: a semantic theory that does *not* correspond to the actual knowledge or processing of the language user does not explain the ability that the user *actually has*; only a theory whose structure corresponds to how the user's mind works can do that. At least, so it may seem.

In a more recent book, Ernie Lepore and Kirk Ludwig (2005) argue in a similar direction as LS, but without such a strong cognitivist point of departure. It appears that they argue from the premise that the semantic theory should explain how speakers potentially can understand infinitely many sentences, to the conclusion that the theory captures what speakers actually know that enables them to have this ability.⁶ Lepore and Ludwig (LL) say:

It looks, then, as if the motivation for his requirement is that a compositional meaning theory should represent how finite speakers of a natural language can understand a potential infinity of sentences, that is, it looks as if the aim of a compositional meaning theory is to explain or exhibit somehow what such speakers know that enables them to understand potentially any sentence of their language. Furthermore, it looks as if to apply this requirement to analyses of particular expressions of natural language, a theorist must assume his analyses capture in part what speakers know in understanding such expressions as a part of their language. Thus, it appears that the theory Davidson produces must explain or exhibit what enables speakers to understand a potential infinity of sentence (Lepore and Ludwig 2005,

⁶In this context, I see no difference between the ideas of *being able to determine/work out the meaning of a sentence* and *potentially understand a sentence*.

31-2).⁷

They are explicit that the requirement of capturing what speakers know and what enables them to understand potentially any sentence of the language is stronger than the requirement of a finite specification that correctly assigns meanings to the sentences:

If this interpretation is right, then providing a finitely specifiable theory of a natural language that correctly assigns meanings to all of its sentences is a necessary, but not a sufficient, condition for providing an adequate theory. To be adequate, it must also recover the structure of our ability to speak and understand our language(s) (Lepore and Ludwig 2005, 32).

LL also elaborate on the requirement later:

And there is a fact of the matter about which axioms are correct, because there is a fact of the matter about the structure of speakers' dispositions to use words (2005, 124).

Thus, LL appear to end up in a position similar to that of LS. The cognitive structure, or structure of the speaker's dispositions, is relevant for the truth or falsity of a T-theory, over and above the fact that it satisfies the Principle of Charity and offers derivations of the theorems from a finite basis, i.e. enables someone who *knows* the theory to work out the meanings of the sentences of the language. For both LS and LL, satisfying the condition of explaining how it is possible for speakers to have the ability to work out the meanings of any sentence in the language requires that the theory somehow captures what speakers actually know, a requirement that was explicitly rejected by Davidson himself.

Are the commentators right and Davidson wrong? In the next section I shall argue that in one respect Davidson was in fact wrong, but not in the respect that LS and LL point to.

⁷LL use 'compositional' in two distinct but both non-standard senses, one as a property of a meaning theory, and the other as property of a language. For the second they say "What this seems to require is that our ability to speak a language be based on our understanding of a finite number of semantical primitives and rules for their combination. We will call any such language compositional" (2005, 27).

3. What property of T-theories is required?

We have two questions:

- i) Can a theory that does not capture the cognitive structure of the speaker's ability to work out the meanings of new sentence explain how it is possible for the interpreter to do so?
- ii) Do T-theories in fact offer a plausible model of the competence of speakers of natural language?

As far as I have understood them, both LS and LL answer question (i) in the negative. This answer, I think, is wrong, since it does not accurately reflect the modal character of Davidson's claim, as stated in (ER):

(ER) An adequate T-theory for a language L explains how it is possible for speakers of L to effectively determine the meaning of any meaningful expression of L .

Let us distinguish between three kinds of claim:

(Lang) For a *language* L there is a non-empty set of properties \mathcal{P} of speakers such that if a speaker S has any property P_i in \mathcal{P} , S can work out the meaning of any sentence in L .

(Poss) Having a cognitive structure captured by a T-theory for L is a property in \mathcal{P} , and it is possible for a speaker to have this property.

(Actu) Having a cognitive structure captured by a T-theory for L is a property in \mathcal{P} , and speakers of L actually have this property.

The first of these three kinds of claim, (Lang), applied to English and to natural languages in general, was actually made by Davidson and is shared by his commentators. It is established as true if we manage to give a recursive theory, such as a T-theory, for the natural language L in question. For then, knowledge of this theory is in fact enough for being able to work out the meaning of any sentence of

L. So, (Lang) is uncontroversial.⁸

The left conjunct of (Poss) is uncontroversial as well, and the right conjunct at most only a little less obvious. If a language user can learn the T-theory, there is little reason to doubt that she can use it to work out the meanings of the sentences of *L*.⁹ Similarly for the left conjunct of (Actu). With English as the instance of *L* it is of course not clear that a full T-theory can actually be given, although it has been done for fragments of English. But even with respect to such fragments, there is hardly any evidence that speakers of English have a cognitive structure that corresponds to the T-theories.

The main objection against LS and LL is, however, that what makes (ER) true, i.e. the explanation of how it is possible to determine the meanings of new sentences, does not require (Actu): it is enough with (Lang) and (Poss). The language allows for cognitive structures that make language users equipped to do this, and it is possible for language users to possess at least one such cognitive structure. Hence, determining the meaning of new sentences is something that is possible for speakers/interpreters of a language for which a T-theory exists. For explaining the possibility, nothing about actual cognitive structures need be assumed.¹⁰

Hence, the answer to question (i) is positive. Turning to the second:

⁸This needs to be qualified, since much turns on the nature of the modality involved. What is uncontroversial is that a recursive theory makes it possible to work out the meaning of a new theory in a finite number of elementary steps, provided that there is no finite upper bound to the number of elementary steps that can be performed. But since complexity issues haven't played any role in Davidson's, LS's or LL's discussions, (Lang) is seen as uncontroversial.

⁹In what sense of 'possible' is the left conjunct of (Poss) uncontroversially true? Is it a question of physical possibility, or perhaps biological possibility? This question was pressed by José Diez. It seems to me that the most natural answer is to take the modality in question to be scientific-epistemic: it is possible that *p* iff it is consistent with current scientific knowledge about human biology that *p*. Under this understanding, it is both knowable and plausible that the right-hand conjunct of (Poss) is true.

¹⁰If we use the locution '*X* has the ability to Φ ' rather than 'it is possible for *X* to Φ ', then intuitions might be different, since the latter may be taken to imply only that it is *possible* for *X* to have the ability, while *having* the ability involves actually having some particular cognitive structure that subserves the ability. It may then be said that (Lang) plus (Poss) only explain the possibility of such an ability, not its actuality.

This may be right, as far as intuitions regarding the meaning of 'ability' goes. The modal character of Davidson's claim is another matter. There is at least one reasonable interpretation of Davidson according to which his modal claim is the weaker one, expressed by (ER), and this interpretation makes his claim consistent with the irrelevance claim (NPR).

As stressed to me by Manuel García-Carpintero and Josep Macià, we are certainly interested as well in what actually subserves our linguistic abilities, but unless a semantic theory is *about* cognition, a semantic theory is not rendered false by not corresponding to cognitive structure.

- (ii) Do T-theories in fact offer a plausible model of the competence of speakers of natural language?

It is clear that Davidson thought that it too has a positive answer, and in this case the view is obviously shared by LS and LL. However, that this is the correct answer is not itself obvious. One might believe it correct since T-theories are recursive (and assigns semantic significance to the intuitively right units), from which it follows that knowledge of T-theories is enough for the ability to work out the meaning of new sentences. However, modeling this ability is *not* itself enough for modeling the abilities of natural language interpreters.

The reason is that in order to model the ability of natural language interpreters, the theory must not only allow speakers to compute the meanings of any sentence, but allow them to compute the meanings of any sentence *in a time-efficient way*. Our cognitive resources are limited; we cannot perform cognitive tasks that are extremely complex, even if they can be solved by an effective method. As interpreters of our mother tongues we normally understand utterances on-line, i.e. as they are being made, and usually accurately, by common sense standards. Our reading comprehension is even faster: we can read a passage with understanding faster than it would normally be spoken. That a T-theory is recursive, and hence provides a method for computing meanings, is not enough for capturing this, since a method of computation can be very inefficient, even if effective in the computational sense. If applying a particular method of computation to a particular task, like working out the meaning of a sentence, under some reasonable assumptions would take a year, while speakers in fact understand them immediately, then this method of computation is not plausibly a good model of human comprehension.

In order to assess the question whether T-theories in fact provide a reasonable model of human comprehension, we need to get clear about their efficiency for semantic computation, i.e. for computing meanings by means of applying the theory. I shall take a look at basic T-theories from this perspective in section 5. In the next section, I shall first provide a brief conceptual background for the question of computational efficiency.

4. Computational complexity and semantics

Classical computational complexity theory is concerned with giving mathematical measures of the difficulty of mathematical problems. The problem need not be a problem within any standard branch of mathematics, such as number theory or geometry, but must be a problem that can be adequately represented in a formal language, as an input to computation.

For measuring the complexity of a problem one needs a computation method. One then asks how much of resources is needed by this method for arriving at a solution to the problem. A standard method that is used as a reference device in this sense is that of one-tape Turing machines. One of the standard resources is time, in the sense of the number of computation steps needed by the Turing machine for arriving at the solution. This is so-called *time complexity*.

With a problem type and computation method is associated a *time complexity function* C . This is a function that takes as argument a measure of the *size* of a problem instance, as a numerical value, and gives as value the size of the *largest* computation that is needed to compute any problem of the same size. We can illustrate this with one of the most classical examples, the problem of the *Traveling Salesman*: a salesman is to visit a number k of cities exactly once and then return home, and the problem is to find a visiting order that minimizes the total distance traveled. In this case the solution consists in selecting the optimal order and verifying that it is optimal. The number of cities k is the *size* of the problem instance, and this is the argument to the complexity function C . Its value $C(k)$ is the number of computation steps needed *at most* for determining the solution for any problem instance of size k .

In complexity theory one is interested not so much in the value of C for a particular argument, but rather in how fast the value $C(k)$ *grows* when the argument increases. If $C(k)$ is bounded by a linear function of k , the time complexity is said to be linear; if it is bounded by k^n , for some natural number n , the time complexity is said to be *polynomial*, or equivalently that the problem is solvable in polynomial time.

Problems that are solvable in polynomial time are generally regarded as *tractable*,

or *feasible*, while if the value of the complexity function grows faster, they are said to be *intractable* (this is known as the *Cobham-Edmonds thesis*). It is not known whether the traveling salesman problem is intractable in this sense. The reason is that no method is known for determining the solution (with certainty, and for any finite k) that is more efficient than calculating the total traveling distance for each visiting order and selecting the shortest. Since the number of visiting orders for k cities is $k!$, the factorial of k , and since $k!$ grows faster than k^n , for any n , the general problem is intractable if there is no method sufficiently faster than checking all possible orders of traveling.¹¹

How does this apply to semantic interpretation? We need a method of computing meanings from disambiguated expressions as inputs. If we then think of semantics in functional terms, we want to compute a semantic function μ that takes as arguments disambiguated expressions — *grammatical terms* — and gives as values *meanings* m of some sort in this format:

$$\mu(t) = m.$$

Since meanings are non-syntactic abstract entities, they must be syntactically represented, i.e. by means of a sufficiently formal meta-language ML . That means that in an equation instance of this format, ‘ t ’ is replaced by an expression denoting a grammatical term, and ‘ m ’ by an expression of ML .

Then we need an algorithmic method of some kind for computing meanings. A type of method that particularly well suited is that of *term rewriting systems*. In general, a term rewriting system (a TRS) \mathcal{R} is a pair (F, R) of a signature F and a set R of rewrite rules over that signature. The signature consists of a set of basic terms, and a set of operators. To this is added a set of *rewrite variables* which are used in stating the rules. A rewrite rule has the form

$$F(\vec{x}) \rightarrow G(\vec{y})$$

¹¹For interesting partial results concerning this problem, cf. the Wikipedia article http://en.wikipedia.org/wiki/Travelling_salesman_problem. The problem is known to be *NP hard*, which entails that if $NP \neq P$, as is generally believed, it is intractable.

(where the arrows over the variables indicate that it is a sequence of variables).¹²

An example would be

$$h(x_1)bx_2 \rightarrow g(x_1, c)bd$$

where ‘ b ’, ‘ c ’ and ‘ d ’ are constants. Every rule application is a substitution operation, where an instance of the left-hand-side (lhs) of the rule is replaced by the corresponding instance of the right-hand-side (rhs) of the same rule. The substitution may be performed on a subterm of a larger term. An instance of a term s is any term s' resulting from s by uniform substitution by terms for rewrite variables. Thus, ‘ $h(s_7)bf(s_9)$ ’ is an instance of the lhs above.

A *derivation* is a sequence of rule applications, where every step except the initial one is an application to a term that results from a previous step. In case a term is reached such that no rule of the TRS applies to it (and hence not to any of its subterms either), the derivation has *terminated*, and the term is said to be in *normal form*. The original term is then *reduced* to normal form. A rewrite system R terminates iff every derivation eventually leads to a term in normal form. \mathcal{R} is said to be *confluent* iff it holds for any distinct terms s_1, s_2, s_3 such that s_2 and s_3 both can be derived from s_1 , that there is a term s_4 such that s_4 can be derived from both s_2 and s_3 . \mathcal{R} is *convergent* iff \mathcal{R} both terminates and is confluent.

Rewriting systems are general computation devices, in the sense that the reduction of a rewrite term to normal form is a computation. It is a standard result that any Turing machine can be simulated by a term rewrite system (cf. Baader and Nipkow 1998, 94-97). We also get a very natural measure of time complexity by just counting the number of rule applications, i.e. reduction steps, until normal form is reached. One reason why term rewriting is a natural choice for semantic interpretation is that the clauses by means of which a semantic system is defined correspond closely to rewrite rules, and can be transformed into rules by a minimal change.

To illustrate, consider Davidson’s *Annette* example (Davidson 1967, 17-18) of a compositional semantics:

¹²For an excellent introduction to term rewriting, see Baader and Nipkow 1998.

- (1) i) $\text{Ref}(\text{'Annette'}) = \text{Annette}$
 ii) $\text{Ref}(\text{'the father of'} \wedge t) = \text{the father of } \text{Ref}(t)$

This simple definition has the form of a system of equations, and provides a method for deriving the interpretation of ‘the father of the father of the father of Annette’ in four steps of substitution. Let ‘ F ’ be the object language father operator and ‘ \mathbf{F} ’ its analogue in the meta-language, and let ‘ a ’ be the object language name of Annette. Then we have in four steps with the semantic function μ_a :

$$\begin{aligned}
 (2) \quad & \mu_a(F(F(F(a)))) \\
 & = \mathbf{F}(\mu_a(F(F(a)))) \\
 & = \mathbf{F}(\mathbf{F}(\mu_a(F(a)))) \\
 & = \mathbf{F}(\mathbf{F}(\mathbf{F}(\mu_a(a)))) \\
 & = \mathbf{F}(\mathbf{F}(\mathbf{F}(\text{Annette})))
 \end{aligned}$$

where (what corresponds to) the second clause of (1) is applied three times and the first clause once.

Each derivation step in (2) is a substitution step. Each substitution is performed in accordance with (what corresponds to) equations in (1). These equations are applied only for substitution from left to right: an instance of the left-hand side is replaced by the corresponding instance of the right-hand side. We have then in fact used the system as a rewrite system. To make that explicit, replace the identity signs with left-right arrows:

- (3) i) $\mu_a(a) \rightarrow \text{Annette}$
 ii) $\mu_a(F(x)) \rightarrow \mathbf{F}(\mu_a(x))$

In rewrite system (3), any term of the system is reduced to normal form in a number of steps that is identical to number of symbol occurrences (i.e. occurrences of ‘ F ’ and ‘ a ’) of the term. If we take the size of the problem to be the size of the input term then the associated time complexity function $C_{(3)}$ is the identity function. That is, $C_{(3)}(k) = k$.

We can easily speed up the the system by adding a third rule:

- (4) i) $\mu_a(a) \rightarrow \text{Annette}$
 ii) $\mu_a(F(x)) \rightarrow \mathbf{F}(\mu_a(x))$
 iii) $\mu_a(F(F(x))) \rightarrow \mathbf{F}(\mathbf{F}(\mu_a(x)))$

Because of the third rule, two occurrences of ‘ F ’ can be processed in one step. So with this addition we get another complexity function: $C_{(4)}(k) = k/2 + 1$ for odd k (i.e. even number of F ’s), and $k + 1/2$ for even k . Clearly, by applying this method, for each system we can find another that is more efficient with respect to time. Still there is an upper bound the speed-up. Since for any system there is finite number n such that no rule application processes more than n symbol occurrences, for that system each full reduction to normal form will take at least k/n steps. Hence, no system has reductions faster than linear time.

The speed-up between systems (3) and (4) is acquired at the cost of enlarging the rule system, adding a redundant rule. Hence, we can see that there is a trade-off between the size of the system, with respect to the number of rules, and the speed of the system. It is natural to ask for the speed of a system that has a minimal number of rules, i.e. a system R such that for any equivalent system R' , one that reduces the same input terms to the same normal form terms, R' has at least the same size as R . It is natural to set identity as the maximum of efficiency for such a system. That is, if for a minimal rule system R the corresponding time complexity function C_R is such that $C_R(k) \leq k$, then we say that R has maximal efficiency. $C_{(3)}$ is maximally efficient in this sense.¹³

Our question will then concern the properties of T-theories considered from a computational perspective, i.e. with the clauses of the T-theory reinterpreted as rewrite rules.

5. The computational properties of T-theories

5.1. Connectives

One possible computational perspective on T-theories is that of looking at the number of elementary steps are needed within a *system of logical deduction*, such

¹³For a more thorough presentation and discussion of these issues, and for sketches of proofs that a certain form of compositional semantics is both necessary and sufficient for maximal efficiency, see Pagin 2012.

a system of Natural Deduction, with T-theoretical non-logical axioms added. The resulting derivation system would be quite complex. To carry out the derivation many extra steps would be needed compared with a derivation of simple substitution steps, even in the case of the propositional fragment.

Davidson himself had something much simpler and much more efficient in mind, which he called *canonical proofs*:

A canonical proof, given a theory of truth, is easy to construct, moving as it does through a string of biconditionals, and requiring for uniqueness only occasional decisions to govern left and right precedence (Davidson 1973, 138).

What Davidson probably had in mind can be illustrated with a miniature T-theory. We use ‘T’ as truth predicate ‘S, S’ etc. as schematic object language (OL) sentence letters, ‘s₁, s₂’ etc. as names of OL sentences, and ‘p₁, p₂’ etc. as (abbreviations of) ML sentences. We use ‘^’ as concatenation operator, ‘⊢’ for marking theoremhood, and ‘&’ as OL conjunction operator. Then we have:

- (5) i) ⊢ T(S^‘&’^S’) iff T(S) and T(S’)
 ii) ⊢ T(s₁) iff p₁
 iii) ⊢ T(s₂) iff p₂

Together with axioms (5ii) and (5iii), and axiom schema (5i), we have two rules: Ins and Sub.

(Ins) If ⊢ F(\vec{S}), then ⊢ F((\vec{s} / \vec{S}))

(Sub) If ⊢ F(e) and ⊢ e iff e’, then ⊢ F(e’/e)

Here ‘ \vec{s} ’ is short for s₁, ..., s_n (for given n). ‘F(...)’ is any context in a theorem, and ‘F([x/y])’ is the result of substituting ‘x’ for ‘y’ in the ‘F’ context (and element for element in case ‘x’ and ‘y’ are vectors). So (Ins) is an instantiation rule: where for each S_i, s_i is a proper instance, the rule allows admitting as an axiom the corresponding instantiation of a T-theoretic schema like (5i). (Sub) is a substitution rule, allowing substitution of the right-hand-side of a T-theorem for the left-hand-side as occurring in another T-theorem.

A derivation in system (5), with rules (Ins) and (Sub), of the theorem for the sentence ' $s_1 \wedge \text{'\&'} \wedge s_2$ ' then runs as follows:

- (6) (i) $\vdash T(s_1 \wedge \text{'\&'} \wedge s_2)$ iff $T(s_1)$ and $T(s_2)$ (by (5i), (Ins))
(ii) $\vdash T(s_1 \wedge \text{'\&'} \wedge s_2)$ iff p_1 and $T(s_2)$ (by (i), (5ii), (Sub))
(iii) $\vdash T(s_1 \wedge \text{'\&'} \wedge s_2)$ iff p_1 and p_2 (by (ii), (5iii), (Sub))

(6) is probably a canonical derivation in the sense of the quotation from Davidson. We can see that it is very efficient. In (6) we have completed the derivation of the T-theorem for the complex sentence ' $s_1 \wedge \text{'\&'} \wedge s_2$ ' in three steps, where each step corresponds to interpreting one of the primitive symbols of the sentence: the connective in the first step, and the atomic sentences in the two following steps. As long as our system consists only of atomic sentence axioms like (5ii) and (5iii) and schemata for the connectives, it is straightforward to show by induction over the sentence complexity (the number of primitive symbols in the *OL* sentence), that the number of steps in the derivation of the T-theorem for an *OL* sentence s will be exactly as many as the number of primitive symbols in s . Hence, any such system has maximal efficiency in the sense of section 4.¹⁴

The corresponding rewrite system consists of rules by which the right hand sides (rhs) of the conditionals in (5) are substituted for the corresponding left hand sides (lhs). That is, we have

- (7) i) $T(S \wedge \text{'\&'} \wedge S') \longrightarrow T(S)$ and $T(S')$
ii) $T(s_1) \longrightarrow p_1$
iii) $T(s_2) \longrightarrow p_2$

There is no need for any further rules, for instantiation and substitution are already built in to the framework of rewrite systems. The corresponding derivation, i.e. reduction of the initial term ' $T(s_1 \wedge \text{'\&'} \wedge s_2)$ ' to normal form again requires three steps.

¹⁴Using only canonical derivations reduces complexity, but also guarantees that we don't get irrelevant theorems (as was also stressed by Max Kölbel). If we take the set of T-theory theorems to be closed under logical equivalence on the right-hand side, we would get denumerably many non-interpretive theorems: if $\vdash T(s)$ iff p is a theorem, then so is e.g. $\vdash T(s)$ iff p and (if q , then q), for any q .

- (8) (i) $T(s_1 \wedge ' \& ' \wedge s_2)$ (initial term)
(ii) $T(s_1)$ and $T(s_2)$ ((i), (7i))
(iii) p_1 and $T(s_2)$ ((ii), (7ii))
(iv) p_1 and p_2 ((iii), (7iii))

In this derivation, the initial term instantiates the lhs of rule (7i), and so we can apply the rule and substitute the initial term with the corresponding instance of the rhs of (7i), which gives us (8ii). The term rewriting derivation is perhaps a more plausible model of the comprehension process than the canonical proofs of T-theorems, since it ends with an *ML* content representation rather than a biconditional, from which a further inference would be needed to ascribe content to an utterance.

5.2. Predicates and terms

What happens when we add a term-predicate structure to atomic sentences? The T-theoretic method is to add reference axioms to constant terms and satisfaction axioms for atomic predicates:

- (9) $\text{Ref}(t) = a$
(10) $\text{sat}(f, P(x_1, \dots, x_n)) \text{ iff } G(f(x_1), \dots, f(x_n))$

In (9) 'Ref' denotes the reference function, '*t*' is a term in *ML* referring to a singular closed term in *OL*, and '*a*' a term in *ML* referring to an object. In (10), '*f*' is a function symbol for an *assignment function*, i.e. a function assigning values to *OL* variables. 'sat' denotes the *satisfaction* relation between assignments and open sentences. '*P*' denotes an *n*-place *OL* predicate and '*G*' is a *n*-place *ML* predicate. Implementing reference axioms like (9) into a rewrite system at first appears straightforward, since it seems to be just a matter of replacing the identity sign with an arrow:

- (11) $\text{Ref}(t) \longrightarrow a$

For reasons given below, the proposal will be slightly different, however.

A rewrite rule for the predicate axioms is clearly not straightforward, because of the variables. The variables in (10) are *OL* variables. They belong to the *OL* syntax, and are not *rewrite variables*. Hence, a rule like

$$(12) \quad \text{sat}(f, P(x_1, \dots, x_n)) \longrightarrow G(f(x_1), \dots, f(x_n))$$

is simply a rule for replacing the particular lhs by the particular rhs, with these particular *OL* variables, ‘ x_1, \dots, x_n ’. Since there are denumerably many variables, we would need denumerately many rules corresponding to the predicate ‘ P ’, one for each choice of n variables. The way to get around this problem is to use a special type of rewrite variables, ‘ v_1, v_2 ’ etc., whose instances are *OL* variables. Similarly, ‘ f ’ is a free *ML* variable over assignment functions, and needs to be replaced a rewrite variable of the appropriate sort: ‘ g, g_1, g_2 etc.’ Instead of (12) we then have

$$(13) \quad \text{sat}(g, P(v_1, \dots, v_n)) \longrightarrow G(g(v_1), \dots, g(v_n))$$

Now the lhs of (12) is an *instance* of the lhs of (13), and analogously for the rhs.¹⁵

The next step is to add rules for processing closed terms as arguments to atomic predicates. A standard way of doing this in T-theories is exemplified in:

$$(14) \quad \text{sat}(f, P(t)) \text{ iff } \text{sat}(f[\text{Ref}(t)/x_j], P(x_j))$$

where x_j is some variable that does not already occur in the relevant context. $f[\text{Ref}(t)/x_j]$ is the assignment function that is like f except that it assigns $\text{Ref}(t)$

¹⁵Normally, in a sentence like

$$(i) \quad \text{sat}(g, \ulcorner P(x_1, \dots, x_n) \urcorner)$$

where ‘ P ’ is an *OL* predicate, we simply use the variables autonomously. That is, in the meta-language, we use ‘ x_1 ’ as a structural-descriptive name of the *OL* variable ‘ x_1 ’ itself. But we can as well use free *ML* variables for *OL* variables to write

$$(ii) \quad \text{sat}(g, P(v_1, \dots, v_n))$$

with ‘ P ’ now in *ML*. (ii) “says that” the value of ‘ g ’ (which is an assignment function, e.g. f_{12}) satisfies the formula constructed from the value of ‘ P ’ (which is an *OL* predicate) with the arguments that are the values of ‘ v_1, \dots, v_n ’ (which are some *OL* variables).

Since the separation of *OL* and *ML* variables will involve explicit quantification over *OL* variables, it will also be made explicit that e.g. an existential sentence is true iff there are *some* *OL* variables v_1, \dots, v_n and *some* assignment function g that satisfies the formula because of what it assigns to v_1, \dots, v_n . No particular *OL* variables are used. This makes it fully clear that the truth conditions do *not* depend on the choice of *OL* variables. An elementary but desirable outcome.

to x_j . Hence, we have

$$f[\text{Ref}(t)/x_j](x_j) = \text{Ref}(t)$$

as desired. The problem with (14), however, is the requirement of selecting an appropriate variable. That this condition cannot be dropped is illustrated by replacing in (14) ' $P(t)$ ' by ' $R(t, x_j)$ '. In such a case, what f assigns to x_j matters, but after the substitution of ' x_j ' for ' t ' and the updating of f to $f[\text{Ref}(t)/x_j]$, the assignment of f to ' x_j ' is lost.

But to accommodate the requirement of variable selection, the checking for variable identity must be computationally implemented. This would require a rewrite system quite different from one that simply corresponds to clauses in a T-theory, and would in any case considerably increase the number of derivation steps.

Fortunately, there is a different solution. We extend the assignment functions to take constants as well as variables as arguments. Instead of rules corresponding directly to the form of (11), we have rules of the form

$$(15) \quad g(t) \longrightarrow a$$

where ' g ' is a rewrite variable for assignment function expressions. By means of this rule, for the constant ' t ', any term ' $f_i(t)$ ' may be replaced by ' a '. We also add rewrite variables y_1, y_2, \dots that take all singular terms, constants as well as variables, as substituends. Hence, ' $f_i(t)$ ' is an instance of ' $g(y)$ '.¹⁶ With this change, all we need for atomic formulas are rules of the form

$$(16) \quad \text{sat}(g, P(y_1, \dots, y_n)) \longrightarrow G(g(y_1), \dots, g(y_n))$$

one for each atomic predicate, and rules of the form (15), one for each constant (simple) singular term.

We also need to revise the rules for the connectives to cover the satisfaction clause. So, instead of (7i) we will have

¹⁶This proposal is quite similar to the practice in model theory of defining $I_f(t)$ to be $I(t)$ in case t is a constant, and $f(t)$ in case t is a variable, where I is the interpretation function. And, of course, such hybrid functions can be employed in T-theories proper as well, yielding the corresponding simplification.

$$(17) \quad \text{sat}(g, S \wedge ' \&' \wedge S') \longrightarrow \text{sat}(g, S) \text{ and } \text{sat}(g, S').$$

5.3. Quantifiers

Tarski showed how to handle quantification in the object language recursively by means of varying the assignments (sequences) in the meta-language. There are two equivalent standard ways of implementing this idea. Tarski's own method was to quantify explicitly over assignments with a restrictive condition of sameness except at most with respect to the variable in question. Let's use ' $D(f', f, x)$ ' to mean that assignment f' differs from assignment f at most in what it assigns to the variable ' x '. Then the first standard way is given by

$$(18) \quad \text{sat}(f, \exists x(F(x))) \quad \text{iff} \quad \exists f'(D(f', f, x) \ \& \ \text{sat}(f', F(x)))$$

Supposing ' F ' is an atomic predicate, then with an appropriate axiom for the predicate we can get something like

$$(19) \quad \text{sat}(f, \exists x(F(x))) \quad \text{iff} \quad \exists f'(D(f', f, x) \ \& \ G(f'(x)))$$

In one respect, this is acceptable from a computational point of view: the prime operator ' $'$ ' automatically generates a new assignment variable ' f' ' from a given assignment variable ' f '. The rhs of (18) is mechanically generated from the lhs of (18), in a way that can be straightforwardly adapted in a rewrite system.

In two other respects, (18) is less adequate. The first problem is that explicit quantification over assignment functions is not plausible as a representation of mental content. This is perhaps a minor problem, but it is related to the second, which is more serious. The rhs of (19) does not show whether or not f' depends on the choice of f . In the particular example it does not, since ' $\exists x F(x)$ ' is a (closed) sentence, and if one assignment satisfies it, so does every assignment. Since we want a complete interpretation, we want the final rhs of the T-theorem not to contain any unnecessary semantic vocabulary. Because the *OL* sentence is closed, nothing turns on the choice of f , and hence (19) reduces to

$$(20) \quad \text{sat}(f, \exists x(F(x))) \quad \text{iff} \quad \exists f'(G(f'(x)))$$

But the reducibility of (19) to (20) requires proof, from the determination of the fact that the *OL* formula does not contain any free variables, and this adds a number of extra derivation steps.

The second standard way of implementing the variation over assignments is apparently due to David Wiggins (1980, 325), and was employed above in (14).¹⁷ Instead of quantifying over assignments, we quantify over objects and construct assignment functions from the objects of the domain. Thus, instead of (18) we will have

$$(21) \quad \text{sat}(f, \exists x(F(x))) \quad \text{iff} \quad \exists b(\text{sat}(f[b/x], F(x)))$$

where $f[b/x]$ is the assignment that is like f except for assigning b to x . Using again the assumed axiom for ‘ F ’ we arrive at

$$(22) \quad \text{sat}(f, \exists x(F(x))) \quad \text{iff} \quad \exists b(G(f[b/x](x)))$$

As observed above, we have in general

$$f[a/x](x) = a.$$

Therefore, (22) reduces by direct singular term substitution to

$$(23) \quad \text{sat}(f, \exists(xF(x))) \quad \text{iff} \quad \exists b(G(b))$$

Since ‘ f ’ does not occur in the rhs of (23), it is clear that the truth of (23) does not depend on the choice of f . Clearly, the rhs here amounts to a full interpretation.

Still, there is a problem with this method. In (22) and (23), ‘ b ’ is a bound *ML* variable. If the *OL* sentence contains several quantifier occurrences, several *ML* variables will have to be introduced in the derivation. We need to keep track of the choice of variables, so that a particular variable does not get inadvertently bound by the wrong quantifier occurrence. But this apparently involves an extra round of checking for identity of variables, which polynomially increases the length of the computation.

¹⁷The originality claim is made in Lepore and Ludwig 2007, 73.

Luckily, there is a solution also to this problem: we use *OL* variables to *index* the *ML* variables. This means that in the above examples (21)–(23) ‘*b*’ is replaced by ‘*b_x*’. This method ensures that if ‘*x₁*’ and ‘*x₂*’ are bound by distinct quantifier occurrences in the *OL* sentence, the corresponding *ML* variables ‘*b_{x₁}*’ and ‘*b_{x₂}*’ are bound by distinct quantifier occurrences in the corresponding *ML* sentence. No extra computation is needed.

Also, the variable indexing can be directly implemented in the rewrite system. All we need is to extend the rewrite system to handle quantified sentences is one rule for each quantifier, and an assignment substitution rule:

$$(24) \quad \text{sat}(g, \exists v(S)) \longrightarrow \exists b_v(\text{sat}(g[b_v/v](S)))$$

$$(25) \quad g[b_v/v](v) \longrightarrow b_v$$

The system is still not complete, however, because of a particular complication, illustrated by the sentence

$$(26) \quad \exists x_1 \exists x_2 (P(x_1 x_2)).$$

Starting with an assignment function symbol *f* and applying the quantifier rule (24) and a predicate rule for *P* gives us

$$(27) \quad \exists b_{x_1} \exists b_{x_2} (G(f[b_{x_1}/x_1][b_{x_2}/x_2])(x_1), f[b_{x_1}/x_1][b_{x_2}/x_2])(x_2)).$$

(27) can be reduced by applying rule (25) to

$$(28) \quad \exists b_{x_1} \exists b_{x_2} (G(f[b_{x_1}/x_1][b_{x_2}/x_2])(x_1), b_{x_2})$$

but that rule does not provide for reduction of the first argument to *G*, because the outermost form of the modified assignment symbol does not correspond to the argument.

We could of course add a rule that would pick up the second outermost modification instead, but since there is no finite upper bound to the number of possible assignment function modifications, we would need infinitely many rules to solve the problem. This is not really an option, since if we were allowed infinitely many rules in the first place, we could have one rule for each sentence, thereby solving

every interpretation problem in unit time.

Instead we need a rule that allows us to “dig out” the right modification, step by step, until we can apply rule (25). This need is satisfied by

$$(29) \quad g[b_{v_i}/v_i](v_j) \longrightarrow g(v_j), \quad \text{if } i \neq j.^{18}$$

By means of (29) we can reduce (28) to

$$(30) \quad \exists b_{x_1} \exists b_{x_2} (G(f[b_{x_1}/x_1])(x_1), b_{x_2})$$

to which rule (25) can be applied to give a normal form.

In case we want to keep the original T-sentence format, an additional rule is added:

$$(31) \quad T(S) \longrightarrow \text{sat}(g,S)$$

Here we use the assignment variable ‘g’ free, rather than bound by an existential or universal quantifier. The reason is that if the T predicate applies (without any context relativity), then the argument is a closed sentence and it will then not matter which assignment we start out with. As is shown in the rules (24) and (25), the assignment variable is eventually dropped from the rewrite term. If, instead of (31) we were to have

$$(32) \quad T(S) \longrightarrow \exists g(\text{sat}(g,S))$$

then the final rewrite term will still contain the large scope quantifier ‘ $\exists g$ ’ vacuously. It cannot be removed more properly except by a rule that involves checking whether it occurs vacuously or not, and this again involves extra computation.

The outline of the rewrite system is now complete, and we can turn to the question of its time complexity.

6. Time complexity of SAT rewrite systems

As is shown in the Appendix, except for the modification dig-out complication, a rewrite system of this kind, a SAT rewrite system, has the property that each ele-

¹⁸Note that (29) is a *conditional* rule, with an elementary condition. Conditional term rewriting is one of the extensions to standard term rewrite systems. See e.g. Terese 2003, 80-85.

ment of the *OL* sentence, a singular term, a predicate, a connective, or a quantifier, is interpreted by one single rule application. Hence, if it were not for the dig-out complication, the time complexity function C_T would be the identity function: $C_T(k) = k$. The system would model interpretation with maximal efficiency, in the sense of section 4.

What is the addition to complexity by the dig-out rule? Applications of the dig-out rule add to the number of steps needed for a particular variable occurrence. Let's say that the *quantifier depth* of a variable occurrence v^o is the number of nested quantifiers that have v^o in its scope (the quantifier depths of the variables in $P(x_1, x_2)$ in (26) is then 2). Let say that the *binding depth* of a variable occurrence v^o is the number of nested quantifiers that have v^o in its scope *in the subformula whose outermost quantifier binds v^o* (the binding depths of the variables in $P(x_1, x_2)$ in (26) are then 2 and 1, respectively). The number of required applications of the dig-out rule for a particular variable occurrence is then identical to its binding depth, as is easily seen.

The total number of added computation steps is simply the sum of the binding depths of all variable occurrences, minus the number of variable occurrences. If we allow for vacuous quantification, it may hold for each variable occurrence that its binding depth equals its quantifier depth. Hence, the maximal number of dig-out rule applications for a sentence with n quantifiers and m variable occurrences is $m \times (n - 1)$. If we let the syntactic size of a quantifier ' $\exists x_i$ ' be 1, then for a sentence of size $2k + 1$, we will maximize the given product by letting $m = n = k$, which is a sentence of the form

$$(33) \quad \exists x_{i_1}, \dots, \exists x_{i_k} (P(x_{i_1}^1, \dots, x_{i_1}^k))$$

where all quantifiers except the first are vacuous, and the superscripts number the occurrences of the variable x_{i_1} .

This means in turn that the maximal number of rewriting steps for a sentence of size $2k + 1$ is exactly $k^2 + k + 1$. Hence, taking the dig-out rule for quantifiers into

account, we have for n odd:

$$C(n) = \left(\frac{n-1}{2}\right)^2 + \frac{n-1}{2} + 1 \leq n^2.$$

The complexity of the SAT rule system is therefore not linear, but *quadratic*. It is not linear, since for any constant i , there is a j such that for all $k \geq j$,

$$C(k) > ik$$

(let $j = 4i + 1$).

Clearly, an interpretation problem with quadratic time complexity is tractable. It is low, but it is not minimal in the sense of section 4.¹⁹

The question whether we can do better with another treatment of a first-order language is a complex one that cannot be fully investigated in this paper.^{20 21}

If we had chosen a set of rules that were closer to the original T-theory format, where the rewrite terms are biconditionals, where we use a T-predicate, with more standard reference axiom rules and substitution rules, time complexity would increase somewhat, although not dramatically.²²

¹⁹English sentences in ordinary use seldom have more than two nested quantifiers, and hardly ever more than three. Because of this, the actual average complexity, by the T-theoretic method, is only slightly above minimal. The further interesting question is whether the quadratic worst-case complexity actually *explains* why the nesting of quantifiers in ordinary English is so limited, or whether some other cognitive/computational parameter is more relevant.

²⁰In Pagin 2012 I did not take variable-binding into account. The conclusion there that polynomial compositionality has minimal complexity only holds for languages without variable-binding operators.

²¹Suppose we turn to an algebraic semantics where meanings are treated as sets of assignments. Then we will allow assignment function symbols in normal form representations, which has not been allowed here. With ‘ $\llbracket \cdot \rrbracket$ ’ as semantic function symbol, the existential quantifier rule will be

$$(i) \quad \llbracket \exists x_i(A) \rrbracket \longrightarrow \{f : \exists f'(D(f, f', x_i) \& f' \in \llbracket A \rrbracket)\}$$

With nested quantifiers in the *OL* sentence we get nested set abstraction expressions in the *ML* representation. If we do not require simplification of the *ML* representation in normal form (so that there is only one occurrence of ‘ $\{\cdot\}$ ’), time complexity can in fact remain minimal, at the cost of size-increasing (and intuitively not fully interpretational) normal forms. There appears to be a trade-off here. For instance, the use of Schönfinkel combinators to eliminate bound variables does lead to a size increase. At present, though, the trade-off thought must remain a conjecture.

²²Formal semanticists do not always care about complexity. On pp 86-87 of their 1995, Larson and Segal provide a derivation in 14 steps of a T-sentence (in their format) for the sentence

- (i) Phil ponders or Chris agrees.

7. Conclusion

We have considered only an extremely simple language, a first-order context free language. We have not taken into account the complexity of the problem of parsing the natural language sentence and regementing it into the desired format. We have also not taken into account the complexity of pragmatic processing.

The conclusion is nevertheless encouraging: T-theories represent the problem of semantic interpretation as tractable, in fact as having low complexity, even if not minimal. They can therefore be used to explain how it is possible for human speakers to understand new sentence (as intended by the speaker).

This explanation does in fact require more than showing that we can provide a recursive semantics for a language in question, i.e. a semantics in accordance with the principle (PRS) (page 3). For the time complexity of such a semantics is exponential: take an n -ary syntactic operator $\alpha(x_1, \dots, x_n)$, and define a sequence T of terms t_0, t_1, \dots , where t_0 is atomic, and $t_{k+1} = \alpha(t_{t_k}^0, \dots, t_k^n)$, where the superscripts number the occurrences. In a recursive semantics, the number of steps required for interpreting the term t_j in T , of term size roughly n^j , is greater than n^{n^j} . To allow any recursive semantics is therefore to allow semantics with intractable complexity.²³

even though the disjuncts are treated as atoms, with one axiom each. The main factor behind this increase of complexity is the use of quantification over truth values, which in turn is part of their way of giving semantic values to logical particles in a way that respects compositionality. They have e.g.

(37a) $\text{Val}(t, [{}_S \text{ ConjP}])$ iff for some z , $\text{Val}(z, S)$ and $\text{Val}(z, \text{ConjP})$.

where $\text{Val}(t, S)$ says that S has value True, and ConjP is a conjunction or disjunction phrase. (Incidentally, this introduction of bound variables brings with it the additional complexity of checking for variable identity; if we disregards that, the LS semantics for the conjunction fragment alone induces a linear complexity increase by at least a factor of about 12). They comment on the situation by saying

[...] it has often proved useful as a research strategy to prefer complex structures and complex derivations to complex principles. This is simply because the former tend to yield a more restrictive theory overall, and hence one to be preferred under the logic of the language acquisition problem (1995, 88).

This contrasts sharply with the viewpoint of the present paper as well as of Pagin 2012, where the advantage of (polynomial) compositionality is that it minimizes processing complexity.

In fact, LS here appeal to the complexity of language *acquisition*, but they do not consider whether ease of processing can be an acquisition enhancing factor, by the simple principle: select the theory that represents processing as easier.

²³Cf. Pagin 2011.

This means that arguments like Davidson's learnability argument can be slightly but crucially qualified into: "When we can regard the meaning of each sentence as a *tractable* (humanly feasible) function of a finite number of features of the sentence, we have an insight not only into what there is to be learned; we also understand how an infinite aptitude can be encompassed by finite accomplishments." And for the fragments that can be handled by them, T-theories do in fact provide this insight.

Department of Philosophy
Stockholm University

Appendix

In this appendix, we shall show that the *SAT* rewrite system for the first-order language *L* is adequate in the relevant respects: that it terminates, that it is confluent, and that it is, or can be, homophonic. The signature for *SAT* consists of an *OL* part and an *ML* part.

The vocabulary of *OL* consists of

- (LV) i) finitely many individual constants, c_1, \dots, c_{n_c}
- ii) denumerably many individual variables, x_1, x_2, \dots
- iii) for finitely many natural numbers k , a finite set P^k of k -place atomic predicates, $P_1^k \dots P_{n_k}^k$
- iv) propositional logical constants: $\neg, \&$
- v) quantifiers: \exists

The syntax of *L* is given by the usual inductive clauses. I shall use ' t, t_1, t_2 ' etc. for singular terms, either individual constants or individual variables.

- (OL) i) if t_1, \dots, t_n are singular terms and P_i^n an n -place predicate, $P_i^n(t_1, \dots, t_n)$ is a formula
- ii) if s is a formula, the $\neg(s)$ is a formula
- iii) if s and s' are formulas, then $\&(s, s')$ is a formula

- iv) if s is a formula and x_i a variable, then $\exists x_i(s)$ is a formula
- v) if x_i occurs in s and is not in the scope of any occurrence of $\exists x_i$ in s , then x_i occurs free in s ; if x_i occurs free in s , then x_i is bound by the outermost occurrence of $\exists x_i$ in $\exists x_i(s)$; if no variable in s is free, then s is a sentence.

The vocabulary of ML consists of

- (MLV) i) finitely many individual constants, a_1, \dots, a_{n_c}
- ii) denumerably many individual variables, b_{x_1}, b_{x_2}, \dots
- iii) for finitely many natural numbers k , a finite set G^k of k -place atomic predicates, $G_1^k \dots G_{n_k}^k$
- iv) propositional logical constants: *not, and*
- v) quantifiers: *some*
- vi) denumerably primitive assignment function symbols f_1, f_2, \dots
- vii) a two-place operator $[\cdot/\cdot]$
- viii) a two-place predicate *sat*
- ix) a rewrite variable for assignment function symbols, g
- x) rewrite variables for OL variables: v_1, v_2, \dots
- xi) rewrite variables for OL singular terms: y, y_1, y_2, \dots
- xii) rewrite variables for OL formulas: S, S'

The syntax of ML is as follows:

- (ML) i) Every primitive assignment function symbol is an assignment function symbol.
- ii) Where v_i is variable for OL variables, b_{x_j} an individual variable, and f_k an assignment function symbol, $f_k[b_{x_j}/v_i]$ is an assignment function symbol.
- iii) An expression t is an ML singular term iff t is an ML individual constant, or t is an ML individual variable, or t is an expression $f(u)$, where f is an assignment function symbol and u is an OL individual variable.

- iv) If G is an n -place ML predicate and t_1, \dots, t_n are ML singular terms, $G(t_1, \dots, t_n)$ is an ML formula.
- v) If s is an OL formula and f an assignment function symbol, then $\text{sat}(f, s)$ is an ML formula.
- vi) If s is a ML formula, then $\text{not}(s)$ is an ML formula.
- vii) If s and s' are ML formulas, then $\text{and}(s, s')$ is an ML formula.
- viii) If s is a ML formula and b_{x_i} is a ML individual variable, then $\text{some } b_{x_i}(s)$ is a ML formula.

The set of rules R_{SAT} of SAT consists of:

- (R_{SAT}) i) for each constant c_i in L there is a *constant rule*: $g(c_i) \longrightarrow a_i$
- ii) $g[b_v/v](v) \longrightarrow b_v$
- iii) $g[b_{v_i}/v_i](v_j) \longrightarrow g(v_j)$, if $i \neq j$
- iv) for each atomic predicate P_j^i a *predicate rule*:
 $\text{sat}(g, P_j^i(y_1, \dots, y_i)) \longrightarrow G_j^i(g(y_1), \dots, g(y_i))$
- v) $\text{sat}(g, \neg(S)) \longrightarrow \text{not}(\text{sat}(g, S))$
- vi) $\text{sat}(g, \&(S, S')) \longrightarrow \text{and}(\text{sat}(g, S), \text{sat}(g, S'))$
- vii) $\text{sat}(g, \exists v(S)) \longrightarrow \text{some } b_v(\text{sat}(g[b_v/v], S))$

We do not use the *truth to satisfaction* rule, which is superfluous.

We can regard SAT in two ways. Either we treat it as a *many-sorted* system, where the rewrite variables only take instances of the designated type.²⁴ Then the rules will be restricted to well-formed ML terms. Or else, we treat the different kinds of variables as merely a notational convenience, and let them take the same instances. In this case we simply stipulate that the set of terms to which the rules apply is restricted to terms that are built from well-formed OL expressions. The result will be the same.

Definition 1. The *size* $\sigma(u)$ of a rewrite term u is define as follows:

- i) for any ML constant a_c , $\sigma(a_c) = 0$
- ii) for any ML variable b_x , $\sigma(b_x) = 0$

²⁴See Terese 2003, 77-78.

- iii) for a *OL* singular term t , $\sigma(t) = 1$
- iv) for any assignment function symbol f and any *OL* variable x ,

$$\sigma(f(x)) = \sigma(x)$$
- v) for any n -ary *L* atomic predicate P , and any *OL* singular terms t_1, \dots, t_n ,

$$\sigma(P(t_1, \dots, t_n)) = 1 + \sum_{i=1}^n \sigma(t_i)$$
- vi) for any n -ary *ML* atomic predicate G , and any *ML* singular terms u_1, \dots, u_n ,

$$\sigma(G(u_1, \dots, u_n)) = \sum_{i=1}^n \sigma(u_i)$$
- vii) for any *OL* formula s , $\sigma(\neg(s)) = 1 + \sigma(s)$
- viii) for any *OL* formulas s, s' , $\sigma(\&(s, s')) = 1 + \sigma(s) + \sigma(s')$
- ix) for any *OL* formula s , and any *OL* variable x , $\sigma(\exists x(s)) = 1 + \sigma(s)$
- x) for any *OL* formula s and any assignment function symbol f , $\sigma(\text{sat}(f, s)) = \sigma(s)$
- xi) for any *OL* formulas s and any assignment function symbol f ,

$$\sigma(\text{not}(\text{sat}(f, s))) = \sigma(\text{sat}(f, s))$$
- xii) for any *OL* formulas s, s' and any assignment function expression f ,

$$\sigma(\text{and}(\text{sat}(f, s), \text{sat}(f, s'))) = \sigma(\text{sat}(f, s)) + \sigma(\text{sat}(f, s'))$$
- xiii) for any *OL* formula s , any assignment expression f and *OL* variable x ,

$$\sigma(\text{some } b_x \text{sat}(f[b_x/x], s)) = \sigma(s)$$

Secondly, we define the *length of an assignment symbol* and of a term:

Definition 2. The *length* $l(f)$ of an assignment symbol f is defined as follows:

- i) $l(f) = 0$, if f is a simple assignment symbol.
- ii) $l(f[b_v/v]) = 1 + l(f)$

Definition 3. The *length* $\lambda(t)$ of a term t is the sum of the lengths of occurrences of assignment symbols in t .

On the basis of these two measures we define a lexicographic ordering of terms:

Definition 4. $t < t'$ iff

- a) $\sigma(t) < \sigma(t')$, or
- b) $\sigma(t) = \sigma(t')$ and $\lambda(t) < \lambda(t')$

We can now make the following observation:

Fact 1. For any *ML* terms u, u' , if u immediately reduces to u' (i.e. by means of a single application of a rule of R_{SAT}), then $u' < u$.

Proof. Proof by cases, where each case is immediate from the rules of R_{SAT} and Definitions 1, 2, 3, and 4. For any rule r except (R_{SAT} iii), if u reduces to u' by means of an application of r , then $\sigma(u) = 1 + \sigma(u')$. For instance, an application of (R_{SAT} vii) lowers the size by 1 according to (ix) and (xiii). If u reduces to u' by means of an application of rule (R_{SAT} iii), then $\lambda(u) = 1 + \lambda(u')$, while $\sigma(u) = \sigma(u')$. \square

Fact 2. Every R_{SAT} reduction of a term u terminates when u has been reduced to a term u' such that $\sigma(u') = \lambda(u') = 0$.

Proof. By inspecting R_{SAT} and Definition 1, we see that any instance u of the lhs of a rule in R_{SAT} has positive size or length: $\sigma(u) > 0$ or $\lambda(u) > 0$. We also note that for any u , if u instantiates the rhs of any rule of R_{SAT} and $\sigma(u) > 0$ or $\lambda(u) > 0$, then u , or some subterm u' of u instantiates the lhs of some rule of R_{SAT} , and hence u can be further reduced. By Fact 1, every rule application lowers the size or else lowers the length without changing the size. Hence, the reduction terminates when both the size and the length has reached 0. \square

Fact 3. R_{SAT} is *confluent*: if u reduces to distinct terms u_1 and u_2 , then there is a term u' such that both u_1 and u_2 reduces to u' .

Proof. Confluence follows by the *Critical Pair Theorem* and the fact that R_{SAT} does not have overlapping rules, i.e. pair of rules that apply to the same subterm (cf. Baader and Nipkow 1998, 139-40). \square

Fact 4. For every formula s of *OL* and any assignment functor f , the term $\text{sat}(f, s)$ reduces to a term u that is syntactically isomorphic to s .

Proof. Proof by induction over expression complexity in *OL*. For a singular term t , $f(t)$ reduces to a singular term: to a constant a_c if t is a constant c , and to a variable b_x if t is a variable x . For an atomic formula $P(t_1, \dots, t_n)$, $\text{sat}(f, P(t_1, \dots, t_n))$ reduces to an atomic formula $G(f(t_1), \dots, f(t_n))$, isomorphic to $P(t_1, \dots, t_n)$ if the terms so reduce. Analogously for the connectives.

For the quantifier we need to ensure that an occurrence of a bound variable in the L sentence corresponds to an occurrence of a bound variable in the ML sentence. But this is ensured by the fact that the choice of ML variable is determined by the corresponding OL variable, both in the binding occurrence, in ‘ $\exists x$ ’ and in the bound occurrences in the scope of ‘ $\exists x$ ’. \square

Fact 5. When ML is an extension of L , the interpretation defined by a system analogous to R_{SAT} induces a homophonic T-theory.

Proof. Define a system R'_L that is like R_{SAT} except that every term a_c is the term c , every variable b_x is replaced by x , and every predicate G_j^i is the predicate P_j^i . The ML logical vocabulary in R'_L is the same as the logical vocabulary in OL . Then, by Facts 3 and 4, the normal form of a term $\text{sat}(f, s)$ is s itself. \square

Remark: Fact 5 is the closest we can get to proving formally that the system R_{SAT} is interpretationally adequate.

References

- Baader, Franz and Tobias Nipkow (1998). *Term Rewriting and All That*. Cambridge: Cambridge University Press.
- Davidson, Donald (1965). ‘Theories of Meaning and Learnable Languages’. In: *Logic, Methodology and Philosophy of Science II*. Ed. by Y. Bar-Hillel. Amsterdam: North-Holland. Reprinted in Davidson 1984, 1-15.
- (1967). ‘Truth and Meaning’. In: *Synthese* 17, pp. 304–23. Reprinted in Davidson 1984, 17-36.
- (1973). ‘Radical Interpretation’. In: *Dialectica* 27. Reprinted in Davidson 1984, 125-39. Page references to the reprint, pp. 313–28.
- (1974). ‘Belief and the basis of meaning’. In: *Synthese* 27. Reprinted in Davidson 1984, 141-54. Page references to the reprint, pp. 329–23.
- (1984). *Inquiries into Truth and Interpretation*. Oxford: Clarendon Press.

- Davidson, Donald (1986). 'A nice derangement of epitaphs'. In: *Truth and Interpretation: Perspectives on the Philosophy of Donald Davidson*. Ed. by Ernest LePore. Reprinted in Davidson 2005, 89-107. Page references to the reprint. Basil Blackwell.
- (2005). *Truth, Language, and History*. Oxford: Clarendon Press.
- Fodor, Jerry and Jerrold Katz (1964). 'The Structure of a Semantic Theory'. In: *The Structure of Language*. Ed. by Jerry Fodor and Jerrold Katz. Englewood Cliffs, NJ.: Prentice-Hall, pp. 479–518.
- Grandy, Richard (1990). 'Understanding and the Principle of Compositionality'. In: *Philosophical Perspectives* 4, pp. 557–72.
- Henkin, L., J.D. Monk, and Alfred Tarski (1971). *Cylindrical Algebras, Part I*. Vol. 64. Studies in Logic and the Foundations of Mathematics. Amsterdam: North-Holland.
- Janssen, T.M.V. (1997). 'Compositionality'. In: *Handbook of Logic and Language*. Ed. by Johan van Benthem and Alice ter Meulen. Amsterdam: Elsevier, pp. 417–73.
- Larson, Richard and Gabriel Segal (1995). *Knowledge of Meaning. An Introduction to Semantic Theory*. Cambridge, Mass.: MIT Press.
- Lepore, Ernest and Kirk Ludwig (2005). *Donald Davidson. Meaning, Truth, Language, and Reality*. Oxford: Oxford University Press.
- (2007). *Donald Davidson's Truth-Theoretic Semantics*. Oxford: Oxford University Press.
- Pagin, Peter (2011). 'Compositionality, Computability, and Complexity'. Draft.
- (2012). 'Communication and the Complexity of Semantics'. In: *The Oxford Handbook of Compositionality*. Ed. by Wolfram Hinzen, Edouard Machery, and Markus Werning. Oxford University Press, pp. 510–29.
- Putnam, Hilary (1975). 'Do True Assertions Correspond to Reality?' In: *Mind, Language and Reality. Philosophical Papers vol. 2*. Cambridge: Cambridge University Press, pp. 70–84.
- Terese (2003). *Term Rewriting Systems*. Ed. by Mark Bezem, Jan Willem Klop, and de Vrijer Roel. Vol. 55. Cambridge Tracts in Theoretical Computer Science. Cambridge: Cambridge University Press.

Wiggins, David (1980). 'Most' and All': Some Comments on A Familiar Programme'.
In: *Reference, Truth and Reality: Essays on the Philosophy of Language*. Ed. by
Mark Platts. London: Routledge and Kegan Paul, pp. 318–46.