# Computability As A Test On Linguistics Theories

*Tony Veale*

**Abstract.** Scientific theories are more than purely formal constructs, but linguistic artefacts that often rely on the rhetorical qualities of language to give their claims additional resonance and argumentative force. This reliance of theory upon language is even greater in those theoretical domains whose main concern is language itself, leading to a sometimes convenient blurring of content and form. Though Cognitive Linguistics has consistently revealed metaphor to be a fundamental building block in the development of complex conceptualizations, Cognitive Linguistic theories often exploit metaphor as an allusive place-holder when more formal clarity is demanded. Nonetheless, in this paper we argue that one such metaphor – the MIND-AS-COMPUTER metaphor that underlies the enterprises of Artificial Intelligence and Cognitive Science – can yield precisely the kind of formal clarity that is required by the most suggestive or radical of theories in Cognitive Linguistics. An exploration of how this metaphor informs the computational realization of cognitive theories will allow us to illuminate the often wide gap that exists between the descriptive suggestiveness of a linguistic theory and its actual computational sufficiency.

*Keywords*: Computability, Tractability, Mind As Computer, Verbal Humour, Blending, Radical Construction Grammar

## 1. Introduction

In seeking mechanical insight into the workings of the mind, researchers of different eras have, unsurprisingly, drawn inspiration from the dominant artefacts of the age. Consequently, the mind have variously been described as a wax tablet, a book, a library of books, a grain mill, a clock, a steam engine, an internal combustion engine and even a telephone exchange. This evolution of MIND AS MECHANISM metaphors has reached it zenith with the metaphor that underlies modern Cognitive Science, that of MIND AS COMPUTER (e.g., see Gardner, 1985; Jackendoff, 1987). But as useful as this metaphor has proven in Cognitive Science and its sibling fields (such as Artificial Intelligence), one might be forgiven for some scepticism as to its ultimate suitability as a model of mind. As older metaphors, such as the mind as grain mill, have passed their use-by date, is it not likely that the mind as computer metaphor will one day be superseded by another, more apropos model of cognition?

A strong rebuttal to this scepticism is to be found in the extreme generality of the modern computer, which is much more than a mere calculating machine but a universal computing device in itself. This universality implies that any realizable computing device can itself be simulated by a general purpose computer. Such a computer is a physical instantiation of a Turing machine, an abstract mathematical device on which every function that can naturally be considered computable *can* be computed (see Turing, 1936). As such, if one is inclined to see the actions of the mind in terms of information processing, as Cognitive Science demonstrates we must, then the mind as computer metaphor has no expiry date. By information processing we mean, of course, more than the manipulation of electronic data, but include also the construction and manipulation of mental representations that have computational correlates in the form of strings, frames, records, trees, graphs and networks (see Veale and O'Donoghue, 2000). This general belief in the mind as computer metaphor has been dubbed the *hypothesis of computational sufficiency* by Ray Jackendoff (1987:24). who makes the following bold claim "Every phenomenological distinction is caused by/supported by/projected from a corresponding computational distinction."

Jackendoff exhorts us to reason at the level of computational rather than phenomenological (or rhetorical) distinctions, for only the former is conducive to a explanatory account of the phenomenon under consideration. As an example, consider an analogical theory of car mechanics: a car travelling at 59 miles per hour is within the speed limit, but a slight acceleration to 61 miles per hour may cause it to break the law. This discontinuity at the social level (between legal and illegal) is not in the least reflected at the mechanical level of the car, which maintains the same engine throughout. To understand the engine then, we must focus only on those distinctions that directly arise from its architectural design and operational limits.

Given this perceived dependence of cognition on computation, it follows that the theoretical foundations of Computer Science, which place fundamental limitations on the actions of universal computing devices, should be of equal significance to any theory that views the workings of cognition from an information processing perspective (e.g. see Veale *et al.*, 1996), and especially so to any linguistic theory that posits complex abstract machinery to capture the intricacies of natural language phenomena. Consider Chomsky's (1957) transformational grammar (TG), which models language as a multi-strata phenomenon in which complex transformations mediate between an unseen deep-structure representation and a visible surface form. The fact that TG approaches to language are often misunderstood as theories of language generation is due only in part to the ambiguity inherent in Chomsky's notion of linguistic *generativity*, for the fact is, TG-based models are more computationally felicitous when viewed as theories of language generation. The burdensome complexity of TG for parsing purposes is demonstrated, for example, by Plath's (1973) transformational parser, which requires an additional layer of transformational inverses to apply TG in reverse to an initial surface structure, and requires a complex sanity-checking mechanism (based on the forward application of TG transformations to the derived deep-structure) to additionally ensure that these inverses are always applied in a truly reversible manner. Just as it much easier to break a teacup than to properly re-assemble its broken parts, some theories can be seen as inherently directional from a computational perspective.

In this paper we shall explore the pertinence of a computational perspective to theories in Cognitive Linguistics by focussing on three specific theories of language use. The first two model creative linguistic phenomena, while the third represents a rather extreme example of the construction-based approach to language study. Since creativity is an area of study where one intuitively expects to find the least amount of formal substance, these first two models – Attardo and Raskin's *General Theory of Verbal Humour* or GTVH (1991) and Fauconnier and Turner's theory of *Conceptual Integration Networks* (1998, 2002) – therefore allow us to explore the often wide gap between descriptive suggestiveness and true computational sufficiency. This notion of sufficiency cuts directly to the computational core of a theory, and allows us to consider distinct theories for distinct phenomena as comparable if they ultimately hinge upon the same computational distinctions. This notion of computational equivalence – a corollary of sorts of computational sufficiency – allows us to consider the computational feasibility of Radical Construction Grammar, not in terms of its own software realization but in terms of a comparable computationally-precise theory of language translation that has been implemented on a large scale.

The rest of the paper assumes the following outline: in section two we provide capsule descriptions of the three linguistic theories that fuel the arguments to come; in section three we then consider three desirable features of scientific theories – falsifiability, explanatory power and formal specificity – before considering, in section four, the desirable properties that one additionally seeks from a computational perspective – computational specificity, efficiency and tractability. In section five we explore theoretical and engineering aspects of computational feasibility. We conclude in section six with a discussion of theories as linguistic artefacts, showing how a computational perspective can look beyond the rhetorical layers of a theory to its core.

## 2. Three Models of Language Processing

Before considering the role of computationalism in linguistic theorization, we first consider a broad overview of the three particular models on which the substance of our arguments will be focussed.

### 2.1. The General Theory of Verbal Humour

Attardo and Raskin's GTVH, which currently dominates humour research, is a juxtapositional theory of humour that is an modular reworking of Raskin's (1985) Semantic Script Theory of Humour (or SSTH). Like the SSTH, the GTVH views a joke as a narrative that is compatible with multiple scripts, one of which will at first appear primary until the punch-line contrives a *incongruity* that must be resolved (see Ritchie, 2003; Veale, 2004). Resolution is achieved, either partially or fully, by a special logical mechanism that analyses the nature of the mismatch between the primary script and the text, before switching the thrust of interpretation from this script to another. For instance, it is suggested that an LM called *false-analogy* is central to jokes whose humour derives from ill-judged comparisons, as in the old joke where a mad scientist builds a rocket to the sun but plans to embark at night to avoid being cremated. Here a false analogy is created between the sun and a light-bulb, suggesting that when the sun is not shining it is not "turned on", and hence, not hot. Different LMs may be employed in different jokes, bringing a distinctive logical flavour to each. More recently, Attardo *et al.* (2002) enumerate a variety of different logical mechanisms (27 in all) and offer a new, graph-theoretic account of script representation that now sees scripts as arbitrarily complex symbolic structures to which juxtapositional processes like sub-graph isomorphism can be applied. GTVH scripts can now accommodate not just the semantic structure of events, but the phonological structure of words. For example, Attardo *et al.* (ibid) argue that the pun in the mathematical book title "The joy of sets" is resolved by structure-mapping two phonological "script" graphs, *[s, e, tz]* and *[s, e, ks]*.

### 2.2. Conceptual Integration Networks, or Blending Theory

Our second model of interest is Fauconnier and Turner's (1998; 2002) theory of conceptual integration networks, more popularly called "blending theory". Like the GTVH, blending theory offers a juxtapositional account of creative language use, one in which multiple mental spaces (called "inputs") are integrated to generate a new mental space, the "blend space", that contains a selective projection of the inputs which is formed subject to a variety of interacting optimality principles. This new space, which is connected to, but independent from, the original input spaces, is invoked to explain the emergent properties of many compositional structures, from metaphors to noun compounds to jokes. When relations between elements of different input spaces are projected into a blended space, they may become compressed through a process of metonymic tightening. Thus, a similarity relation outside the blended space may become an identity relation within the blended space. Coulson (2000) has used blending theory to offer a theory of humour, called "frame shifting", that is in many respects similar to the "script-switching" view of the GTVH. Though some attempts have been made to realize blending theory in computational form (e.g., see Veale and O'Donoghue, 2000), such implementations are necessarily incomplete and require a loose, almost metaphorical reading of the theory's principal mechanisms.

## 2.3. Radical Construction Grammar

Our third model is Croft's (2001) Radical Construction Grammar, which, as its name suggests, is a theory whose simplifying assumptions pose a radical challenge to key elements of linguistic orthodoxy. Like other construction grammars, RCG gives a central position to the role of the construction, a mapping between form and meaning that can be almost entirely substantive (as in the case of idioms) or entirely parameterized (as in the case of grammatical schemata). RCG thus obliterates the traditional distinction between lexicon and grammar, as constructions may lie anywhere along a lexico-syntactic continuum. As in other variants of construction grammar (e.g., see Goldberg, 2003), the purpose of constructions in RCG is to break down the traditional barrier between form and meaning by directly motivating issues of form in semantic and pragmatic terms. But RCG is radical in suggesting that conventional formal categories like S, V and NP are not universal building blocks for grammar rules or constructions. Indeed, RCG is radical to the point of claiming that such categories are not even language-specific, but are merely specific to the constructions that contain them. This pitches conventional linguistic orthodoxy on its head, for since these categories serve a local rather than general role with no absolute meaning, one cannot derive the meaning of a construction from the categories it contains; rather, RCG claims that the meaning of these categories is instead inferred from the constructions in which they appear.

## 3. Desirable Properties of Scientific Theories

To be truly scientific, a theory should make sufficiently strong claims that are open to rebuttal by experimentation or direct observation. This principle, most famously reduced to the single term "falsifiability" (e.g., see Popper, 1959), is tightly woven into the practice of modern day linguistics wherever the inner processes of language impinge on superficial form (consider the linguist's frequent appeal to verification via native speaker intuitions). In Cognitive Linguistics, with its emphasis on conceptual structure, such opportunities for surface-level falsification can be altogether less frequent. Bell (2002), for example, has questioned whether Fauconnier and Turner's (1998;2002) theory of conceptual blending exposes enough of its workings to external observation at the textual level to be falsifiable. Falsification, then, is just one of several properties one should desire of a linguistic theory, and in lieu of observation-based falsification, one must look to other indicators of a theory's soundness. The first of these is explanatory power, which separates theories with a post-hoc descriptive utility from those that exhibit genuine causal insight into the workings of a particular mechanism. The second is formal specificity, which separates those theories whose attractiveness is largely rhetorical from those that exhibit an unadorned logical clarity.

### 3.1. Explanatory Power

In linguistic terms, the difference between an explanatory theory and a descriptive theory is much like the difference between a metaphor and a simile: a good metaphor reveals the deep causal structure beneath a domain (e.g., "the Earth is an electron buzzing around its nucleus, the sun") while a good simile is only superficially revealing (e.g., "the Earth is like a football"). Are the GTVH and blending theories merely similes for the superficial workings of our creative use of words, or do they capture real cognitive mechanisms at work? The GTVH does not explain why a text that is compatible with two overlapping but incongruous scripts should be considered humorous, but merely claims that successful jokes appear to exhibit this property. In specifying 27 different logical mechanisms of humour (and leaving the door open for more to come), Attardo *et al.* (2002) appear to be engaged in what the physicist Ernest Rutherford dismissed as a form of science more akin to "stamp-collecting" than physics. Likewise, in attempting to "explain too much" (see Gibbs, 2000; Bell, 2002), one can argue that conceptual blending theory trivializes its subject matter: is every compositional mental structure to be seen as a blend (albeit, in many cases, a bad or sub-optimal blend)? Are we to take as a blend any conceptual structure whose composition can be described by blending theory, or should we require a more restrictive definition? Unfortunately, there seems little in the formal apparatus of the theory itself to satisfactorily answer these questions.

A descriptive theory can afford to be agnostic as to the specific processes that yield the description for a given phenomenon, even when competing accounts of these processes are available. An explanatory theory, in contrast, should commit to just one account, and moreover, should offer reasons as to why this account is not simply the "official" account, but a truly superior account in light of the available facts and data. Conceptual blending is a mechanism that, when run *forward* from its inputs, attempts to explain how a complex conceptual product is created, and when run *backward* from an integrated product, attempts to explain how this product is comprehended. Since conceptual blending provides a very detailed account of how creative products are comprehended, it is tempting to believe that the theory also explains how such products are created. However, one can easily imagine other, simpler and more computationally felicitous accounts, of how such conceptual products are created.

Consider the complex ideas "drive-by shooting" and "date-rape", two apparently archetypal products of conceptual blending in action. The linguistic form of "date-rape" directs us to comprehend the concept as an integration of two scenarios ("date" and "rape") that necessitates a tight mapping of participants (e.g., suitor = rapist, target-of-affection = victim), as does the linguistic form of "drive-by-shooting" (e.g., driver/passenger = shooter, pedestrian = victim). Each also gives rise to emergent inferences: drive-by shootings are faster but less accurate forms of attack, and typically involve higher rates of collateral damage; date-rape is typically harder to prosecute, and may not even be acknowledged as rape by its perpetrator. From a generation perspective, however, it is surely more intuitive to think of each not as an explicit integration of distinct ideas, but as a simple deviation from a single prototype. Date-rape represents a subversion of the prototypical dating scenario (or script) in which consent for sexual activity has been edited out, thereby resulting in a scenario that more resembles the rape scenario. Likewise, a drive-by shooting can be seen as an edited version of a prototypical "hit", in which the attacker does not leave (or even stop) his car. This single-input subversion account can also by used to explain the mechanics of many jokes. For instance, an industrial-drowning script can be subverted into a form that resembles (and thus recalls) a swimming-pool script, either by editing-out the tragic conclusion (though this is unlikely to result in humour) or by editing-*in* some elements that suggest the tragic event was a pleasurable one[1]. Each of these examples demonstrate that complex products with emergent features can be constructed from a single input structure by assuming a simple form of internal editing. We should be wary then of cognitive just-so stories: we cannot conclude from the fact that “date-rape” and “drive-by-shooting” (or, for that matter, “house-boat”, “sofa-bed”, or any number of classic blends and jokes) *can* be described as an integration of multiple input structures, that they *should* be so described. Ultimately, explanatory force arises not from apparent possibility, but from apparent necessity. We consider how computational concerns might provide this necessity in section 4.2.

### 3.2. Formal Specificity

Though formalization can seem a dry affair, it is a necessary step if a theory is to achieve unambiguous clarity, particularly so if the entities to which it is ontologically committed are to be specified in a way that neither relies on metaphor or unarticulated intuition. A theory of creativity, for instance, must be capable of formally separating those processes and artefacts that are deserving of the label “creative” from those that are not. Similarly, a theory of humour must be capable of unambiguously defining a joke. As we have argued, a theory of conceptual blending must likewise be capable of defining what kinds of construct constitute a blend, and more importantly, what constructs do not constitute a blend. The classical approach to such formalization is the use of necessary and sufficient feature sets. For instance, Raskin (1985) claims that two features are individually necessary and jointly sufficient for a text to constitute a joke: the joke must be partially compatible with at least two scripts (the criterion of “script overlap”) and these scripts must be opposed to each other in a particular way (the criterion of “script opposition”). Neither script overlap alone (e.g., as found between dentist and doctor scripts) or script opposition alone (e.g., as found in baptism/life and funeral/death scripts) is sufficient in this view to produce humour.

Necessary and sufficient features have long been rejected by cognitivists as a model of human category structure, as they lead to brittle and simplistic structures that are easily invalidated by real world examples. To a researcher in Cognitive Linguistics, it may therefore seem unrealistic to build a theory around such features, since theories are categories too and we would like our theories, like our categories, to be robust in the face of real world data. For instance, it is commonly held outside Cognitive Linguistics that semantic anomaly is a necessary feature of metaphors (e.g., see Fass, 1988). However, this claim is easily refuted by metaphors like “my lawyer is my bodyguard” and “my mechanic is a magician”. Likewise, it is commonly held, even by cognitivists, that incongruity is a necessary feature of humour (see Ritchie, 2003 for a review), while Veale (2004) argues that incongruity is merely a by-product of a listener's desire to opportunistically seek humour in a text when socially licensed to do so. Nonetheless, though the anomaly theory is unattractive as a theory of metaphor on aesthetic grounds, inasmuch as it tends to over-simplify the phenomenon, it's bold claims are very attractive in Popperian terms. Likewise, incongruity theories of humour are unattractive to some researchers (such as Veale, 2004) for similar reasons, inasmuch as they deny the listener a collaborative role in humour construction, yet they too are attractive for so readily courting falsification. From a scientific perspective then, a lack of robustness in the face of real data can be seen as a desirable by-product of the bold claims made by necessary and sufficient feature definitions, since bold claims are more easily falsified than weak ones. This scientific perspective has a valuable engineering corollary: since necessity and sufficiency more easily translate into computational form (e.g., as an exhaustive collection of set-theoretic conditionals) than the equivalent system of potentially incomplete family resemblances, a theory formalized in this classical way can automatically be applied to a much larger corpus of potentially falsifying data.

## 4. Computational Realization of Scientific Theories

From a computational perspective, two further properties are desirable in a theory. Firstly, the theory must be specific enough to allow a computational realization to be constructed, which in turn requires that the theory is shorn of any linguistic ambiguity or vagueness, either by the theorist (which is the most desirable case) or, ultimately, by the programmer (via the much less desirable case of theoretical "interpretation"). Secondly, this computational implementation must be tractable, which is to say, it must exhibit desirable run-time performance that is in line with human performance on the same inputs. We now consider each of these properties in greater depth.

### 4.1. Computational Specificity

Once sufficiently formalized, a theory can be recast in a computational form. This transformation requires that the entities of concern to the theory are sufficiently defined as to suggest adequate data representations for a computer to operate upon, and that the processes of concern to the theory are sufficiently defined (in terms of chronology, inputs and outputs) to be expressed algorithmically. There are good engineering reasons for realizing a cognitive theory computationally, not least the A.I. goal of imbuing software systems with some semblance of human intelligence. But there are strong scientific reasons also, not least the opportunity for large scale automated testing of the theory that a software realization would enable. For instance, a semantic theory for the interpretation of noun-noun phrases could, if implemented in software, be applied to all of the noun-noun phrases in a machine-readable dictionary, or better yet, to all of the noun-noun phrases in WordNet, a large-scale hierarchical database of English terms and their meanings (see Miller, 1995). Suppose such a theory makes the simplifying claim that the meaning of a non-idiomatic noun-noun compound is only rarely a specialization of the modifier term (e.g., a "rat-*catcher*" is a kind of person, not a kind of rat). A large scale analysis of WordNet entries would reveal that the exceptions to this view – such as "sofa-bed", "boy-scout" and "lady-friend" – are simply too numerous to discount as rare occurrences.

In fact, the very act of computationally implementing a cognitive theory is itself an extreme test of the formal specificity of the theory and the extent to which it can support itself without the aid of human interpretation. For inasmuch as computers lack the ability to process metaphorical or vague, suggestive language, they are immune to the seductive qualities of language that can otherwise bewitch humans. Computational specificity requires that every step and every data element of a process must be explicitly defined, so to the extent that a software realization requires a programmer to make certain guesses or fudges – e.g., by using random selection of candidates when a theory simply assumes that the best selection is made in a given context – then the underlying theory must also be seen to be damagingly fudged.

Computational specificity also requires a theory to explicitly quantify the thresholds that implicitly govern its application. For instance, how much inter-script similarity should the GTVH consider a sufficient overlap to support humour? How many optimality constraints can a juxtaposition of two mental spaces violate before the juxtaposition is discarded as a failed blend (as opposed to, say, a poor blend). While such questions can go largely unanswered at the pre-computational level, they cannot remain so in any software realization that demands raw numbers, or at the very least, workable heuristics, with which to make these decisions.

## 4.2. Computational Efficiency

When multiple accounts can be offered for the same phenomenon, how should one choose between them? For example, how can one reliably choose one account of conceptual creativity, such as the single-input subversion account described earlier (in the context of the concepts *date-rape* and *drive-by shooting*), over another, such as the multiple-input integration account favoured by blending theory or the comparable multiple-script resolution account advanced by the GTVH? Computationalism does not provide a complete answer, but it can decisively boost the explanatory status of a given account by showing it to be more efficient than its competitors. This argument can be seen as a computational extension of Occam's razor, which might be stated thus: computational entities, such as representations, constraints and processes, should not by multiplied without cause, so that theories that presuppose the least number of such entities are to be preferred to the extent that they can account for the same data. This makes good explanatory sense, for one because it reflects our intuition that the brain, while not always optimal, is nonetheless thrifty in its use of valuable processing resources, and secondly (though no less important) because it requires a theorist to strongly motivate the use of entities which are shown to be superfluous in a simpler competing account. In fact, this perspective often reveals such entities to be post-hoc rationalizations whose only purpose is to "save" a theory from under- or over-generation (e.g., see Bell, 2002).

The multiple-input account offered by blending theory is inefficient as a theory of creative conceptualization not because of the need to computationally process these inputs (though this is a non-trivial factor, as we shall discuss in section 5), but because of the computational effort needed to actually populate these input spaces. Though blending theory has been criticized for attempting to explain too much (Bell, 2002), in this key respect it does not attempt to explain enough. Despite the fact that blending theory is typically employed to describe creative linguistic activities, from jokes to poetic allusions to brainteasers and clever advertisements, the creativity *per se* is assumed to happen off stage, before blending even begins, for it is the idea of the combination, rather than its principled execution, that constitutes the creative insight. Consider a gastronomic example: it is the decision to combine duck with orange, rather than the act of executing the combination (e.g., by covering the duck with orange sauce) that exemplifies the culinary creativity of *duck a l'orange*. For the single-input subversion account, the relevant search space is defined as the set of possible edits of the given input structure, which can be explored in a highly constrained and controlled manner (such as hill-climbing). However, since it is not at all clear that the optimality principles that guide the construction of a blended space can be applied in reverse to determine the ideal contents of the input spaces, one must thus assume that a computational realization of blending theory must engage in a broad-horizon search of all possible partner concepts if it is to populate these spaces. Compare this situation with that faced by the GTVH, whose creative search space is confined to just those script pairings that exhibit a sharing of structure and some explicit opposition of elements. The relative computational inefficiency of blending theory, born of an inherent computational insufficiency, suggests that the theory offers a descriptive account of creativity that is not yet a genuinely explanatory one.

### 4.3. Computational Tractability

Even if sufficiently formalized to be computationally realized, a cognitive theory meets one final hurdle. It is well known in Computer Science that problems that are solvable in principle (and thus realizable in software) may nonetheless prove unsolvable in practical terms. The distinction is akin to that between competence and performance that is commonly made in linguistics (see Chomsky, 1957): computational specificity reflects the ability to encode a theory as abstract software (competence), while computational tractability reflects the ability to run this software on a specific instantiation of a universal computing device and obtain the desired outputs in a reasonable amount of time (performance). Problems are considered "intractable" if they cannot be coded in such a way as to allow them to be solved for a wide range of inputs without making unrealistic assumptions about time or resources. As the reader will detect from the judicious use of the words "reasonable" and "unrealistic", intractability is very much a practical notion that depends both on the patience of the user and the quantity of resources (such as memory, disk-size, etc.) that is available, though generally speaking, the problems of most interest in Computer Science are such that no earthly amount of either can make an optimal solution viable. Our point in this paper is that many of the problems of most interest to Cognitive Linguists, such as the optimal selection of a subset of objects or the mapping of two structured representations, correspond, in computational terms, to these very same problems.

From a tractability perspective, solutions (i.e., computationally realized theories) can be segmented into two broad categories. Polynomial-time solutions are those whose time complexity can be given as a formula of variables raised to constant powers; for instance, a naïve sorting program for a list of $n$ elements may require $n^2$ steps and take a corresponding amount of time, denoted $O(n^2)$, to finish. In contrast, exponential-time solutions are those whose time complexity is given by a formula of terms raised to variable powers; for instance, a naïve algorithm for choosing an optimal selection from $n$ elements may require a consideration of $2^n$ different combinations, and so take $O(2^n)$ time to finish (see Garey and Johnson, 1979).

Problems that can be solved with a polynomial-time solution belong to a class of problems that computer scientists denote as P. For example, the class P includes many theories of syntactic processing via phrase-structure rules, for in general, a grammar can be used to validate a sentence of n words in $n^3$ steps or less. However, most of the interesting problems (and thus theories) in Cognitive Science and Artificial Intelligence appear to require an exponential-time solution, at least if one seeks an optimal solution. Many such problems could be solved in polynomial time if realized on a computer capable of pursuing an unlimited number of processing pathways in parallel, so that it could always execute both branches of every conditional statement simultaneously. Such a non-deterministic machine is of course physically impossible, but theoretically useful nonetheless. Computer scientists denote the class of problems that can be solved in polynomial time on such a machine as NP, for non-deterministic polynomial. Though it is tempting to simply read NP as meaning "non-polynomial", the question of whether P = NP remains amongst the most important, and elusive, of any question in theoretical Computer Science.

The hardest known problems in NP are dubbed NP-Hard[2]. One such problem is the travelling salesman problem (or TSP), in which one must select the shortest tour route among a set of *n* cities (which would seem to necessitate a consideration of *n!* different tours). To prove that a problem is NP-Hard, one must reduce a known NP-Hard problem to the given problem, to demonstrate that when solving this particular problem, one is in effect solving an NP-problem like TSP in another guise. Computational complexity is thus a relative (perhaps even parasitic) notion; if one were to find an optimal polynomial-time solution for TSP, the set NP would collapse into P, such that all NP problems would be solvable in principle in polynomial time. Since this is highly unlikely to happen, complexity classes like P and NP are extremely useful to computationalists and cognitivists alike: if a cognitive theory is sufficiently specific to indicate NP-hardness, one must anticipate either limited application of the theory (to very small problem instances) or that a sub-optimal variation of the theory is implemented by the brain[3].

## 5. Computational Feasibility and Complexity

The more daring or radical a theory, the greater the challenge it can pose to our sensibilities about what is and is not feasible in practice. But once again, the substance of this challenge is to be found not in the novelty of the theory or the sparsity of its assumptions, but in the computational feasibility of the processes that it entertains. Certain commonplace abilities and processes, such as the ability to pair off like with like from two different collections, are so familiar as to suggest no computational challenge whatsoever. Nonetheless, such mapping abilities are presupposed by any conceptual output that is the product of structural matching, from the integrated spaces of metaphors and blends, to the overlapping scripts of the GTVH, and the overlapping semantic maps of radical construction grammar (RCG). Viewed in abstract terms, these abilities become recognizably more complex as one moves from the limited demands of toy examples to the scale often demanded by real-world data.

## 5.1. Feasibility by Proxy: Radical Construction Grammar

Though Croft (2001) considers RCG to be “disarmingly simple” in its theoretical claims, one might ask whether the same can be said of its implied computational claims, namely, that one can implement an algorithmic model of language by eschewing long-standing linguistic orthodoxies like distinct lexicons and rules, or grammar-wide formal categories. The issue then is not so much whether RCG can be efficiently realized but whether it can be computationally realized at all. It may be that RCG is too disarming simple to provide the computational sufficiency required for a practical implementation. While there exist some computational proofs-of-concept for other varieties of construction grammar, such as embodied construction grammar (see Bryant, 2003; Bergen and Chang, 2005), there are none, to our knowledge, that demonstrate the large-scale computational feasibility of RCG’s most radical claims.

Nonetheless, large scale computational systems have been successfully created and deployed for a linguistic model that is, from the perspective of computational sufficiency, remarkably similar to RCG. Within the paradigm of *example-based machine translation* or EBMT (e.g., see Carl and Way, 2003), researchers also dispense with traditional linguistic assumptions about the absoluteness of formal categories and the hard distinction between lexicon and grammar. Instead, EBMT exploits a database of construction-like exemplars which map fragments of a source language like English onto corresponding fragments of a target languages like Japanese and German. Translations for a given text are then produced, quilt-like, by combining the appropriate translation fragments as provided by these exemplars. EBMT exemplars are not hand-coded, but are typically derived from existing parallel corpora such as the multilingual product manuals that international corporations produce in abundance. Like RCG, EBMT strives toward a radical view of language processing that jettisons as much linguistic convention as possible, not because of a fundamental distaste for orthodoxy, but because of the onerous knowledge-engineering demands and resulting brittleness that comes from the need to develop and maintain hand-crafted grammars and lexicons.

The exemplars of EBMT exhibit many of the key properties of constructions in RCG. Since exemplars range in size from single lexemes to entire sentences, they collectively constitute both the lexicon of the system and its implicit grammar. Likewise, EBMT systems like Veale and Way's *Gaijin* (see Carl and Way, 2003) support schematic exemplars that serve the role of abstracted grammar rules. And like RCG, EBMT does not attempt to cluster exemplars into families of mappings that exhibit the same deep-structural form. Rather, each exemplar is treated as an island, and no generalization is formed, for example, from active and passive variations of the same textual proposition. Furthermore, because exemplars are typically derived via automated statistical alignment techniques from bilingual corpora, the same exemplars can be extracted many times over, thus providing a basis for considering some exemplars as providing more prototypical translations than others.

While EBMT exemplars are mappings from form to form, rather than form to meaning, their structure (as again judged from the viewpoint of computational sufficiency) is comparable. Indeed, in no sense do form-to-form mappings underestimate the difficulties of dealing with form-to-meaning mappings. Exemplars that are semantically compatible at a propositional level are frequently incompatible at a form level, due to disagreements in number, case, grammatical role or register. This problem, denoted *boundary friction*, means that EBMT exemplars can be extremely sensitive to the ways they are combined. Since EBMT strives toward a linguistics-lite theory of translation, most systems tackle boundary friction with statistical or corpus methods, such as establishing the validity or otherwise of possible translation candidates by determining which candidate occurs most often on the world-wide-web.

To the extent that RCG can also be seen as a rules-lite approach to language, RCG and EMBT are sufficiently similar to ultimately imagine a role for the semantic maps of RCG in the detection and elimination of boundary friction, for EBMT and RCG do attempt to model comparable issues of language use in comparable ways. As such, the engineering practicality of EBMT as a model of machine translation strongly supports the computational feasibility of RCG as a linguistic theory.

## 5.2. Complexity Theory in Action

As we have seen, both the GTVH and blending theory rely crucially on a cognitive mechanism capable of mapping two arbitrarily complex mental structures. In the case of the GTVH, this mapping determines the extent of script overlap, where scripts are now mathematically conceived in graph-theoretic terms (see Attardo *et al.*, 2002). In the case of blending theory, such mappings are central to the identification of correspondences between input spaces (see Veale and O'Donoghue, 2000). Several analyses have been made of the complexity of the structure mapping process required for analogical reasoning, and so by extension, conceptual blending and script-based joke analysis (see Winston, 1982; Falkenhainer *et al.,* 1989; Forbus and Oblinger, 1990). Nonetheless, though structure-mapping is intuitively NP-Hard, none of these analyses have the status of a proof. In this section, we present our own proof and a subsequent analysis to highlight some of the important properties of structure-mapping.

Structure-mapping between two structured representations (essentially graphs) proceeds by first identifying obvious partial mappings between sub-structures of each representation. These partial mappings are then combined to create successively larger mappings until a maximal partial mapping is generated; such mappings are maximal in the sense that no other elements can be added without violating the 1-to-1 isomorphism of the mapping. Though some metaphors and blends may involve many-to-one correspondences, as in the blend "one-man-band" (which maps every position in a band onto the same musician), most analogical mappings are intelligible by virtue of being isomorphic (e.g., see Falkenhainer *et al.*, 1989; Veale and Keane, 1997). The goal of structure-mapping is to find the largest maximal partial mapping that is possible between both conceptual representations. In an ideal situation, every element of each representation is mapped in such a mapping; in other situations, the largest partial mapping represents a "best match" of the available knowledge structures.

Defining structure-mapping in graph-theoretic terms allows us to identify the problem as a rewording of the known NP-Complete problem LCS, or Largest Common Sub-Graph (see Garey and Johnson 1979):

> **Analogical Mapping (AM):** Given the directed and arc-labelled graphs S = (SV, SA) and T = (TV, TA) which represent, respectively, the source and target domains of the analogy, we ask, do there exist subsets SA' SA, TA' TA, SV' SV, TV' TV, with |SV'| = |TV'| and |SA'| = |TA'| K such that the sub-graphs S' = (SV', SA') and T' = (TV', TA') are isomorphic? Two graphs S' and T' are isomorphic if there exists a function f: S' T' such that $<v_i, v_k>$ S' iff $<f(v_i), f(v_k)>$ T'.

However, since LCS is a decision problem (returning true or false, rather than actual structure), it does not provide us with a particularly useful low-level picture of the computation performed during structure-mapping, which is centred around the aggregation and combination of partial mappings. The following NP-Hard problem 3DM (see again Garey and Johnson 1979) yields a clearer picture:

**Unique 3-Dimensional Matching (3DM):** Given a set M of points in 3-D space, i.e., $M \subseteq X \times Y \times Z$, where X, Y and Z are disjoint sets of integers and $|X|=|Y|=|Z| = q$, find the largest set $M' \subseteq M$ such that no two elements of $M'$ agree in any co-ordinate.

We can recast structure-mapping in terms of 3DM quite easily, if the set of points in three-dimensional space is seen as the set of possible cross-domain correspondences between source and target structures. Partial mappings will thus be aggregates of these points, such that the largest maximal partial mapping will correspond to the largest set $M'$ of non-overlapping points. A detailed reduction of 3DM to structure-mapping, and thus, a complete proof of the NP-hardness of the latter, can be found in Veale and Keane (1997).

### 5.3. Ramifications

Such a proof of NP-hardness imposes a number of very real constraints on cognitive theories that are predicated on an ability to generate systematic mappings between arbitrarily complex structures. First, these theories cannot assume an optimal mapping, unless they can demonstrate that the structures involved will always possess a simplicity bordering on the trivial (as is not the case for either the GTVH or blending theory). Second, computational specificity demands that the nature of any sub-optimal mapping scheme be reflected at the theory level, for Veale and Keane (1997) demonstrate that the effectiveness of any sub-optimal mapping scheme is intimately bound up in the nature of the representations on which it is used. In the case of the GTVH, for instance, this suggests that scripts not be viewed as arbitrary graph structures after all; recall that this arbitrariness was introduced in Attardo *et al*. (2002) to allow script-overlap to encompass both narrative similarity for jokes and phonological similarity for puns. Generally speaking, the more highly constrained a representation, the more highly constrained and delimited will be the search space derived from this representation. Thus, a more constrained script representation (like that originally assumed by Raskin, 1985) may lead to a polynomial-time mapping algorithm more akin to the efficient, top-down unification of feature structures. We see efficient unification of this kind in computationally realized models of construction grammar, such as the implementation of Embodied Construction Grammar (or ECG) described in Bryant (2003) and Bergen and Chang (2005).

## 6. Concluding Remarks

Language can be a bewitching medium in which to theorize, for one can easily be seduced into assuming a clear understanding of a given concept simply because one knows how to use the corresponding words. Much of the later philosophy of Wittgenstein (e.g., see Wittgenstein, 1958; 1979) is characterized by an attempt to reveal this seductive power for what it is, as when he argues – counter to our linguistic intuitions on the matter – that there is no coherent concept of Knowledge behind a word like "know"[4] despite the ease with which this word is used. Many of the problems that hinder the true understanding of a scientific phenomenon are, in essence, problems of language, for theories are linguistic artefacts in their own right. One must separate out the rhetorical devices used by a theory to gain widespread acceptance from the ontological commitments that the theory makes to specific entities and processes. For instance, a theory that makes reference to the notion of a "domain" might make an ontological commitment to a particular cognitive structure, or might simply use the term as a rhetorical shorthand. Since rhetorical and ontological commitments are often tightly entwined in the persuasive exposition of a theory, these are best separated by a computationalist perspective that appeals to Jackendoff's notion of computational sufficiency: only those claims of a cognitive theory that are mirrored by a corresponding computational distinction should be considered as having a potentially explanatory value.

This separation is easier if a theory makes few rhetorical commitments, or in the case of Radical Construction Grammar, the theory makes what amount to *negative* ontological commitments. For instance, RCG explicitly denies the existence of universal syntactic categories, and even denies that such categories have language-wide scope. But to make this minimalism possible, RCG makes a *positive* ontological commitment to the existence of semantic maps in a shared conceptual space.

The rhetorical qualities of a term often colour its theoretical function. Consider again the term "script", which denotes the ontological core of Attardo and Raskin's general theory of humour. Raskin (1985) originally employs the term in the sense popularized by the work of Schank and Abelson (1977), that is, to denote a schematic structure that imposes a sequential, causal ordering on a narrative and which reflects a single top-down interpretation of events based on an abstracted distillation of relevant episodic memories. because of the cinematic connotations of the word, one is intuitively directed to view a cognitive script as a film script, one that provides a narrative sequence of actions for a set of actors to perform, albeit for mundane actions like visiting the doctor, eating at a restaurant or being the victim of a mugger. However, this is not the interpretation of the term favoured by a computational reading of the SSTH and GTVH, as evident from attempts to realize these theories in software. In Raskin (1996), a script is taken to mean a conventional frame structure, much like the frames of conceptual blending theory, one supposes, or the unification structures used to represent constructions in Bryant's (2003) implementation of construction grammar. Raskin's frames thus comprise a collection of labelled slots, denoting event roles, into which semantic fillers may be placed. But in Attardo *et al.* (2002), a script is revised to mean little more than a generalized graph structure (that now presupposes an NP-hard mapping problem rather than a polynomial time unification process). With this generalization, the GTVH moves further from the conventional cinematic interpretation of a script toward a completely neutral symbolic representation. The term "script", however, remains, to suggest that the GTVH still views humour as a by-product of our cognitive faculty for reasoning about events and their social consequences. How else can a social phenomenon like humour arise out of an abstract operation over graphs? But however suggestive the term "script", the underlying computational distinction needed to support this suggestion is now absent, and we conclude that the explanatory power of the GTVH is diminished as a result.

The GTVH and blending theory serve as interesting case studies of cognitive theorizing for yet another reason: each represents a separate evolution of Koestler's (1964) influential theory of Bisociation, in which creativity is said to arise at the juncture of orthogonal mental "matrices". Both theories embody different solutions to the issues of specificity that vex Koestler's original theory, which arise for the most part from Koestler's lack of a suitably concrete vocabulary with which he might express his claims. The computationalist influence on the cognitive sciences has since – through its emphasis on specific well-formed representations and processes – allowed theories like blending and the GTVH to surmount these descriptive difficulties. We now possess a rich vocabulary of frames, scripts, schemata and mappings that each suggest largely the same notion, more or less, as we move from Cognitive Linguistics to Cognitive Science to Artificial Intelligence and Computer Science. As we have seen, however, there is still a substantial difference between, on one hand, the rhetorical suggestiveness of such terms, and on the other, their precise theoretical meaning as seen from the perspective of computational sufficiency. The ongoing importance of the computationalist perspective is to ensure that these terms are always understood from the latter, formally-precise, perspective. Only then can cognitive theories be judged and evaluated on their fundamentals, that is, on those aspects that arise directly from computational distinctions, rather than on the suggestive linguistic penumbras that surround them.

## Notes

1. Such a subversion is evident in the following joke: “A worker at the Guinness beer factory drowns tragically one day after falling into a giant vat of Guinness. A manager is dispatched to inform the unfortunate man’s wife. ‘Did he die quickly?’, she asks the manager. ‘No’, he replies, ‘in fact, he got out twice to use the toilet’”.
2. For “decision problems”, which are those that diagnose a given set of inputs to return a simple true or false, the corresponding class of hardest problems is called NP-Complete.
3. There is a third option, which is to assume that the brain is, in fact, a non-deterministic machine (exploiting, say, some oddity of quantum mechanical physics) or that it does not operate on Turing-compatible computational principles. Though possible in principle, this option is still largely the stuff of science fiction.
4. “One is often bewitched by a word. For example, by the word ‘know’” (Wittgenstein, 1979).

## 7. References


Attardo, Salvatore and Victor Raskin
1991 Script theory revis(it)ed: joke similarity and joke representational model. *Humor: International Journal of Humor Research* 4-3: 293-347.

Attardo, Salvatore and Christian F. Hempelmann and Sara Di Maio
2002 Script oppositions and logical mechanisms: Modeling incongruities and their resolutions. *Humor: International Journal of Humor Research* 15-1: 3-46.

Bergen, Benjamin and Nancy Chang
2005 Embodied construction grammar in simulation-based language understanding. In: Jan-Ola Östman and Miriam Fried (eds.), *Construction Grammars: Cognitive Grounding and Theoretical Extensions*: 147-190. Amsterdam: John Benjamins.

Bryant, John
2003 Constructional analysis. M.A. thesis, University of California, Berkeley.

Bell, Philip
2002 Blending linguistics and cognition: a methodological critique. In: Anders Hougaard and Stefan Nordahl (eds.), *Odense Working Papers in Language and Communication* 23: 163-180. Syddansk University, Denmark.

Carl, Michael and Andy Way (eds.)
2003 *Recent Advances in Example-based Machine Translation.* (Text, Speech and Language Technology 21). Berlin: Springer.

Chomsky, Noam
1957 *Syntactic Structures*. The Hague: Mouton.

Coulson, Seana
2000 *Semantic Leaps: Frame-shifting and Conceptual Blending in Meaning Construction.* New York and Cambridge: Cambridge University Press.

Croft, William A
2001 *Radical Construction Grammar; Syntactic Theory in Typological Perspective*. Oxford: Oxford University Press.

Falkenhainer, Brian and Kenneth D. Forbus and Dedre Gentner
1989 Structure-mapping engine: algorithm and examples. *Artificial Intelligence*, 41: 1-63.

Fass, Dan
1988 An account of coherence, semantic relations, metonymy, and lexical ambiguity resolution. In: Steven I. Small, Garrison W. Cottrell and Mark K. Tanenhaus (eds.), *Lexical Ambiguity Resolution: Perspectives from Psycholinguistics, Neuropsychology and Artificial Intelligence*:151 – 178. San Mateo California: Morgan Kaufman.

Fauconnier, Gilles and Mark Turner
1998 Conceptual integration networks. *Cognitive Science*, 22(2):133–187.

Fauconnier, Gilles and Mark Turner
2002 *The Way We Think.* New York: Basic Books.

Forbus, Kenneth D. and Daniel Oblinger
1990 Making SME pragmatic and greedy. In: *Proceedings of the Twelfth Annual Meeting of the Cognitive Science Society:61-68.* Hillsdale, New Jersey: Lawrence Erlbaum.

Gardner, Howard
1985 *The Mind's New Science*. New York: Basic Books.

Garey, Michael R. and David S. Johnson
1979 *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: W.H. Freeman.

Gibbs, Ray
2000 Making good psychology out of blending theory. *Cognitive Linguistics* 11(3/4): 347-358.

Goldberg, Adele
2003 Constructions: a new theoretical approach to language. *TRENDS in Cognitive Sciences*, 7(5): 219-224.

Jackendoff, Ray
1987 *Consciousness and the Computational Mind*. Cambridge, Massachusetts: MIT Press.

Koestler, Arthur
1964 *The Act of Creation.* New York: Macmillan.

Miller, George
1995 WordNet: A lexical database for English. *Communications of the ACM*, 38(11): 39-41.

Plath, William J.
1973 Transformational grammar and transformational parsing in the REQUEST system. In: Antonio Zampolli (ed.), *The Proceedings of the fifth International Conference on Computational Linguistics*, Pisa, Italy: 585-610.

Popper, Karl
1959 *The Logic of Scientific Discovery*. New York: Basic Books.

Raskin, Victor
1985 *Semantic Mechanisms of Humor*. Dordrecht: D. Reidel.

Raskin, Victor
1996 Computer Implementation of the General Theory of Verbal Humor. In: Jan Hulstijn and Anton Nijholt (eds.), *The International Workshop on Computational Humou*r: Proceedings of the Twelfth Twente Workshop on Language Technology (Series TWTL 12):9-20. Enschede: UT Service Centrum.

Ritchie, Graeme
2003 *The Linguistic Analysis of Jokes*. London: Routledge.

Schank, Roger C. and Richard P. Abelson
1977 *Scripts, Plans, Goals and Understanding*. New York: Wiley.

Turing, M.
1936 On computable numbers, with an application to the Entscheidungsproblem. In: Robin O. Gandy and C. E. Michael Yates (eds.), The *Collected Works of Alan Turing (volume 4*) Amsterdam: North Holland.

Veale, Tony and Diarmuid O'Donoghue and Mark T. Keane
1996 Computability as a limiting cognitive constraint: complexity concerns in metaphor comprehension. In: Masako Hiraga, Chris Sinha and Sherman Wilcox (eds.), *Cognitive Linguistics: Cultural, Psychological and Typological Issues:* 129-155. Amsterdam and Philadelphia: John Benjamins

Veale, Tony and Mark T. Keane
1997 The competence of sub-optimal structure mapping on 'hard' analogies. In: Martha E. Pollack (ed.), *Proceedings of the fifteenth International Conference on Artificial Intelligence,* Nagoya, Japan: 232-327. San Mateo, California: Morgan Kaufmann.

Veale, Tony and Diarmuid O'Donoghue
2000 Computation and blending. *Cognitive Linguistics*, 11(3-4): 252-281.

Veale, T.
2004 Incongruity in humor: root-cause or epiphenomenon? *Humor: International Journal of Humor Research* (Festschrift for Victor Raskin) 17-4: 410-428.

Winston, Patrick H.
1982 Learning new principles from precedents and exercises. *Artificial Intelligence* 19: 321-350.

Wittgenstein, Ludwig
1958 *Philosophical Investigations*. (Translated by G.E.M. Anscombe). London: Blackwell.

Wittgenstein, Ludwig
1979 *On Certainty* (Translated by D. Paul and G.E.M. Anscombe). London: Blackwell.