ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Procedia - Social and Behavioral Sciences 97 (2013) 464 – 473

Procedia
Social and Behavioral Sciences

The 9th International Conference on Cognitive Science

Does (linguistic) computation need culture to make cognition viable?

Prakash Mondal*, Ravi Shankar Mishra

Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India

Abstract

This paper will deal with how and in what ways (linguistic) computation as part of linguistic competence may relate to aspects of culture in the context of the cognition which becomes viable by being grounded in the possible conjunction of mental computations and cultural praxis. The possibilities of cultural capacities are enormous across societies and/or cultures, but linguistic computations as have been postulated are restricted by the nature of constraints specific to natural language. The purpose of this paper is to see the consequences of how these two can make cognition viable.

Selection and/or peer-review under responsibility of the Universiti Malaysia Sarawak.

Keywords: Linguistic computation; culture; cognition

1. Introduction

Culture and natural language are not simply a part of nature; rather they configure and in many ways, constrain the very nature of human cognition. One often encounters debates on whether natural language is itself a part of culture or whether culture is a part of nature or not. These debates may be tangential to the issue of how language within cognition relates to culture. Now what does it mean for language (within cognition) to relate to culture if this is what characterizes the fulcrum of issues revolving around mind, language and culture? Computation, on one hand, underlies the core operations buttressing linguistic (especially syntactic) constraints, principles/rules as crystallized in Generative linguistics [1], [2] and elsewhere in theoretical linguistics. On the other hand, socialization processes as part of the cultural praxis carry over to a notion of culture and cultural capacities though it is palpably visible that there is no way of having a transparently demarcated notion of culture. Placed in the context of our cognitive substrate, (linguistic) computation and culture appear to be at odds with one another it is as if cognition is sandwiched between (linguistic) computation and culture. However, there is no way one can sidestep these issues if the purported goal is to probe the intricately knotted relationship between (linguistic) computation, mind and culture even if the relationship between (linguistic) computation, mind and culture put together has a bewildering amount of impermeability. This is more so because their ontological domains do not properly overlap although they interpenetrate one another in other ways too. In such a situation, to make a theoretical/conceptual move is like

* Corresponding author. Tel.: +91 9811881538.
E-mail address: mndlprksh@yahoo.co.in

Selection and/or peer-review under responsibility of the Universiti Malaysia Sarawak.
doi:10.1016/j.sbspro.2013.10.260

ploughing on through the maze of abstractions, emergent layers of phenomena and a confounded plexus of relations. It is unclear how such a complicated network of relationships can be dealt with in a swoop with the relevant issues thoroughly addressed.

Given this scenario, the current paper will make no attempt to build a theory of how the relationship between (linguistic) computation, mind and culture can be satisfactorily constituted and thereby defined. Nor will it pretend to resolve the complications each one presents when taken alone. Rather, it will try to figure out the points of contact at which (linguistic) computation, mind and culture come together but simultaneously do not even chime with each other in ways which sort of destabilize the very relationship, however postulated, between (linguistic) computation, mind and culture. It is believed that this exercise will be fruitful in the sense that this can help unravel some of the deeper but persistent puzzles and problems encompassing (linguistic) computation, cognition and culture in broader terms.

The paper is structured as follows. The first section will discuss the relation between (linguistic) computation and cognition. The second section will shed some light on the construal of a relationship that may obtain between cognition and culture. The third section will focus on linguistic phenomena like long-distance dependency, gapping, presupposition etc. from a swathe of the territory of syntax and semantics to uncover the kind of recalcitrant messiness that emerges at the cross-section of (linguistic) computation, culture and cognition. And finally, the fourth section will briefly tease apart what the emerging puzzles and problems mean for any understanding of cognition, culture and language.

2. (Linguistic) Computation and Cognition

The relation between (linguistic) computation and cognition is not as straightforward as it may sound though the way the relation between (linguistic) computation and cognition in theoretical linguistics has been projected makes it appear so. But before we plunge into the details of how that relation may be conceived of, the notion of (linguistic) computation needs to be clarified. So when a question on whether something is computational or not is asked, much hinges on the fact that the right concept of computation is applied to the phenomenon that is to be scrutinized to see whether it falls under computation. Similar considerations apply to the case here as we focus on language and wonder what can be linguistic computation. For all the vagueness surrounding the notion of (linguistic) computation, it appears that (linguistic) computation fits well with the classical sense of computation where inputs are mapped to outputs according to some well-defined rules by means of symbolic manipulation of digital vehicles in the form of linguistic strings. Much of formal linguistics has employed this notion of linguistic computation implicitly or explicitly mainly because the representational vehicles of language are discrete in form. Still a question remains. Can we take linguistic computation as a *generic* computation that encompasses both digital and analog computation [3]? Even if this is a bit difficult to answer, the answer is more likely to be no. It is somewhat clearer that the digital notion of computation has been predominant all throughout the field of cognitive science, in general and theoretical linguistics, in particular; hence the analog sense of computation does not apply to linguistic computation since in analog computation computational processes are driven and determined by the *intrinsic* representational content of the vehicles which are analog in nature [4], whereas digital computation involves mappings of inputs onto outputs that are executed without any regard to the content-determining properties of the representational digital vehicles.

In the Generative linguistics framework linguistic representations are construed to be (internalized) mental representations, and operations on such representations by means of rule systems are computations. And these operations have to have an algorithmic character to be called computations; otherwise it does not make any sense to have operations which are posited to be computational in form and character. The core computational operation in the current version of Generative Grammar is Merge [1]. Merge is a binary operation involving two syntactic objects- say, α and β. Now the operation Merge will merge α and β to create γ $\{\alpha, \beta\}$ when γ is the label of the new object. An example can be given to illustrate this. Let's take the sentence below:

John believes Mary to be beautiful. (1)

The way the words of the sentence in (1) have been combined to make the sentence (1) can be shown below with the help of a tree as standard in theoretical linguistics.

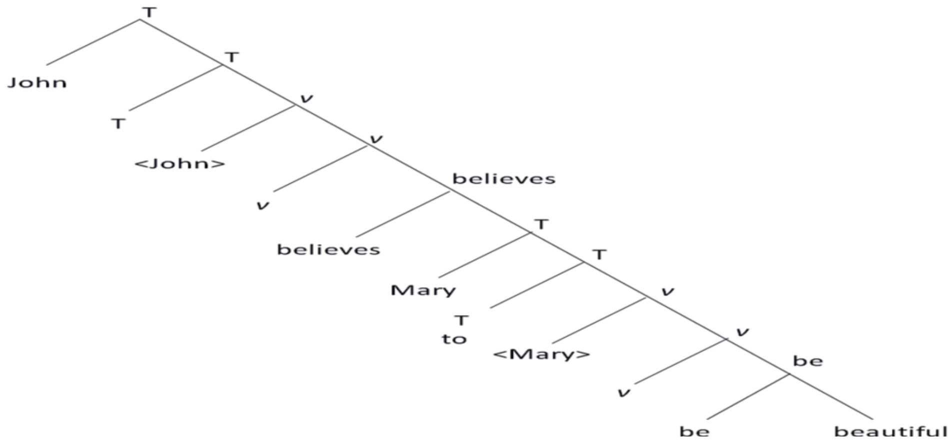

Fig. 1. Derivation of (1) through the operation Merge

Here the operation Merge builds the tree through an iteration of the application of the operation (with *v* denoting the verb phrase and T the tense phrase). 'Mary' and 'John' are placed in brackets to show that they have been Internally Merged higher. 'Mary' and 'John' have been Internally Merged in that they are Merged higher to the structure Merge has so far built, of which 'Mary' and 'John' (independently) are already a part. The rest has come about through what is called External Merge. Now it needs to be seen whether this operation is a computational operation or not. One may note in this connection that a function does not become a computable function just by virtue of being a function as there are a lot of functions which are not computable functions. And a computable function is generally the one which can be stated as an algorithm. If so, not all functions can be algorithmically defined. However, it is easy to see that the operation Merge can be stated algorithmically. Here is how. First, Merge can be stated in the form of a binary function that operates on S, the set of syntactic objects. So it will look like: Merge (S × S) = S. Second, Merge in this form can be stated, by following Foster [5], as a sequence of transitions between states of a machine-say, a Turing machine- each of which consists of a value and a label. This can be shown below.

$$[SO_1\text{: } LI_1\ SO_2\text{: } LI_2\ \ L\text{:}\mathcal{L}] \rightarrow [SO\text{: } LI_1, LI_2\ \ L\text{: } \mathcal{L}] \rightarrow [SO\text{: } \{LI_1, LI_2\}\ L\text{: } \mathcal{L}] \rightarrow [SO\text{: } \mathcal{L}\ \{ LI_1, LI_2\}] \qquad (2)$$

(2) represents a sequence of states. Here SO_i is a syntactic object, LI is a lexical item (including functional categories like tense T or verb *v*) and L is a label (of an SO). The left hand side of the colon represents a label and the right hand side of the colon denotes the value of that label. Each structure enclosed within braces [ ] with such label-value pairs constitutes a state followed by the next state- the arrows represent the relevant transitions between such states. One can see that the values of SO_1 and SO_2 have been grouped in the second state. This gives us a handle to the representation of objects created by Merge as a pair, if needed. However, the next state gives us the set of values of SO_1 and SO_2 thus having the flexibility that one can choose either of the states (the second or the third) as and when needed. This also gets us out of the necessity to impose any (in)determinacy on the syntactic objects generated by Merge.

Now that a characterization of how linguistic operations can be computational in nature has been arrived at. Moreover, this is not just a way of building structures-syntactic structures. Building semantic structures can also be characterized this way. Semantic structures are built in a compositional manner in that the semantic structure of an expression is a function of the semantic structures of the expressions that are parts of the whole semantic structure. This insight has been a central theme of Montague Grammar and has been incorporated in formal semantics too. So the semantic structure of the sentence (1) can be compositionally derived in the following fashion.

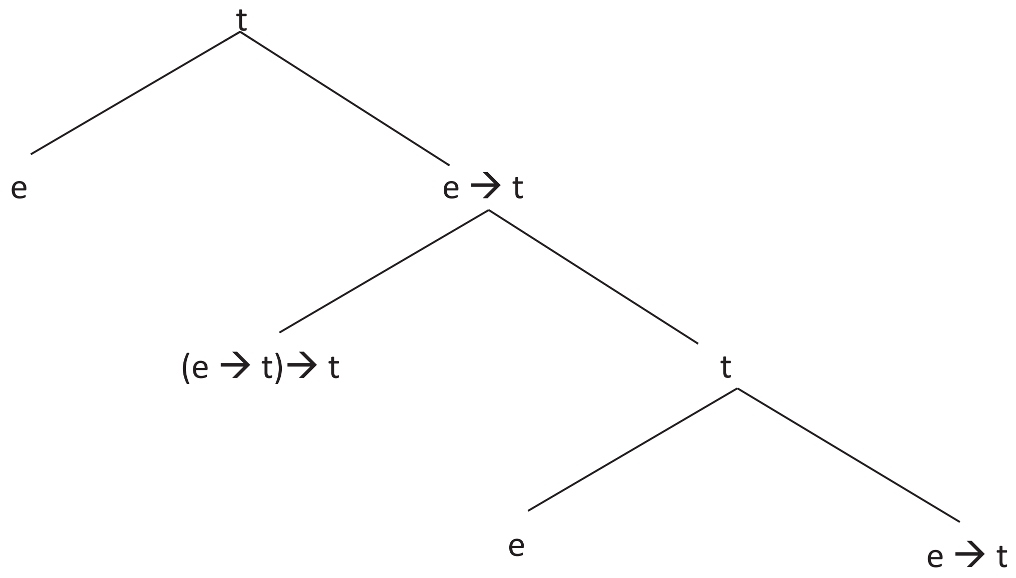

Fig. 2. Compositional semantic derivation of the sentence in (1)

Here *e* represents an entity denoting a noun/noun phrase and *t* is truth value. The arrow represents the mapping/function from one semantic value to another. One can check that both *e*'s at the bottom of the tree are cancelled out to give rise to *t* at the mother node and then *t*'s are cancelled out at the next level and so on, finally leading to a *t* at the top of the tree which is the semantic value of a proposition. This is what a compositional semantic derivation looks like. Additionally, the mapping/function from one semantic value to another is also a computable function- that is, this function can be algorithmically described. This can be simply shown, as above, in the form: $[S^V_1: x \;\; S^V_2: y] \rightarrow [S^V: x,y] \rightarrow [S^V: \langle x, y\rangle]$ when S^V is a semantic value. Thus one can have a function like f (x) =y when x and y denote semantic values, and complex functions can be constructed through function composition. So far, linguistic operations have been shown to be computational separately for syntactic structures and semantic structures thereby leading someone to feel that operations on syntactic structures and semantic structures are independently computational. But this is not quite true because semantic structures are always deductively defined on the set of strings syntax generates. In fact, this can be clearly seen in Categorial Grammar [6] where syntactic structure is just a medium for the derivation of semantic structures. Let's look at the derivation of (1) in Categorial Grammar below. There are a set of primitive categories like N (noun), NP (noun phrase), S (sentence). Any other complex category is derived through a combination of these primitive categories by means of backward and/or forward slashes. The compositional rules via cancellation of categories are similar to the ones applied in Fig. 2. Since the slashes are mapping functions, they are sensitive to the order of application of functions and can be algorithmically described in a way similar to what has been shown above. So one can see how semantic composition piggybacks on syntactic structure building. This is more strongly visible in Dynamical Semantics [7].

So far so good. But how does this all relate to cognition? The link provided in Generative Grammar is via a kind of psychologism grounded in the conception of a mental grammar constituted by a complex relation of rules, representations and constraints. What this means is that when a language is acquired, what is acquired is not exactly a language but a grammar mentally instantiated and this allows the person to have a competence in the language by virtue of having internalized the rules and constraints the grammar consists of. Though this link has been challenged by Langendoen and Postal [8] on the ground that mathematical properties of natural language are such that they militate against any psychologically grounded ontology of natural language, we will not pursue this line of thought. Suffice it to say that a connection between computational operations underlying linguistic rules, constraints and our cognitive substrate has also been motivated on the basis of a conception of language as a module/a faculty of the human mind.

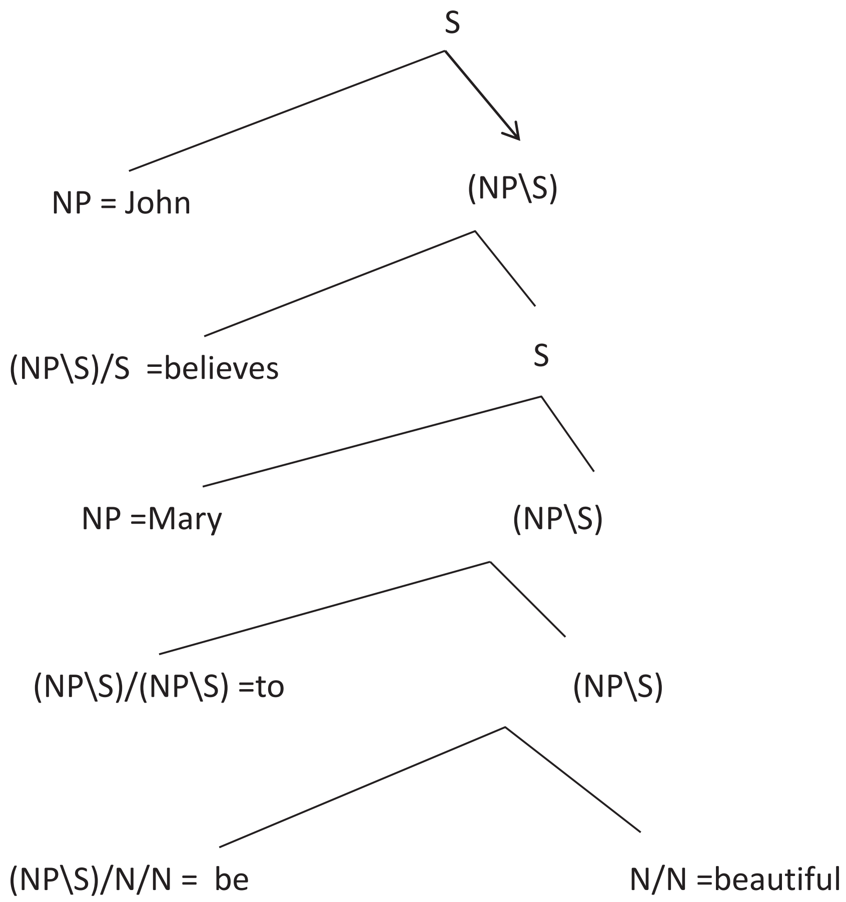

Fig. 3. Compositional semantic derivation of the sentence in (1) in Categorial Grammar

Other linguistic frameworks like Parallel Architecture [9] have also established a connection between language and cognition by having distributed the interaction among the components of language-syntax, semantics and phonology- in a way that each component as part of the faculty of language in mind has its own primitives, combinatorial rules/principles, and constraints on the mutual correspondence among syntax, semantics and phonology. Since structures in this framework are not intentionally defined and thus not about anything, this raises the question of whether the combinatorial rules/principles in each subsystem of language and the constraints can be computational or not. It may be, in principle, possible given the characterization of combinatorial rules/principles and constraints, though the issue is best left open. Even if Montague Grammar or Categorial Grammar does not clearly make any commitment to any brand of psychologism, structure building process in Dynamical Semantics is modelled on the interpretation process in mind and a similar tenet is also to be found in Discourse Representation Theory [10]. Important to note in this connection is that fact that the link between operations governing linguistic structures and cognition as postulated by these frameworks is extrinsic in the sense that structural operations in Dynamical Semantics or Discourse Representation Theory are posited to parallel/match any computations that the mind does in language processing but are not themselves identified with mental computations, however characterized. Overall, we get both logically intrinsic and extrinsic connections between linguistic computation and cognitive computation.

3. Culture and Cognition

The term 'culture' evokes such a myriad mesh of senses and connotations that it is perhaps easier to understand than definitionally demarcate it. The notion of culture carries over not just to a notion of environmental interaction of the individual, though that is definitely a part of it. It broadly encompasses a whole gamut of social processes, capacities, activities, humanly defined relations, conventions, norms and values and so on. One cannot certainly deny the role of symbolic forms mediating socio-cultural interactions and relations. Perhaps that is the reason why structuralist movement had such an impact in both sociology and anthropology. But the notion of symbolic forms does not merely incorporate linguality; rather it embraces the process of semiosis itself since at the heart of semiosis is interpretation, as propounded by Charles Sanders Peirce and Thomas Sebeok. One may see that this is what provides the (missing) link between culture and cognition. Cultural patterns are not just fluctuations in the social space, they have their unstably regular form-semiotic or otherwise. It is not *just* that culture changes along with or in reaction to the undulating conditions of discourse any more than the physical environment changes with the changing conditions of theories of ecology or of the biosphere.

Perhaps there is more to it than meets the eye. There may well be a lot of resistance from or constraints on the semiotic processes linking culture with cognition. One example might make this clearer. Marriage and kinship relations are socio-culturally constructed, but this does not license the implication that humans can/do conceptualize any vertical/horizontal depth in the form of a kinship mediated by marriage, birth, race or whatever. Thus even though a kinship that relates an individual A to B and B to C and hence A to C (a case of transitivity) or a kinship that relates an individual A to B and A to C and hence B to C (a case of euclideanity) is easier to conceptualize; but this does not mean that one can conceptualize, in daily interactions, any arbitrarily long chain of length n connecting any X to Y horizontally (transitively or symmetrically and so on) or vertically (any n-th order relation among relations).

All the same, it is too simplistic to just show that all socio-cultural interactions are cognitively grounded. This is as trivial as saying that all cultural capacities are biologically mediated/grounded. It is quite the case that one's construal or conceptualization mandates a kind of construction-discursive and/or social- exercising a causal efficacy through acceptance and efficiency [11]. A discursive construction is constituted by the way one's construal of something becomes relevant for action on the part of the recipient (for example, abusing somebody). Social construction is more persistent and stable, and forms the macrostructure of understanding of a socially available target prevailing over and then transcending the boundaries of the discourse of its origin (learning how to make a pot, for example). But these are all ways of constraining the form socio-cultural interactions assume. Culturalization and socialization processes format the nature of cognition when humans learn and engage in language(s), mathematics, games, artistic activities etc. In fact, these are logically extrinsic orders of connection between culture and the cognitive substrate. It would be equally or perhaps more interesting to see whether cultural interaction/practices as part of the cultural praxis is/are not merely instantiated in but *identified* with cognitive processes or vice versa. This will give us the logical intrinsicality of cultural-cognitive processes/interactions. As we move on to the later sections, we will see that this question becomes sharper than can be appreciated at this juncture.

4. At the Cross-section of (Linguistic) Computation, Culture and Cognition

Now it is necessary to articulate what really matters to the relationship between (linguistic) computation, culture and cognition. To show this, we will consider a range of linguistic phenomena from areas of syntax and semantics. No attempt will be made to bring out a comprehensive treatment of the phenomena to be discussed in this connection. Rather, the relevant issues will be spelled out by having them positioned at the intersection of (linguistic) computation, culture and cognition.

Let's first look at some examples from long-distance dependency. The examples below show that any dependency should be local but long dependencies are possible as well in some circumstances.

What does John think (that) he needed_? (3)
*Who/what did John ask why Mary looked for_? (4)
What did she move out to buy some stuff from_? (5)
*Which place did John meet a lot of girls without going to_? (6)

The sentences in (3-4) show that *Wh*-movement or the dependency between the gap and the occurrence of the *Wh*-phrase should be local. In (3) the apparently long dependency between the gap and the occurrence of the *Wh*-phrase is actually summed over smaller dependencies/movements (through the top layer of the embedded clause-the CP (complementizer phrase), while it is not so in (4) where the CP of the embedded clause already contains a *Wh*-phrase. The sentence (6) is ungrammatical because of a long-distance dependency between the gap in an adjunct and the *Wh*-phrase; but interestingly (5) is fine even though there exists a long-distance dependency between the gap in an adjunct and the *Wh*-phrase. Cases like these have puzzled linguists and have led to the refinement of much of the technology in the Generative linguistics paradigm. It is because if a dependency/movement is posited to be local, it is mysterious why some dependencies/movements are sometimes possible (as in 5) and sometimes banned (as in 6) when the establishment of locality through successive cyclic movement/summing over smaller dependencies is controlled.

To solve this problem, one proposal made by Truswell [12] states Single Event Grouping Condition which requires, in the case of a long distance dependency of a *Wh*-phrase, two events described in the matrix verb phrase and the adjunct clause/phrase to form a macro-event with the events (and sub-events) overlapping spatiotemporally and at most one of the events from either the matrix verb phrase or the adjunct clause/phrase having an agentive character. Two events form a macro-event (joint/extended event) only if they have a causal (direct or enabled) relationship and the causing event precedes the one caused. This explains why (5) is fine but (6) is banned as the events in the matrix verb phrase and the adjunct clause/phrase in (5) form a macro-event while those in (6) do not (the event in the adjunct containing 'without' is not agentive in having not occurred at all). Since such a condition obtains at the interface between syntax and semantics, it is computed in cyclic manner at each step of the verb phrase building process in syntax. If so, it would be interesting to see how this works in a situation where the issues involving (linguistic) computation, culture and cognition come out clearly.

Let's suppose that we have a program P (which is run on a machine, say, a version of the Turing machine-plausibly in the human mind) that can determine which event causes which out of the two events in a sentence containing the matrix verb phrase and the adjunct clause/phrase. Additionally, it can also determine which event has an agentive character. Now two sentences 'This program P checks causal relations and the agentive character of events to determine which event causes which out of the two events in a sentence' and 'This program P determines which event causes which out of the two events in a sentence to check causal relations and the agentive character of events' are given simultaneously as input to P. In such a situation, the program P determines that the event, say e_1, in the matrix clause of the first sentence is the event that causes the one, say e_2, in the adjunct clause and hence the two events form a macro-event. But when the program moves over to the second sentence, the relationship is reversed and it is e_2 that causes e_1. The problem for the program is that now both e_1 and e_2 mutually cause each other and thus none precedes the other. If so, technically there does not seem to be a problem for the long-distance dependency formation of a *Wh*-phrase when moved to the front (as far as Single Event Grouping Condition is concerned) in that the events in each sentence form a macro-event. But then the paradox that arises is this: on one hand, the program P determines that e_1 causes e_2 and thus the first sentence can allow *Wh*-extraction from the adjunct satisfying Single Event Grouping Condition but the same program then finds that e_2 causes e_1 in the second sentence, and thus this violates Single Event Grouping Condition (the precedence relation among events must not be symmetric). So the program finds that long-distance dependency formation of a *Wh*-phrase is possible for each sentence taken alone and at the same time, not possible for any.

To solve the problem posed by the paradox above, one may argue that the program P will differentiate the events in the first sentence with some index *i* from those in the second sentence with some index *j*. One possibility is to index each event or pair of events in a sentence differently from any other events in other sentences. If so, this gives rise to another paradox. One can note that each of the sentences is about the program P itself. Now P determines that e_1 causes e_2 given that P itself checks causal relations and the agentive character of its behavior first and then this results in the determination of what causes what. But when P goes over to the second sentence, P finds that e_3, a new event causes e_4, another new event given that P's own determination of what causes what eventuates in the checking of the causal relations and the agentive character of its own behavior. Now in this situation, P determines that Single Event Grouping Condition is not violated in each sentence but still violated in each (again because of symmetric precedence of events)! One cannot appeal to truth in this case since the program P does not have access to any relation in which truth/non-truth may obtain. Even if P has access to the truth about itself, P determines, upon checking the first sentence, that the causal/contingent relation between the events e_1 and e_2 (when e_1 causes e_2) is true of its behavior. Next, it goes over to the second sentence and determines that the causal/contingent relation between the events e_3 and e_4 is false given its behavior observed during the reading of the first sentence but true on the basis of the structure of the second sentence (if the adjunct rationale clause is the resulting event). That is a contradiction! But have not we also stated that $e_1 \neq e_3$, $e_2 \neq e_3$, $e_1 \neq e_4$ and $e_2 \neq e_4$? If so, then this leads to another level of contradiction in that even though it is the case that $e_1 \neq e_3$, $e_2 \neq e_3$, $e_1 \neq e_4$ and $e_2 \neq e_4$, the events e_2 and e_3 refer to the same thing and so do the events e_1 and e_4 and on the basis of this fact P finds something to be true of its own behavior (upon reading the first sentence) and at the same time, false too (upon reading the second sentence). This is how all this begets one paradox after another when (linguistic) computation applies to cognition.

We shall now concentrate on another phenomenon-gapping. Let's look at the examples below. Here we have two sentences with the verb missing (when matched in interpretation with the one in the matrix clause) in the second conjunct of each of the sentences (7-8).

John wants to build a tower and Mary, _ a bridge. (7)
*John wants to erect a structure and Mary, _ a structure. (8)

The ungrammaticality of (8) shows that the object of the missing verbs needs to be the FOCUS (contrastive focus- by virtue of being contrasted with the object noun phrase of the matrix clause against the common ground constituted by the common verb 'wants to build/erect'). Overall, this is how gapping has been dealt with [13], [14]. Once again we imagine that there is some program, let's call it G, instantiated in the mind by some Turing machine. Let's also suppose that G can determine which phrase is the FOCUS since this is what turns out to be crucial in the examples of gapping above. Now a sentence 'Either John wants to erect a structure and Mary wants to erect a tower or Sony wants to destroy a structure and Amy wants to erect a structure' is input to G. Upon reading the first major conjunct of the sentence, G finds that 'wants to erect' is repeated twice across the first major conjunct in the two sub-conjuncts and thus determines that 'wants to erect' is the common ground (that can be deleted) and that 'a tower' is the FOCUS when contrasted with 'a structure'. But when G scans the second major conjunct, it determines that 'a structure' is the common ground and 'wants to erect' is the FOCUS contrasted with 'wants to destroy'. This gets G mired in a paradoxical situation. On one hand, 'wants to erect' is the FOCUS and 'a structure' is the common ground but on the other hand, 'wants to erect' is not the FOCUS and 'a structure' is not the common ground. Hence, this is a contradiction! To pull G out of the paradox, one may argue that G may scan each major conjunct separately and index the FOCUS and the common ground. Even if this is allowed, this will not save G from getting bogged down in another paradox. Here is how. Now suppose the sentence (7) in a form before the deletion of 'wants to build' is input to G. While scanning this sentence, G will determine that there is no common ground and there is no FOCUS in any of the conjuncts when independently checked. If so, there is no gapping in (7), but at the same time, there is a gapping in (7), as a matter of fact! Because of such contradictions, G will either halt or loop forever.

There is yet another phenomenon to be dealt with here. It is presupposition. Presupposition is like conventional implicature derived from the meanings implicated in a phrase/clause. This can be illustrated with the following example.

John_i is looking at the picture of his_i dead mother (9)
Philip_k regrets that he_k hates John. (10)

The sentence (9) has the presupposition that John's mother is dead/has died while the sentence (10) presupposes that Philip hates John. There is a well-known literature on how presupposition can be computed from sentences [15], [16], [17]. For example, the principle *Maximize Presupposition* [16] states that in a situation when *p* is the case, the strongest presupposition *q* out of all the choices {*q*, *r*, *s*, …} needs to be selected. Therefore, when one considers (10) if it is the case that Philip hates John, no such presupposition can be derived from a sentence like 'Philip opines that Philip hates John' since regretting holds true in more restricted conditions than mere opining. Be that as it may, what is relevant for our purpose is that presupposition can be derived from their sentences of origin by seeing how meanings of expressions are composed and what they (conventionally) implicate. We may now proceed to another example to unpack the relevant insights into the relationship between (linguistic) computation, culture and cognition.

They believe that Robin's land has a glossy look. (11)
They know that Robin's land has a glossy look. (12)

The sentence (11) has the presupposition that Robin has/owns a piece of land whereas (12) contains the presupposition that Robin's land has a glossy look. Let's now suppose that Robin refers to a person from a culture in which lands do not belong to individuals, rather individuals or groups of individuals belong to the land. That is indeed the case in many cultures across the world (as in, for example, Australian aboriginals). What happens in such a situation would be puzzling. Let's see how. The presupposition that is supposed to be computed from (11) by some program in the mind of such a person (having a native competence in English) would be that Robin, just as others in his surrounding culture do, belongs to a land a certain piece of which he happens to stay in and/or to simply live off. But the dilemma is that the presupposition that Robin has/owns a piece of land is supposed to be logically derived from this sentence and still not derived. If the purported linguistic computation in the mind is not sensitive to any conceptual/semantic/pragmatic content, any such program will get ensnared in a dilemma.

But as far as our experience is concerned, humans go about exploring their world, handle, produce and understand such expressions with whatever presupposition that comes forth in the relevant context. That is what makes cognition viable. If that is not what takes us so far, what if we push all feasible conceptual/semantic/pragmatic content into the computational space of a mental program computing linguistic rules and constraints? Needless to say that, such a move would be computationally intractable as the *frame problem* in artificial intelligence tells us. A similar dilemma arises with (12) as well. The presupposition that Robin's land has a glossy look is valid based on the nature of the verb 'know' which is differentiated from 'believe'. Even if knowledge is claimed to be justified true belief, knowledge pretty much works like belief in many ways when we shape and thereby are shaped by the socialization processes. Knowledge is extrapolated not merely from epistemic processes (objective observation) but also from cultural practices (norms, values); therefore it is not always obtrusive that truth can be conferred upon the intentional object of knowledge (as in 'I know that the sun revolves around the earth' where the presupposition that the sun revolves around the earth is as true as it is false when looked at from different spatiotemporal and phenomenological scales.).

In point of fact, the vector of the problem goes so deep that it transpires that the problems may have a liaison with the socio-cultural world that we co-construct and co-construe. In such a constitution, we often wallow in circular causality manipulated through our socio-cultural praxis as the paradoxes and contradictions emerging in connection with the phenomenon of long-distance dependency show us. For example, as we humans engage with the world, we mold, shape and manipulate symbolic capital (like wealth, honor, knowledge etc.) to respond to the socially constituted matrix of expectations, beliefs and practices [18]. But it is equally true to say that humans respond to the socially constituted matrix of expectations, beliefs and practices to mold, shape and manipulate symbolic capital (which becomes codified and objectified through that very stream of the intentional act of responding). Cognition grows in and with this reciprocal structuring of representational contours which, through that co-dependent growth, constitutes the very becoming of cognition itself. This is further related to how we intentionally ground our language processing in the wider realm of the socio-cultural praxis, which may have engendered the paradoxes with gapping too. What all else means can now be briefly drawn up below.

5. Final Remarks

Mainstream theoretical linguistics has long maintained that linguistic representations are (internalized) mental representations, and operations on such representations by means of rule systems are computations. This connects language to the human mind and computation. Beneath all this lie a number of deep-seated assumptions about the connection between language, cognition and computation. This paper shows that there does not appear to exist any determinate, coherent and consistent way in which computations can be both cognitive and linguistic or cognition cannot be both cultural and computational if grammars for natural languages are assumed to be mentally represented in a non-intentional manner. That is, if the mentally represented grammar is supposed to be not an intentional object, this invites fiendish and formidably recalcitrant paradoxes, puzzles and inconsistencies at points of contact of grammar, culture and cognition. This has been demonstrated here by way of examination of a number of standard and familiar linguistic phenomena such as *Wh*-movement, gapping, presupposition. More poignantly, this strongly indicates that the notion of interpretation applied to syntactic and semantic phenomena cannot be the notion of a frozen entity that comes attached to syntactic/semantic objects in that interpretation often as a cultural process of the mind creeps into everything syntactically and semantically constructed, and by virtue of that, interpretation cannot be both a form of (mental) computation outside of the architecture of grammar and a part of syntactic/semantic objects in linguistic constructions. Thus it also transpires that computation, especially digital computation, applied to mental operations over linguistic structures is either vacuous if employed in its general algorithmic sense or is full of inconsistencies if juxtaposed with mental interpretation in the context of the cultural scaffolding of cognition. The present work argues that these problems arise because of certain unacknowledged assumptions about intentionality, interpretation, mentality and computation

If human cognition is ultimately viable, that is what makes us re-think where culture and language (dis)place each other. If it turns out that computation is also one of the ways of intentional grounding made viable by language-culture co-construction, the exact relations in terms of which language, computation and culture are manifested, instantiated and thus project realizational possibilities with respect to one another become more convoluted than has been so far assumed.

Acknowledgements

We would like to thank two anonymous reviewers of ICCS2013 for their helpful suggestions and useful comments on the contents of the paper.

References

[1] Chomsky N. *The minimalist program*. Cambridge, Mass: MIT Press; 1995.

[2] Chomsky N. Derivation by phase. In: Kenstowicz M, editor. *Ken Hale: A life in language.* Cambridge, Mass: MIT Press; 2001, pp. 1-54.

[3] Piccinini G, Scarantino A. Information processing, computation and cognition. *Journal of Biological Physics* 2011; **37**, 1-38.

[4] O'Brien G, Opie J. Representation in analog computation. In: Newen A, Bartels A, Jung E, editors. *Knowledge and representation*. Stanford: CSLI Publications; 2011.

[5] Foster, C L. *Algorithms, abstraction and implementation: Levels of detail in cognitive science*. San Diego: Academic Press; 1992.

[6] Moortgart, M. Categorial type logics. In: van Benthem J, Ter Meulen A, editors. *Handbook of logic and language*. Amsterdam: Elsevier; 1997.

[7] Kempson R, Meyer-Viol W, Gabbay D. *Dynamic syntax: The flow of language understanding*. Oxford: Blackwell; 2001.

[8] Langendoen D T, Postal P. *The vastness of natural languages*. Oxford: Blackwell; 1984.

[9] Jackendoff R. *Foundations of language: Brain, meaning, grammar, evolution*. New York: Oxford University Press; 2002.

[10] Kamp H, Reyle U. *From discourse to logic*. Dordrecht: Kluwer; 1993.

[11] Harder P. *Meaning in mind and society*. Berlin: Mouton de Gruyter; 2010.

[12] Truswell R. *Events, phrases, and questions*. New York: Oxford University Press; 2011.

[13] Agbayani B, Zoerner E. Gapping, pseudogapping and sideward movement. *Studia Linguistica* 2004; **58(3)**, 185-211.

[14] Steedman M. Gapping as constituent coordination. *Linguistics and Philosophy* 1990; **13(2)**:207-263.

[15] Stalnaker R. Common ground. *Linguistics and Philosophy* 2002; **25(5-6)**: 701-721.

[16] Sauerland U. Implicated presuppositions. In: Steube A, editor. *Sentence and context: Language, context, and cognition*. Berlin: Mouton de Gruyter; 2008.

[17] Schlenker P. *Maximize presupposition and Gricean reasoning*. Unpublished manuscript.

[18] Bourdieu P. *Practical reason: On the theory of action*. Stanford: Stanford University Press; 1998.