Metadata of the chapter that will be visualized in SpringerLink

Book Title	Language Production, Cognition, and the Lexicon
Series Title	6636
Chapter Title	Recursion and Ambiguity: A Linguistic and Computational Perspective
Copyright Year	2014
Copyright HolderName	Springer International Publishing Switzerland

Corresponding Author	Family Name	**Delmonte**
	Particle	
	Given Name	**Rodolfo**
	Prefix	
	Suffix	
	Division	
	Organization	Computer Linguistics Laboratory
	Address	Venice, Italy
	Email	delmont@unive.it

Abstract

In this chapter I will be concerned with what characterizes human language and the parser that computes it in real communicative situations. I will start by discussing and dismissing Hauser et al. (2002) (HC&F) disputed claim that the "only uniquely human component of the faculty of language" be "recursion". I will substantiate my rejection of HC&F's claims, with the fact that recursion only appears in mature and literate language—an opinion also shared by some papers in a book on recursion by Harry van der Hulst (2010). I will then present in detail Chomsky's proposal—now part of the Minimalist Theory (MT)—of the architecture of the human parser as being based on Phases. I will accept this part of the theory and compare it with the computational architecture contained in a system for deep text understanding called Getaruns (Delmonte 2007, 2009). I will then argue in favour of what I regard the peculiar component of human language faculty that is "the ability to associate meaning to deficient propositions and generic linguistic expressions, which are highly ambiguous in their structure". And this is also due to the presence of recursion (but not only). I will then speak in favour of a parser that takes "context'' into account and strives for a "complete" syntactic representation. As to the problem of ambiguity, I will introduce the use of a computational device, a lookahead mechanism, which is presented in association with the need to specify UG parameters characterizing a given language. I will discuss the use of psychologically viable Parsing Strategies implemented in the parser to overcome ambiguity and prevent Garden Path, whenever possible. This will be highlighted by reference to peculiar features associated to different languages, Italian and English. Eventually, I will present a theory that encompasses all my previous proposals and is called LSLT.

Keywords (separated by '-'): Human language faculty - Human language parser - Recursion - Linguistic maturation - Natural language understanding - Parsing arguments and adjuncts - Ambiguity and meaning - Lookahead and parsing strategies - Garden path structures universal grammar parameters - Lexical semantic language theory (LSLT)

Recursion and Ambiguity: A Linguistic and Computational Perspective

Rodolfo Delmonte

Abstract In this chapter I will be concerned with what characterizes human language and the parser that computes it in real communicative situations. I will start by discussing and dismissing Hauser et al. (2002) (HC&F) disputed claim that the "only uniquely human component of the faculty of language" be "recursion". I will substantiate my rejection of HC&F's claims, with the fact that recursion only appears in mature and literate language—an opinion also shared by some papers in a book on recursion by Harry van der Hulst (2010). I will then present in detail Chomsky's proposal—now part of the Minimalist Theory (MT)—of the architecture of the human parser as being based on Phases. I will accept this part of the theory and compare it with the computational architecture contained in a system for deep text understanding called Getaruns (Delmonte 2007, 2009). I will then argue in favour of what I regard the peculiar component of human language faculty that is "the ability to associate meaning to deficient propositions and generic linguistic expressions, which are highly ambiguous in their structure". And this is also due to the presence of recursion (but not only). I will then speak in favour of a parser that takes "context'' into account and strives for a "complete" syntactic representation. As to the problem of ambiguity, I will introduce the use of a computational device, a lookahead mechanism, which is presented in association with the need to specify UG parameters characterizing a given language. I will discuss the use of psychologically viable Parsing Strategies implemented in the parser to overcome ambiguity and prevent Garden Path, whenever possible. This will be highlighted by reference to peculiar features associated to different languages, Italian and English. Eventually, I will present a theory that encompasses all my previous proposals and is called LSLT.

R. Delmonte (✉)
Computer Linguistics Laboratory, Venice, Italy
e-mail: delmont@unive.it

N. Gala et al. (eds.), *Language Production, Cognition, and the Lexicon*,
Text, Speech and Language Technology 48, DOI 10.1007/978-3-319-08043-7_15

Keywords Human language faculty · Human language parser · Recursion · Linguistic maturation · Natural language understanding · Parsing arguments and adjuncts · Ambiguity and meaning · Lookahead and parsing strategies · Garden path structures universal grammar parameters · Lexical semantic language theory (LSLT)

1 Introduction

Recursion has received a lot of attention lately, after HC&F's (2002) article which claims that recursion is the "only uniquely human component of the faculty of language", and the book edited by Harry van der Hulst (2010) which collects papers focusing on the same issue from a variety of different perspectives. The book is a mixture of different theories, levels of analysis, perspectives and points of view, none of which addresses the issue in the same manner as will be done here.

I will develop the notion of recursion from a distinct point of view which encompasses both properties of the language faculty and of the parser, this one simulated by a real parser, the central component of the system called GETA-RUNS. My point of view is partly hinted at in A. Verhagen's chapter, where he points to two important notions: long distance dependencies, and center-embedding, which is phrased as "the specification of certain phrases which requires the application of a rule to its own output". It is just these two elements of a computational architecture of the human/computational parser that make recursion a highly specialized and complex phenomenon. Looked at from inside the parser it is possible to distinguish two basic types of recursion as Pinker and Jackendoff put it, tail recursion and true recursion (2005, p. 211). Tail recursion can be mimicked by iteration, it is said, but as will be shown below, it requires coping and solving ambiguities and the attachment problem. True recursion on the contrary coincides with clausal recursion and the problem of long-distance dependencies. This latter case requires a computational device with a stack and pointers: but it is much more than that. Sentences are the only structure in a parser that cannot be fed on the output of previous computation. Sentences are semantically closed structures: in our case, extraposition variables for long distance dependencies may be passed on to the next structure in the case of complement clauses, or they may be passed inside a clause in the case of relative clauses. But the essence is that semantically speaking, finite clauses are totally independent of previous computation.

I will use the term "recursion" to refer to a syntactic property of sentence-level constructions focussing only to two types of syntactic constructions: sentential complements and relative clauses. Neither adverbial nor subordinate clauses will be taken into account, because in fact they do not constitute real embedded recursive structures. Adverbial clauses like temporal, concessive, causal and other similar subordinate sentence structures prevent the existence of long distance dependencies between a preceding and a following phrase or sentence. Besides, only sentential level recursion is able to generate semantically plausible—but the higher limit of clause embedding in real performance cases is language dependent and is however equal or below 3 (see Karlsson, 63)—infinite grammatical constructions: in this sense it is only sentence structure that is strictly speaking linked to recursion as a unique element of human language faculty. For instance in

- John said that it will rain yesterday

the adverbial can be bound to the main higher clause. But if we add an adverbial clause then the dependency is no longer possible,

- *John wanted to come because it will rain yesterday

As Verhagen comments in his chapter, recursion is relevant for grammar only for some rather specific phenomena, and it may as well be a product of cultural evolution which involves literacy, rather than be an intrinsic part of genetic evolution as Hauser and Chomsky maintain. We assume it may only come about as a consequence of linguistic maturation and triggered by the need to satisfy communicative goals in highly articulated conversational exchanges—more on this below.

From a linguistic point of view, neither constructions can be regarded a product of lexical selection: relative clauses are totally independent being adjuncts in nature. As to sentential complements, they are selected as such by certain communication verbs, but semantically speaking they are "closed" complements in the sense that they do not share any internal element with a higher governor or controller seen that a sentence or a tensed clause is a semantically independent propositional structure. In this sense, they are syntactic structure which may be motivated by semantic and pragmatic triggers: by the need to identify and describe referents and to report other people's utterances, or as commented again in Verhagen (ibid.:102), "perspective-taking a cognitive capacity—putting oneself in someone else's shoes, thus ascribing to them one's own cognitive capacities, including perspective taking—that implies recursivity". Further on, Verhagen (ibid.:103) links maturational factors with the increase of writing abilities as the main trigger of clause embedding, and frequency criteria that make "type-frequency of complement taking predicates to increase in people's linguistic experience".

Besides, we believe that the most distinctive ability humans show in their use of natural language for communication purposes, is syntactic and semantic disambiguation. This ability is usually ascribed to the existence of a "context'' (see Kuhn 2013), a general term that encompasses, amongst other things, elliptical/unexpressed and/or implicit/entailed/implicated linguistic material, presumed intentions and aims of the interlocutor/s, and encyclopaedic knowledge of the world. In the best current systems for natural language, the linguistic components are kept separate from the knowledge representation, and work which could otherwise be done directly by the linguistic analysis is duplicated by the inferential mechanism. The linguistic representation is usually mapped onto a logical representation which is in turn fed onto the knowledge representation of the domain in order to understand and validate a given utterance or query. Thus the domain world model or ontology must be priorly built, usually in view of a given task the system is set out to perform. This modeling is domain and task limited and generality can only be achieved from coherent lexical representations. We assume that access to knowledge should be filtered out by the analysis of surface linguistic forms and their abstract representations of the utterances making up the text. However, we have to admit that world knowledge can be an integral part of the parsing process only in specific domains. No system is yet able to account for all the unexpressed linguistic material that is nonetheless essential for the complete semantic representation of a text or dialogue. We will discuss some of these unexpressed linguistic materials. The appropriate definition of the Context to be used here is the one related to the existence of a rich lexical representation. Consider now some ambiguous examples taken from P. Bosch (246), which cover some types of ambiguity:

1. Five companies sold two hundred installations
2. Fred saw the woman with the binoculars
3. Visiting relatives can be boring
4. Pete went to the bank this morning
5. This paper is ten pages long
6. Faulkner is hard to understand
7. a. The school made a major donation
   b. The school has a flat roof
   c. He enjoys school very much
   d. School is one of the pillars of our civilization

Ambiguity problems to be tackled are not all the same, as can be noticed. In 1. we have a problem of quantifier scope, which we think is only solvable by allowing a Quantifier Raising (hence QR) module produce two different representations for the same f-structure and then letting the semantic/pragmatic component do the rest. In our system, QR would compute as a preferential reading the one in which the subject NP takes scope over the object NP when both are numerals. In case the ambiguity had to be solved in favour of the second reading a distributive floating quantifier (each) should have been added. In 2 and 3 the parser would have come out with the most likely interpretation of the sentence and that might very well happen to be the wrong one: however the feeling one gets when discussing such sentences, is that they are very unlikely to be found in real texts—or at least this is what we assume. If we consider in more detail example 2, we could come up with common sense knowledge that prevents “binoculars” to be computed as an adequate adjunct of the head “woman”. To be sure, this is what at present our system does, and assigns it rather as a predicative (Instrumental) complement of the verb SEE. However, there might be special scenarios in which women walk around carrying binoculars around their neck: this does not happen in

Venice(Italy), where everything can be comfortably looked at without binoculars, but could happen in the Grand Canyon where distances require it.

As to 4. we are in presence of a case of ambiguous lexical semantics, and represents a typical case of Word Sense Disambiguation. We should note here that since the semantic role associated to the PP "to the bank" would always be Locative, disregarding its semantic features, the actual meaning or sense associated to the noun "bank" could be easily accommodated by the semantic/pragmatic component, and this would in no way affect syntactic analysis. It is also important to note that the system may have to look for the most adequate referent to the singular definite NP, in the "Discourse Model"—a semantic/pragmatic module of Getaruns (See Delmonte 2007). The same applies to example 5 and 6, provided that a "Discourse Model" is available in the system where previous referents with additional information may be searched for.

Coming now to the last set of examples, where the "school" is assigned different meanings according to context. Here, we may easily assume that the system of linguistic description—WordNet in this case—should cover the whole of them. In a. the school is the SUBJect of the predicate MAKE and this requires an Agent which may be a Social_Institution, but not an object, i.e. a building, as is required by the meaning of the b. example. In this case, the meaning is not conveyed by the verb BE which has non-thematic arguments, but by contents of the predicate NP "the flat roof": this would be classified as [object, part_of], thus implying that the predication requires also an Object as its controller. In c. we have a psych verb "enjoy" which has an EXPERIENCER as SUBJect NP and a CAUSER_EMOT as OBJect NP, in our lexicon—but see also VerbNet and PropNet. The school in this case will be assigned a semantic value by the semantic roles associated to verb predicate, as happened with the a. example. The same applies to the c. example. In other words, it is the linguistic description which enables the semantic interpreter do its job properly by means of the conjoined information made available by semantic roles and semantic features.

The chapter is organized as follows: the second section below presents the parsers and argues for its being compliant with Phases as proposed by Chomsky; in Sect. 3. we present the parser; in Sect. 4. we discuss the psycholinguistic and cognitively founded parsing strategies; in 5 an evaluation and a conclusion.

2 The Parser and Phases

In the last 10 years or so Chomsky has been referring to the human parser as a gauge of the way in which syntactic processes are carried out in the mind of the language user. The parser has also been referred to as a metaphor of grammatical processes underlying sentence comprehension as is being purported within the current Minimalist Theory (hence MT). This interest for performance related notions can be regarded as an attempt on Chomsky's side to support/endow MT with a psychological and computational basis, thus making MT a unified theory for language. However, a parser based on any linguistic theory that aims to realize such a goal, should account also for the determining factor in sentence comprehension, that is ambiguity. This in turn is the cause of Garden Path on the one side, and on the other it motivates the existence of parsing preferences in the human processor. So, in the last resort, a theory that aims at the explanation of performance facts should satisfy three different types of requirements: psycholinguistic plausibility, computational efficiency in implementation, coverage of grammatical principles and constraints. This is also what a parser should satisfy, but see below.

It is plausible to say that for the first time performance facts can be brought to bear on theoretical assumptions based on competence. In fact this is also what HCF02 seem to be aiming at.[1] In particular, the notion of Phases will be used in this chapter to test its validity in coping with the effects of ambiguity and Garden-Path related examples. We will try to show that the way in which Phases have been formulated is plausible, even though their final status and their theoretical status are still under debate (see Svenonius 2001, 2004; Legate 2002, 2003; Epstein 2004). But it is both too strong and too weak. Rather than Phases, we will be referring to a related and/or derivable notion, that of Argument and Adjunct as the semantically complete object of any step the parser should pursue in its process of analysis. Sentence level parsing requires in turn a first clause-level preliminary structure (something close to pseudo-syntax, as Townsend and Bever (2001) (hence T&B) call it), which is then submitted to proper interpretation—and possible LF computation, before interacting with higher than clause level for complex sentences, which can eventually license the parser output for PF.

To reduce computational complexity Chomsky (1998/2000) introduced the idea of Phases—units of syntactic computation—within MT. The general idea is that of limiting the burden of syntactic operations in order to ease workload or what must be retained in active memory. This seems a counterargument to the fact that human language (actually FLN) exhibits an intrinsic defining property that makes computation hard, that of recursion (HCF02). So it would seem that Chomsky's concern in proposing Phases is double-fold: on the one side it is motivated by performance related issues, on the other hand it is coupled to theory internal motivations. In fact, we will only tackle performance questions and not questions affecting the Minimalist Program. We would like to prove Phases to be a theory-independent principle governing the functioning of the human parser which we will investigate from a psycholinguistic and a computational point of view. The human parser is so efficient that it must obey some principle-based criterion in coping with recursion: Phases are Chomsky's solution.

[1] "Recent work on FLN suggests the possibility that at least the narrow-syntactic component satisfies conditions of highly efficient computation to an extent previously unsuspected…. [T]he generative processes of the language system may provide a near-optimal solution that satisfies the interface conditions to FLB. Many of the details of language that are the traditional focus of linguistic study … may represent by-products of this solution, generated automatically by neural/computational constraints and the structure of FLB—components that lie outside of FLN." (Note that FLN stands for Faculty of Language Narrow, and FNB stands for Faculty of Language Broad).

Constituency-based parsing models are lately starting to be supplanted by word-level parsing models in the vein of Dependency-Based parsing (see Kuhn 2013). These parsers are organized in such a way as to limit the scope of syntactic operations to adjacent head-dependent word pairs. Recursion is thus eliminated from the grammar and computational efficiency is usually guaranteed. The same applies to bottom-up cascaded ATN-like parsers, which decompose the task of syntactic structure building into a sequence of intermediate steps, with the goal of avoiding recursion as much as possible. However, both coverage, precision and recall don't speak in favor of such parsers which working bottom-up adopt parsing policies which are not strictly left-to-right. In this respect, we believe that a parser should embody a psycholinguistically viable model, i.e. it should work strictly left-to-right and be subject to Garden Path effects. We also believe that by eliminating constituents from the parsing process, and introducing the notion of Head-Dependent relations, grammaticality principles may become harder to obey. Parsers today are required to produce a semantically interpretable output for any text: in order to achieve such a goal, Grammatical Relations need to be assigned to words in some kind of hierarchical (constituent-based) representation, before some Logical Form can be built. Word-based head-dependent parsers are not good candidates for the generation of such an output. In fact, no implicit categories are usually computed by such parsers, hampering in this way any semantic mapping from taking place (See Delmonte 2013a, b).

2.1 Phases and Semantic Mapping

Quoting from Chomsky,

> A phase is a unit of syntactic computation that can be sent to the Spell-Out. Syntactic computation proceeds in stages: a chunk of structure (a vP or a CP) is created and then everything but its edge can be sent off to the interfaces.

Phases are semantically complete constituents or "complete propositions", which could be independently given a Logical Form and a Phonetic Form. Carnie and Barss (2006) propose to relativize the definition of phase to that of Argument which we subscribe fully here below: in their words,

"Each phase consists of an argument, the predicative element that introduces the argument (V or vP) and a functional category that represents a temporal operator which locates the predicate in time or space (Asp, T, etc.). Phases consist of:

(a) a predicative element (v or V)
(b) a single argument
(c) a temporal operator that locates the predicate and argument in time and space (Asp or T)"

To this definition we will add the need to regard arguments as semantically complete constituents with their adjuncts and modifiers, something which is asserted by Epstein (2004) and introduced in Chomsky 2004b, when they assume that the specification of a phase has "full argument structure". In addition this could be derived where they assume that a partial LF could be produced. It goes without saying that in order to produce a partial or complete LF from syntactic chunks, they need to be semantically interpretable: this includes semantic role assignment, being exempt from quantificational related problems like the presence of unbound variables. The LF we are referring to is a flat version with unscoped quantifiers.

In line with Pinker and Jackendoff's paper (hence P&J 2005) produced as an answer to HCF02, we assume that lexical information is the most important static knowledge source in the processing of natural language. However, we also assume that all semantic information should be made to bear on the processing and this is only partially coincident with lexical information as stored in lexical forms. In particular, subcategorization, semantic roles and all other semantic compatibility evaluative mechanisms should be active while parsing each word of the input string. In addition, Discourse Model and External Knowledge of the World should be tapped when needed to do coreference and anaphora resolution. Antecedents in turn would be chosen on the basis of grammatical information like Grammatical Relations and Semantic Roles, and not independently of it.

In that perspective, we believe that a sound parsing strategy should opt for a parser that strives for an even higher than constituent semantically closer level: i.e. arguments and adjuncts, where mixed/hybrid strategies (bottom-up and top-down) are activated by the use of a strongly language-dependent lookahead mechanism. We would like to speak in favour of such an approach in which locality is sacrificed for a mixed or hybrid model, partially bottom-up, which uses both grammatical function driven information and lexical information from subcategorization frames to direct the choices of the argument vs adjunct building parsing process.

On the one side we endorse a position purported by linguistic theories like MT which require LF licensing of constituency at some level—and clause level Phases are here assumed as the only possible counterpart to LF; on the other side, we speaks against MT as in a sense—at least some MT linguist would accept it—Dependency Parsing implements it because it assumes that parsing cannot just be bottom-up word level parsing, but some top-down guidance is needed. Furthermore, neither MT nor Dependency Parsing would accommodate a strictly semantic and lexicalist notion like "Argument-Adjunct" parsing together with a performance related notion like ambiguity and the accompanying effect of Garden Path, which is familiar in psycholinguistic literature. In addition, language dependent rules would suit best an MT-like approach with parameters driven options or any other linguistic theory which allows rule of Core Grammar to be set apart from rules of the Periphery.

3 GETARUNS: An A-As Hybrid Parser

As commented above, to be Phase-compliant a parser needs to build up each constituent as a fully interpreted chunk with all its internal arguments and adjuncts if any. In this process, we know that there are two boundaries which need to be taken into account: the CP level and the Ibar level, where the finite verb is parsed. From a computational perspective we might paraphrase the concomitant contribution of the two Phases as follows:

v. parse all that comes before the finite verb and then reset your internal indices.

Our parser is not a dependency parser in that it imposes constituent-based global restrictions on the way in which words can be parsed: only legal constituents are licensed by the parser.

We defined our parser "mildly bottom-up" because the structure building process cycles on a call that collects constituents until it decides that what it has parsed might be analysed as Argument or Adjunct. To do that it uses Grammatical Function calls that tell the parser where it is positioned within the current parse. We use Grammatical Function because in LFG theory they are regarded as linguistic primitives. This proceeds until finite verb is reached and the parse is continued with the additional help of Verb Guidance by subcategorization information. The recursive procedure has access to calls collecting constituents that identify preverbal Arguments and Adjuncts including the Subject if any. When the finite verb is found the parser is hampered from accessing the same preverbal portion of the algorithm and switches to the second half of it where Object NPs, Clauses and other complements and adjuncts may be parsed. Punctuation marks are also collected during the process and are used to organize the list of arguments and adjuncts into tentative clauses.

When the parser reaches the Verbal Phrase the syntactic category associated to the main verb—transitive, unergative, unaccusative, impersonal, atmospheric, raising, psych, copulative—and the lexical form of the predicate, are both used as topdown guidelines for the surface realization of its arguments. Italian is a language which allows for empty or morphologically unexpressed Subjects, so that no restriction may be projected from the lexicon onto c-structure: in case it is empty, a little pro is built in subject position, and features are left as empty variables until the tensed verb is processed.

The clause builder looks for two elements in the input list: the presence of the verb-complex and punctuation marks, starting from the idea that clauses must contain a finite verb complex. Dangling constituents will be adjoined to their left adjacent clause, by the clause interpreter after failure while trying to interpret each clause separately. The clause-level interpretation procedure interprets clauses on the basis of lexical properties of the governing verb: verbless clauses are not dealt with by the bottom-up parser, they are passed down—after failure—to the top-down parser which can license such structures.

The final processor takes as input fully interpreted clauses which may be coordinate, subordinate, main clauses. These are adjoined together according to their respective position. Care is taken to account for Reported Speech complex sentences which require the Parenthetical Clause to become Main governing clause.

We opted to deal with Questions and Imperatives with the top-down parser rather than with the bottom-up one. Also sentences with Reported Direct speech are treated in that way due to the presence of inverted commas that must be interpreted accordingly. Non-clausal Subject sentences and extraposed That-clause fronted sentences are also computed top-down. The advantage of using fully top-down processing is that the clause-building stage is completely done away with. The parser posits the clause type as a starting point, so that constituents are searched for and collected at the same level in which the parsing has started. However, this is only conceivable in such non-canonical structures as the ones listed here above.

If the parser does not detect any of the previous structures, control is passed to the bottom-up/top-down parser, where the recursive call simulates the subdivision of structural levels in a grammar. All sentential fronted constituents are taken at the CP level and the IP (now TP) level is where the SUBJect NP must be computed. Otherwise SUBJect NP will be either in postverbal position with Locative Inversion structures, or the parser might be trying a subjectless coordinate clause. Then again a number of ADJuncts may be present between SUBJect and verb, such as adverbials and parentheticals. When this level is left, the parser is expecting a verb in the input string. This can be a finite verb complex with a number of internal constituents: but the first item must be definitely a tensed verb. After the (complex) verb has been successfully built, the parser looks for complements: the search is restricted by lexical information. If a copulative verb has been taken, the constituent built will be labelled accordingly as XCOMP where X may be one of the lexical heads, P,N,A,Adv.

The clause-level parser simulates the sentence typology where we may have as SUBJect a verbal clause, an Inverted postverbal NPs, a fronted that-clauses, and also fully inverted OBJect NPs in preverbal position. We do that because we purport the view that the implementation of sound parsing algorithm must go hand in hand with sound grammar construction. Extragrammaticalities can be better coped with within a solid linguistic framework rather than without it.

The parser has a manually-built grammar and is written in Prolog, a programming language that provides for backtracking freely and has a variable passing mechanism useful to cope with a number of well-known grammatical problems like agreement (local and non-local) as well as Long-Distance Dependencies. The parser is a rule-based deterministic parser in the sense that it uses a lookahead and a Well-Formed Substring Table to reduce backtracking. It also implements Finite State Automata in the task of tag disambiguation, and produces multiwords whenever lexical information allows it. Recovery procedures are also used to cope with elliptical structures and uncommon orthographic and punctuation patterns. In particular, the parser is written in Prolog Horn-clauses and uses Extraposition variables to compute Long-Distance Dependencies.[2]

[2] We use XGs (extraposition grammars) introduced by Pereira (1981, 1983). Prolog provides naturally for backtracking when allowed, i.e. no cut is present to prevent it. Furthermore, the instantiation of variables is a simple way for implementing the mechanism for feature percolation and/or for the creation of chains by means of index inheritance between a controller and a controllee, and in more complex cases, for instance in case of constituent ellipsis or deletion. Apart from that, the grammar implemented is a surface grammar of the chosen languages.

Being a DCG, the parser is strictly a top-down, depth-first, one-stage parser with backtracking. Differently from most principle-based parsers presented in Berwick et al. (1991), which are two-stage parsers, our parser computes its representations in one pass. This makes it psychologically more realistic. The final output of the parsing process is an f-structure which serves as input to the binding module and logical form: in other words, it constitutes the input to the semantic component to compute logical relations. In turn the binding module may add information as to pronominal elements present in the structure by assigning a controller/binder in case it is available, or else the pronominal expression will be available for discourse level anaphora resolution. AQ1

Grammatical functions are used to build f-structures and the processing of pronominals. They are crucial in defining lexical control: as in Bresnan (1982, 2001), all predicative or open functions are assigned a controller, lexically or structurally. Lexical control is directly encoded in each predicate-argument structure, and it will bind the empty subject of all predicative open functions built in all predicative structures (or small clauses) to the appropriate syntactic controller (or binder).

The parser is made up of separate modules:

1. The Grammar, based on DCGs, incorporates Extraposition to process Long Distance Dependencies, which works on annotated c-structures: these constitute the output to the Interpretation Module;
2. The Interpretation Module checks whether f-structures may be associated to the input partially annotated c-structure by computing Functional Uniqueness, Coherence and Completeness. Semantic roles are associated to the input grammatical function labels at this level, after semantic selectional restrictions are checked for membership;
3. The Mapping scheme, to translate trees into graphs, i.e. to map c-structures onto f-structures. The parser builds annotated c-structure, where the words of the input sentence are assigned syntactic constituency and functional annotations. This is then mapped onto f-structure, i.e. constituent information is dropped and DAGs are built in order to produce f-structure configuration.

3.1 Parsing Ambiguities Coping with Recursion

The lexicon as the source of syntactic variation is widely accepted in various theoretical frameworks. We assume that be it shallow or deep, parsing needs to be internally parameterized in order to account for ambiguities generated both at structural and at semantic level.

As said above, a parser that achieves psychological reality should closely mimic phenomena such as Garden Path effects, or an increase in computational time in presence of semantically versus syntactically biased ambiguous structures. We also assume that a failure should ensue from strong Garden Path effects and that

Layout: T1 Standard SC Book ID: 323543_1_En Book ISBN: 978-3-319-08042-0
Chapter No.: 15 Date: 27-6-2014 Time: 7:38 pm Page: 12/28

this should be justified at a psycholinguistic interpretation level. In other words, looking at parsing from a performance-based perspective, the parser should anticipate ambiguities that may cause unwanted Garden-Paths and Crashes, in order to refrain from unwanted failures in order to mimic human processing. But how should a "sound" parser be told which ambiguous structures are expected in which language?

In general terms, ambiguity is generated by homophonous words in understanding activities and by homographs in reading activities. In both cases Garden Paths or Crashes may only result in a given language in presence of additional conditions which are strictly dependent on the structure of the lexicon and the grammar. But some UG related parameters, like the "OMISSIBILITY OF THE COMPLEMENTIZER" in English may cause the parser to crash or freeze. Generally speaking, all types of ambiguity affecting parsing at a clause level will cause the parser to go into a Garden Path. Developing this line of thought, we assume that from a psycholinguistic point of view, parsing requires setting up a number of disambiguating strategies, like for instance telling arguments apart from adjuncts and reducing the effects of backtracking. And this is how it has been implemented.

Whenever a given predicate has expectancies for a given argument to be realized either optionally or obligatorily, this information will be passed below to the recursive portion of the parsing process: this operation allows us to implement parsing strategies like Minimal Attachment, Functional Preference and other ones (See Delmonte 2009).

The DCG grammar allows the specification of linguistic rules in a highly declarative mode: it works topdown and by making a heavy use of linguistic knowledge may achieve an almost complete deterministic policy. Parameterized rules are scattered throughout the grammar so that they can be made operative as soon as a given rule is entered by the parser. In particular, a rule may belong either to a set of languages, e.g. Romance or Germanic,[3] or to a subset thereof, like English or Italian, thus becoming a peripheral rule. Rules are activated at startup and whenever a switch is being operated by the user, by means of logical flags appropriately inserted in the right hand side of the rule. No flags are required for rules belonging to the common core grammar.

Some such rules include the following ones: for languages like Italian and Spanish, a Subject NP may be an empty category, either a referential little pro or an expletive pronoun; Subject NPs may be freely inverted in postverbal position, i.e. preverbal NP is an empty category in these cases. For languages like Italian and French, PP or adverbial adjuncts may intervene between Verb and Object NP; adjectival modifiers may be taken to the right of their head Noun. For languages like English and German, tense and mood may be computed in CP internal

[3] As to multilinguality, the basic tenet of the parser is based on a UG-like perspective, i.e. the fact that all languages share a common core grammar and may vary at the periphery: internal differences are predicted by parameters.

Layout: T1 Standard SC | Book ID: 323543_1_En | Book ISBN: 978-3-319-08042-0
Chapter No.: 15 | Date: 27-6-2014 Time: 7:38 pm | Page: 13/28

position, when taking the auxiliary or the modal verb. English allows an empty Complementizer for finite complement and relative clauses, and negation requires do-support. Italian only allows it for a highly genre marked (literary style) untensed auxiliary in Comp position.

Syntactic and semantic information are accessed and used as soon as possible: in particular, both categorial and subcategorization information attached to predicates in the lexicon is extracted as soon as the main predicate is processed, be it adjective, noun or verb, and is used to subsequently restrict the number of possible structures to be built. Adjuncts are computed by semantic compatibility tests on the basis of selectional restrictions of main predicates and adjuncts heads.

Thus, we build and process syntactic phenomena like wh-movement before building f-structure representations, where quantifier raising and anaphoric binding for pronominals takes place. In particular, all levels of Control mechanisms which allow coindexing at different levels of parsing give us a powerful insight into the way in which the parser should be organized. In addition, we find that topdown parsing policies are better suited to implement parsing strategies that are essential in order to cope with attachment ambiguities. Also functional Control mechanisms—both structural and lexical—have been implemented as close as possible to the original formulation, i.e. by binding an empty operator in the subject position of a propositional like open complement/predicative function, whose predicate is constituted by the lexical head.

3.2 Lookahead and Ambiguity

Lookahead is used in a number of different ways: it may impose a wait-and-see policy on the topdown strategy or it may prevent following a certain rule path in case the stack does not support the first or even second match:

a. to prevent expanding a certain rule
b. to prevent backtracking from taking place by delaying retracting symbols from input stack until there is a high degree of confidence in the analysis of the current input string.

It can be used to gather positive or negative evidence about the presence of a certain symbol ahead: symbols to be tested against the input string may be more than one, and also the input word may be ambiguous among a number of symbols. Since in some cases we extend the lookahead mechanism to include two symbols and in one case even three symbols, possibilities become quite numerous. The following list of 14 preterminal symbols is used (Table 1):

As has been reported in the literature (see Tapanainen and Voutilainen 1994; Brants and Samuelsson 1995), English but also Italian (see Delmonte 1999) is a language with a high level of homography: readings per word are around 2 (i.e. each word can be assigned in average two different tags depending on the tagset).

Table 1 Preterminal symbols used for lookahead

1. v = verb-auxiliary-modal-clitic-cliticized verb
2. n = noun—common, proper;
3. c = complementizer
4. s = subordinator;
5. e = conjunction
6. p = preposition-particle
7. a = adjective;
8. q = participle/gerund
9. i = interjection
10. g = negation
11.d = article-quantifier-number-intensifier-focalizer
12. r = pronoun
13. b = adverb
14. x = punctuation

Lookahead in our system copes with most cases of ambiguity: however, we also use disambiguating before passing the input string to the parser.

Consider now failure and backtracking which ensues from it. Technically speaking, by means of lookahead we prevent local failures in that we do not allow the parser to access the lexicon where the input symbol would be matched against. It is also important to say that almost all our rules satisfy the efficiency requirement to have a preterminal in first position in their right-hand side. Cases like complementizerless sentential complements are allowed to be analysed whenever a certain switch is activated. Suppose we may now delimit failure to the general case that may be described as follows:

- a constituent has been fully built and interpreted but it is not appropriate for that level of attachment: failure would thus be caused only by semantic compatibility tests required for modifiers and adjuncts or lack of satisfaction of argument requirements for a given predicate. Technically speaking we have two main possibilities:

A. the constituent built is displaced on a higher level after closing the one in which it was momentarily embedded. This is the case represented by the adjunct PP "in the night" in example below:

(8) The thieves stole the painting in the night.

The PP is at first analysed while building the NP "the painting in the night" which however is rejected after the PP semantic features are matched against the features of the governing head "painting".

B. the constituent built is needed on a lower level and there is no information on the attachment site. In this case a lot of input string has already been consumed before failure takes place and the parser needs to backtrack a lot before constituents may be safely built and interpreted. This is the case of an NP analysed as OBJect of a higher clause but is needed as SUBJect of a following clause.

To give a simple example, suppose we have taken the PP "in the night" within the NP headed by the noun "painting". At this point, the lookahead stack would be set to the position in the input string that follows the last word "night". As a side-effect of failure in semantic compatibility evaluation within the NP, the PP "in the night" would be deposited in the backtrack WFST storage. The input string would be restored to the word "in", and analysis would be restarted at the VP level. In case no PP rule is met, the parser would continue with the input string trying to terminate its process successfully. However, as soon as a PP constituent is tried, the storage is accessed first, and in case of non emptiness its content recovered. No structure building would take place, and semantic compatibility would take place later on at sentence level. The parser would only execute the following actions:

- match the first input word with the (preposition) head of the stored term;
- accept new input words as long as the length of the stored term allows it by matching its length with the one computed on the basis of the input words.

Differences in reanalysis are determined by structural requirements and by analysis load imposed on the parser by backtracking: in case a sentential adjunct has to be destroyed/broken up and reconstructed it represents a far lighter load than a subordinate/main clause. Let's say, that whenever a clausal structure has to be destroyed/broken up a whole set of semantic decisions have to be dismantled, and structure erased.

4 Linguistically-Plausible Relaxation Techniques

With the grammar above and the parameters we are now in a position to establish a priori positions in the parser where there could be recovery out of recursion with ungrammatical structures with the possibility to indicate which portion of the input sentence is responsible for the failure. At the same time, parsing strategies could be devised in such a way to ensure recovery from local failure. We will start by commenting on Parsing Strategies first and their implementation in our grammar[4] Differently from what is asserted by global or full paths approaches (see Schubert 1984; Hobbs et al. 1992), we believe that decisions on structural ambiguity should be reached as soon as possible rather than deferred to a later level of representation. In particular, Schubert assumes "…a full paths approach in which not only complete phrases but also all incomplete phrases are fully integrated into (overlaid) parse trees dominating all of the text seen so far. Thus features and partial logical translations can be propagated and checked for consistency as early as possible, and alternatives chosen or discarded on the basis of all of the available information (ibid., 249)." And further on in the same paper, he proposes a system of numerical 'potentials' as a way

[4] We also assume that a failure should ensue from strong Garden Path effects and that this should be justified at a psycholinguistic interpretation level (See Pritchett 1992).

Layout: T1 Standard SC | Book ID: 323543_1_En | Book ISBN: 978-3-319-08042-0
Chapter No.: 15 | Date: 27-6-2014 Time: 7:38 pm | Page: 16/28

of implementing preference trade-offs. “These potentials (or levels of activation) are assigned to nodes as a function of their syntactic/semantic/pragmatic structure and the preferred structures are those which lead to a globally high potential. Among contemporary syntactic parsing theories, the garden-path theory of sentence comprehension proposed by (Frazier 1987a, b), Clifton and Ferreira (1989) among others, is the one that most closely represents our point of view. It works on the basis of a serial syntactic analyser, which is top-down, depth-first—i.e. it works on a single analysis hypothesis, as opposed to other theories which take all possible syntactic analysis in parallel and feed them to the semantic processor. From our perspective, it would seem that parsing strategies should be differentiated according to whether there are argument requirements or simply semantic compatibility evaluation for adjuncts. As soon as the main predicate or head is parsed, it makes available all lexical information in order to predict if possible the complement structure, or to guide the following analysis accordingly. As an additional remark, note that not all possible syntactic structure can lead to ambiguous interpretations: in other words, we need to consider only cases which are factually relevant also from the point of view of language dependent ambiguities.

The parser has been built to simulate the cognitive processes underlying the grammar of a language in use by a speaker, taking into account the psychological nuances related to the well-known problem of ambiguity, which is a pervading problem in real text/communicative situation, and it is regarded an inseparable benchmark of any serious parser of any language to cope with.

We implemented in our parser a number of strategies that embody current intuitions on the way in which sentence comprehension mechanisms work at a psychological level. The parsing strategies are the following: Minimal Attachment/Late Closure (MA), Argument Preference (AP), Thematic Evaluation (TE), Referential Individuation (RI), Cross Compatibility Check (CCC). From the way in which we experimented them in our implementation it appears that they are strongly interwoven. In particular, MA is dependent upon AP to satisfy subcategorization requirements; with semantically biased sentences, MA and AP, and finally TE should apply in hierarchical order to license the phrase as argument or adjunct. RI seems to be required and activated every time a singular definite NP is computed. However, RI is a strategy that can only become operative whenever a full parse of possible modifiers is available and not before. In addition, subcategorization and thematic requirements have priority over referential identification of a given NP: a violation of the former is much stronger than the latter. Generally speaking, redundancies in referential properties might simply be accommodated by the speaker: but lack of consistency, uniqueness and completeness leads to ungrammaticality.

As discussed above, we follow a mixed topdown depth-first strategy which we believe better accounts for the way in which human psychological processes work. In order to prevent failures and control backtracking, depth-first analysis should be organized as much as possible deterministically. Nondeterminism can be very time consuming and it should be reduced or at least controlled according to the parsing strategy selected.

Layout: T1 Standard SC | Book ID: 323543_1_En | Book ISBN: 978-3-319-08042-0
Chapter No.: 15 | Date: 27-6-2014 Time: 7:38 pm | Page: 17/28

As Altmann (1989)[5] comments in his introduction (ibid.86), and we also believe, it is an empirical question whether the constraints assumed by the thematic processor (single initial syntactic analysis, semantic evaluation only within the domain of this analysis) are constraints actually observed by the parser, or whether a less-constrained mechanism that makes appeal to context and meaning at the earliest stages of sentence comprehension is a more adequate description of the true state of affairs. It is our opinion that all lower level constraints should work concurrently with higher level ones: in other words, all strategies are nested one inside another, where MA occupies the most deeply nested level. The higher level strategy has control over the lower level one in case some failure is needed. Suppose we have the following examples which can be disambiguated only at the level of pronominal binding.

i. The doctor called in the son of the pretty nurse who hurt herself.
ii. The doctor called in the son of the pretty nurse who hurt himself.

Pronominal binding is a level of computation that takes place after f-structure has been completely checked and built in LFG—the same applies in GB framework, where S-structure gives way to L-structure and this is where binding takes place. In this case however, it would be impossible to address the appropriate level of representation after destroying all previous structures with backtracking. In this case, backtracking by itself would be inefficient and would not assure termination—simply because the same structure could be constructed at sentence level. We assume, instead, that a specific mechanism should be activated before f-structure is licensed in order to check the presence of a reflexive pronoun, i.e. an anaphoric pronoun or short anaphora, that needs the SUBJect to be an appropriate antecedent, agreeing in all features with the anaphora itself (See Delmonte 2002).

The following two examples are also computed without any special provision for the ambiguous structural position of the final temporal adverbial, simply by matching semantic information coming from verb tense and temporal configuration associated to the adverbial in its lexical entries in terms of a precedence relation between td (discourse time), and tr (reference time). Thus, in the case of "tomorrow" the parser will have td < tr and the opposite will apply to "yesterday". In turn, this configuration is matched against tense, "past" or "future" and a failure will result locally, if needed.

iii. Mary will say that it rained yesterday.
iv. Mary said that it will rain yesterday.

[5] Altmann offers a functional argument against a system in which choices are initially made by a syntactic processor, and later corrected by appeal to meaning and context. He says that if referential or discourse information is available, only a strange processor would make decisions without appealing to it. However, syntactic information is always available and always informative.

4.1 Graceful Recovery Actions from Failures

As discussed above, recovery from garden-path requires a trial and error procedure, i.e. the parser at first has to fail in order to simulate the garden-path effect and then the recovery will take place at certain conditions. Now consider the well-known case of Reduced Relatives[6](9) The horse raced past the barn fell.is one such case. The English speaker will attempt treating the verb "raced" as main tensed verb, but on discovery of sentence final verb "fell" which can only be interpreted as tensed past tense the whole sentential level analysis crashes and a Garden Path ensues causing a complete restart of the mental parser. which have always been treated as a tough case (but see Stevenson and Merlo 1997). From an empirical point of view we should at first distinguish cases of subject attachment reduced relatives from all other cases, because it is only with subject level attachment that a garden-path will actually ensue. In fact, this is easily controllable in our parser, given the fact that NPs are computed by means of functional calls. In this way the information as to where the NP is situated in the current sentence analysis is simply a variable that is filled with one of the following labels: subj, obj, obj2, obl, adj, ncomp, where the last label stands for predicative open complements.

From a purely empirical point of view, we searched the WSJ corpus in order to detect cases of subject attachment vs all other cases for reduced relatives and we came up with the following figures: SUBJECT-ATTACHEMENT 530; OTHERS 2982; Total 3512. If we subtract present participle cases of reduced relatives which do not constitute ambiguous words the total number is lowered down to 340. Subject-attachment thus constitute the 9.68 % of all cases, a certainly negligible percentage. In addition, 214 of all subject-attachment are passive participles and lend themselves to easy computation being followed by the preposition "by". So there will reasonably be only 116 possible candidates for ambiguous reduced relatives. The final percentage comes down 3.3 % which is very low in general, and in particular when computed over the whole 1 million occurrences, it comes down to a non classifiable 0.0116 %. The same results can be obtained from an investigation of the Susanne Corpus, where we found 38 overall cases of reduced relatives with ambiguous past participles, 0.031 % which is comparable to the 0.035 % of the WSJ (Table 2).

If we look into matter closely, then we come up with another fairly sensible and easily intuitive notion for reduced relatives disambiguation: and it is the fact that whenever the governing Noun is not an agentive, nor a proto-agent in any sense of the definition (see Stevenson and Merlo), no ambiguity may arise simply because non agentive nominal governors may end up with an ambiguous interpretation only in case the verb is used as ergative. However, not all transitive verbs can be

[6] The typical example quoted in psycholinguistic literature is the reduced relative case, reported here below, determined by the lexical ambiguity of English verbs being at the same time interpretable as Past Participle—Past Tense and shown below in the Reduced Relative Clause well-known example,

Table 2 List of 27 verb-types used in WSJ in subject-attached reduced relatives

Accused	Afforded	Based	Boosted
Bought	Canceled	Caught	Caused
Completed	Contacted	Derived	Designed
Filed	Honed	Involved	Led
Listed	Made	Managed	Owned
Paid	Purchased	Related	Represented
Requested	Sold	Unsettled	

Table 3 List of 36 verb-types used in SUSANNE in subject-attached reduced relatives

Altered	Become	Bent	Burned
Charged	Clouded	Compared	Cooled
Cut	Deserted	Distilled	Dominated
Estimated	Fed	Figured	Filmed
Focused	Frozen	Internalized	Intertwined
Known	Left	Made	Opened
Posted	Proposed	Puckered	Put
Removed	Reported	Seen	Shown
Shut	Soiled	Studied	Torn

made ergatives and in particular none of the verbs used in WSJ in subject-attachment for reduced relatives can be ergativized apart from "sell". We report here above verb-types, i.e. verb wordforms taken only once. As can be easily seen none of the verbs are unergative nor unaccusatives (Table 3).

If we look at the list of verb-types used in Susanne Corpus we come up with a slightly different and much richer picture. The number of ergativizable verbs increases and also the number of verb types which is strangely enough much higher than the one present in WSJ. We also added verbs that can be intransitivized, thus contributing some additional ambiguity. In some cases, the past participle is non ambiguous, though, see "frozen, seen, shown and torn". In some other cases, the verb has different meanings with different subcategorization frames: this is case of "left".

In any case, the parser will procede by activating any possible disambiguation procedure, then it will consider the inherent semantic features associated to the prospective subject: in order to be consistent with a semantic classification as proto-agent, one of the following semantic classes will have to be present: "animate, human, institution, (natural) event, social_role, collective entity".

In the affirmative case, and after having checked for the subject position/functional assignment, the analysis will proceed at NP internal adjunct modifier position. If this is successful, the adjunct participial clause will be interpreted locally. Then the parser will continue its traversal of the grammar at i_double_bar position, searching for the finite verb.

In case no finite verb is available, there will be an ensuing failure which will recovered gracefully by a recovery call for the same main constituent expected by the grammar in that position. Two actions will take place:

1. the current input word will have to be a nonfinite verb;
2. the already parser portion of the input sentence must contain a possibly ambiguous finite verb;
3. this token word should correspond to the predicate lemma heading the modifier adjunct clause computed inside the NP which is scanned to search for the appropriate structural portion.

The first two actions are carried out on the lookahead stack, while the third action is carried out on the NP structure already parsed and fully interpreted by the parser.

5 LSLT: A Comprehensive Theory

To motivate our criticism and our approach we now introduce the foundations of our theory LSLT—Lexical Semantic Language Theory. LSLT encompasses a *psycholinguistic theory* of the way the language faculty works, a *grammatical theory* of the way in which sentences get analysed and generated—for this we will be using Lexical-Functional Grammar, a *semantic theory* of the way in which meaning gets encoded and expressed in utterances—for this we will be using Situation Semantics, and a *parsing theory* of the way in which components of the theory interact in a common architecture to produce the needed language representation to be eventually spoken aloud or interpreted by the *phonetic/acoustic language interface*.

As a start, we assume that the main task the child is faced with is creating an internal mental LEXICON, where we further assume (with Pinker and Jackendoff 2005) each word should contain two types of information: Grammatical—to feed the Grammatical component of the language faculty—and Semantic—to allow for meaning to be associated to each lexical entry. This activity is guided by two criteria: the Semantic and the Communicative Criteria.

Semantic Criterion

The goal of the language faculty is that of creating meaning relations between words and (mental) reality, that is events, entities and their attributes

Communicative Criterion

The goal of the language faculty is that of allowing communication between humans to take place

We start by addressing the *psycholinguistic theory* in which the basic goal is the creation of meaning relations between linguistic objects—words—and bits of reality—situations for short. To do that we set forth the strong claim that in order to have Analysis and Generation become two facets of the same coin, Semantics needs to be called in and Lexical information be specified in such a way to have

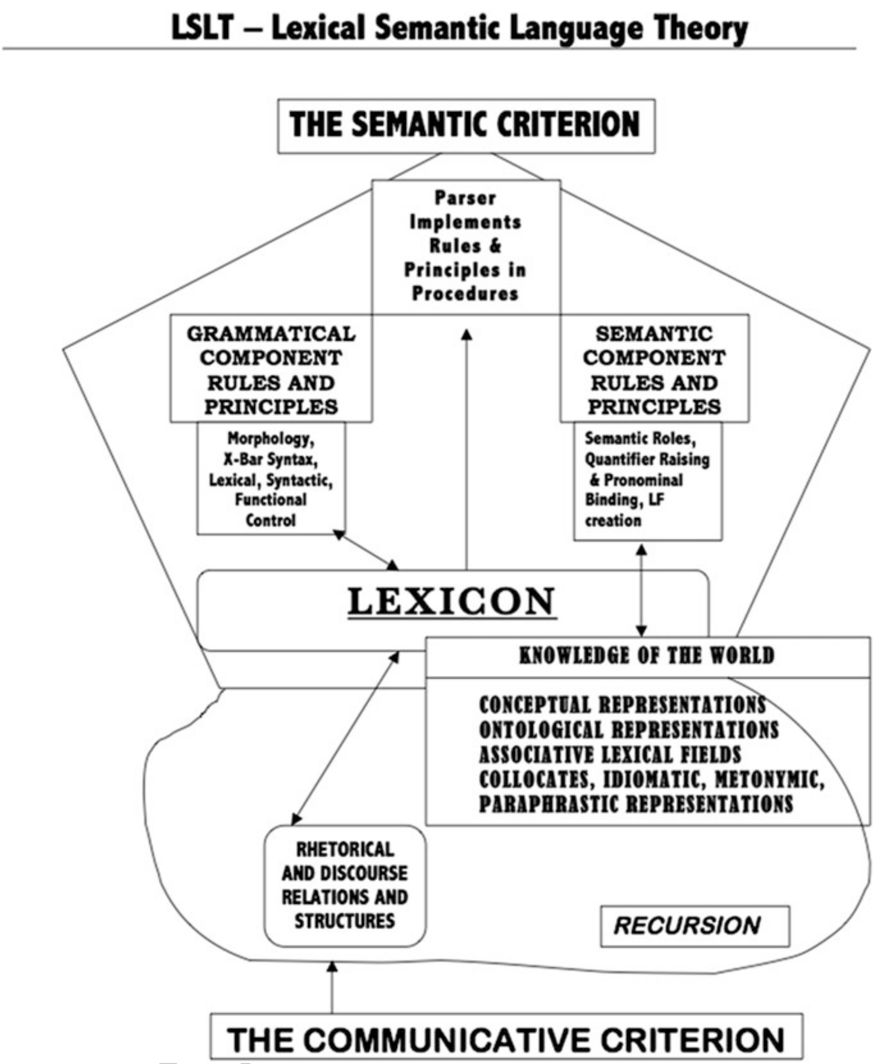
Layout: T1 Standard SC | Book ID: 323543_1_En | Book ISBN: 978-3-319-08042-0
Chapter No.: 15 | Date: 27-6-2014 Time: 7:38 pm | Page: 21/28

the Parser/Generation work properly. However, language generation implies the existence of a planning phase which may be driven by communicative needs. On the contrary, language understanding is substantially conditioned by what is usually referred to by "Shared Knowledge" between two or more interlocutors. Syntax only represents a subcomponent of the Grammatical theory and as such has no relevance in the definition of the primitives of the LSLT.

We will take the stance that the existence of a backbone of rewriting rules with reference to recursion is inherently innate (see Hauser and Chomsky 2002). However, at the same time we agree with Tomasello and others supporting a "usage-based theory of language acquisition", that the major part of linguistic competence "… involves the mastery of all kinds of routine formulas, fixed and semi-fixed expressions, idioms, and frozen collocations. Indeed one of the distinguishing characteristics of native speakers of a language is their control of these semi-fixed expressions as fluent units with somewhat unpredictable meanings"(Tomasello 2006, p. 259). The two hypothesis about language acquisition are not in contrast and coalesce in the need to have a Grammatical Maturation or Development Phase, where children start (over)generalising linguistic knowledge to new combinations. In this case, we can say that both the *Communicative Criterion* together with the *Semantic Criterion* converge on the need to express more and more complex concepts from simple holophrases to event related fully accomplished predicate-argument structures: these alone contain both functions of predicating and referring.[7]

This leads us to the second important goal of a psycholinguistic theory, that is motivating the necessity the child has to communicate with the external world. All complex constructions will appear only in a later phase of linguistic development and, in particular, they include sentential complement and relative clause constructions. As to the role of recursion in language acquisition, we believe it will only take place when the child is aware of the existence of a point of view external from his own. As said above, high level recursion in utterances is represented basically by two types of structures: sentential complements which have a *reportive* semantic content, and relative clauses which have a *supportive* semantic content.

[7] ".. from the beginning children are attempting to learn not isolated words, but rather communicatively effective speech act forms corresponding to whole adult constructions…" Tomasello (2006, p. 261). And further on, "… The language learning child is thus faced with a prodigious task: acquiring simultaneously many dozens and dozens (perhaps hundreds) of constructions based on input in which all of the many different construction types are semirandomly strewn. On the other hand, the task is made a bit easier by the fact that many of, indeed the majority of, the utterances children hear are grounded in highly repetitive item-based frames that they experience dozens, in some cases hundreds, of times every day. Indeed, many of the more complex utterances children hear have as a major constituent some well-practiced item-based frame. This means that the more linguistically creative utterances that children hear every day constitute only a small minority of their linguistic experience, and even these quite often rest on the foundation of many highly frequent and relatively simple item-based utterance frames." (ibid., 262).



Reportive contents are governed by *communication* predicates, which have the semantic content of introducing two propositions related to two separate situations in spatiotemporal terms. Supportive contents are determined by the need to bring in at the interpretation level a situation which helps better individuate the entity represented by the governing nominal predicate. These two constructions only appear at a later phase of linguistic development, as indicated also in Tomasello (2006, p. 276). And now some details on how LSLT implements its principles. AQ2

The *Grammatical Theory* (hence GT) defines the way in which lexical entries need to be organized. However, the Lexicon is informed both by the Grammatical and the Semantic Theory which alone can provide the link to the Ontology or Knowledge of the World Repository. At the analysis/comprehension level, we assume as in LFG, the existence of lexical forms where lexical knowledge is encoded, which is composed of grammatical information—categorial, morphological, syntactic, and selectional restrictions. These are then mapped onto semantic forms, where semantic roles are encoded and aspectual lexical classes are associated. In Analysis, c-structures are mapped onto f-structures and eventually turned into s-structures. Rules associating lexical representations with c-structures are part of the GT. The mapping is effortless being just a bijective process, and is done by means of FSA—finite state automata. C-structure building is done in two phases. After grammatical categories are associated to inflected wordforms, a disambiguation phase takes place on the basis of local and available lexical information. The disambiguated tagged words are organized into local X-bar based head-dependent structures, which are then further developed into a complete clause-level hierarchically based structure, though a cascaded series of FSA which make use of recursion only when there are lexical constraints—both grammatical, semantic and pragmatic—requiring it. C-structure is mapped onto f-structure by interpretation processes based on rules defined in the grammar and translated into parsing procedures. This would be a simplistic view of the parsing process, backed by constructional criteria in which syntactic/semantic constructions are readily available in the lexicon and only need to be positioned in adjacency and then glued together, as maintained by Tomasello and other constructionalists. The question is that whenever a new sentence is started there is no way to know in advance what will be the continuation at the analysis level and telling dependencies between adjacent constituents is not an easy task, as has been shown above.

It is a fact that Grammatical relations are only limited to what are usually referred to as Predicate-Argument relations, which may only encompass obligatory and optional arguments of a predicate. The Semantic Theory will add a number of important items of interpretation to the Grammatical representation, working at propositional level: negation, quantification, modality and pronominal binding. These items will appear in the semantic representation associated to each clause and are activated by means of parsing procedures specialized for those two tasks. Semantic Theory also has the task of taking care of non-grammatical objects usually defined with the two terms, Modifiers and Adjuncts. In order to properly interpret meaning relations for these two optional component of sentential linguistic content, the Semantic theory may access Knowledge of the World as represented by a number of specialized lexical resources, like Ontology, for inferential relations; Associative Lexical Fields for Semantic Similarity relations; Collocates for most frequent modifier and adjunct relations; Idiomatic and Metonymic relations as well as Paraphrases for best stylistic purposes.

In Generation, a plan is created and predicates are inserted in predicate-argument structures (hence PAS) with attributes—i.e. modifiers and adjuncts. Syntax plays only a secondary role in that PAS are hooked to stylistic, rhetorical rules which are in turn genre and domain related. They are also highly idiosyncratic, strongly depending on each individual social background. Surface forms will be produced according to rhetorical and discourse rules, by instantiating features activated by semantic information.

5.1 LSLT and Contextual Reasoning

There are two main additional tenets of the theory: one is that it is possible to reduce access to domain world knowledge by means of contextual reasoning, i.e. reasoning triggered independently by contextual or linguistic features of the text under analysis. In other words, we adopt what could be termed the Shallow Processing Hypothesis: access to world knowledge is reduced and substituted whenever links are missing through inferences on the basis of specifically encoded lexical and grammatical knowledge, and are worked out in a fully general manner. In exploring this possibility we make one fundamental assumption and it is that the psychological processes needed for language analysis and understanding are controlled by a processing device which is completely separated from that of language generation with which it shares a common lexicon though.

In our approach there is no language model for probabilistic processing even though we use statistical processing for strongly sequential tasks like tag disambiguation. Our algorithms are based on symbolic rules and we also use FSA to help tag disambiguation and parsing. The reason for this is twofold: an objective one, machine learning for statistical language models need linguistic resources which in turn are both very time-consuming to produce and highly error-prone activities. On a more general level, one needs consider that highly sophisticated linguistic resources are always language and genre dependent, besides the need to comply with requirements of statistical representativeness. No such limitations can be deemed for symbolic algorithms which on the contrary are more general and easily portable from one language to another. Differences in genre can also be easily accounted for by scaling the rules adequately.

It is sensible to assume that when understanding a text a human reader or listener does make use of his encyclopaedia parsimoniously. Contextual reasoning is the only way in which a system for Natural Language Understanding should tap external knowledge of the domain. In other words, a system should be allowed to perform an inference on the basis of domain world knowledge when needed and only then. In this way, the system could simulate the actual human behaviour in that access to extralinguistic knowledge is triggered by contextual factors independently present in the text and detected by the system itself. This would be required only for implicit linguistic relations as can happen with bridging descriptions, to cope with anaphora resolution phenomena, for instance. In other words, we believe that there are principled ways by which linguistic processes interact with knowledge representation or the ontology—or to put it more simply, how syntax interacts with pragmatics.

In fact, no solution for such an interaction has yet been found, nor even tackled by current deep systems (See papers in Bos and Delmonte 2008). In these systems, linguistic components are kept separate from knowledge representation, and work which could otherwise be done directly by the linguistic analysis is duplicated by inferential mechanism. The output of linguistic analysis is usually mapped onto a logical representation which is in turn fed onto the knowledge representation of the domain in order to understand and validate a given utterance or query. Thus the domain world model or ontology must be priorily built, usually in view of a given task the system is set out to perform. This modeling is domain and task limited and generality can only be achieved from coherent lexical representations. In some of these systems, the main issue is how to make the two realms interact as soon as possible in order to take advantage of the inferential mechanism to reduce ambiguities present in the text or to allow for reasoning on linguistic data, which otherwise couldn't be understandable. We assume that an integration between linguistic information and knowledge of the world can be carried out at all levels of linguistic description and that contextual reasoning can be thus performed on the fly rather than sequentially. This implies that access to knowledge must be filtered out by the analysis of the linguistic content of surface linguistic forms and their abstract representations of the utterances making up the text. Thus the two important elements characterizing human language faculty, that is "ambiguity" and "recursion", find a fully justified role in the theory and are also entirely justified by it.

6 An Evaluation and a Conclusion

The system and the parser have gone through an extended number of evaluations: starting from the latest one, where they been used to produce the output for the Events Workshop (2013); then, Sentiment, Factuality and Subjectivity analysis, (Delmonte and Pallotta 2011); Relational Models of Semantics, (Tonelli and Delmonte 2011); Automatic Identification of Null Instantiations, at SEMEVAL (Delmonte and Tonelli 2010); Semantic Processing for Text Entailment, (Delmonte et al. 2010, 2005, 2006, 2007); Semantic and Pragmatic Computing in STEP (Delmonte 2008); Causality Relations in Unrestricted Text, (Delmonte et al. 2007); Evaluation of Anaphora Resolution, (Delmonte et al. 2006); Comparing Dependency and Constituency Parsing, (Delmonte 2005); Evaluating Grammatical Relations, (Delmonte 2004).

All these evaluations have shown the high resiliency of the system for the different applications with little adjustments. Results are comparable if not better, thus that manually organized linguistic grammars and parsers are better suited for semantically related tasks. It is a fact that Machine Learning does not easily adapts to the presence of null elements in the training set and this represents a fatal drawback for any further improvement along the lines indicated in the paper.

References

Abney, S. (1996). Part-of-speech tagging and partial parsing. In K. Church, S. Young, & G. Bloothooft (Eds.), *Corpus-based methods in language and speech* (pp. 118–136). Dordrecht: Kluwer Academic Publishers.

Berwick, R. C., Abney, S., & Tenny, C. (1991). *Principle-based parsing: Computation and psycholinguistics*. New York: Kluwer Academic Publishers.

Bresnan, J. (1982). *The mental representation of grammatical relations*. Cambridge Mass: MIT Press.

Bresnan, J. (2001). *Lexical-functional syntax*. Oxford: Blackwells.

Brants, T., & Samuelsson, C. (1995). Tagging the teleman corpus. *In Proceedings 10th Nordic Conference of Computational Linguistics* (pp. 1–12). Helsinki.

Briscoe, T. (1995). Robust parsing. In G. B. Varile & A. Zampolli (Eds.), *Survey of the state of the art in human language technology, language analysis and understanding* (pp. 113–114). Cambridge, England; New York: Cambridge University.

Carnie, Andrew, & Barss, Andrew. (2006). Phases and nominal interpretation. *Research in Language, 4*, 127–132.

Carroll, J. A. (2000). Statistical parsing. In R. Dale, H. Moisl, & H. Somers (Eds.), *Handbook of natural language processing* (pp. 525–543). New York: Marcel Dekker.

Clifton, C., & Ferreira, F. (1989). Ambiguity in context. In G. Altman (Ed.), *Language and cognitive processes* (pp. 77–104). op.cit.

Delmonte, R., & Bianchi, D. (1991). Binding pronominals with an LFG parser. *Proceeding of the Second International Workshop on Parsing Technologies, Cancun (Messico)* (pp. 59–72). ACL 1991.

Delmonte, R. (2013). Coping with implicit arguments and events coreference in Eduard Hovy, Teruko Mitamura, Martha Palmer. *Proceedings of the Conference The 1st Workshop on EVENTS, HLT-NAACL 2013, Atlanta, Georgia* (pp. 1–10) .

Delmonte, R. (2013). Predicate argument structures for information extraction from dependency representations: Null elements are missing. In C. Lai, A. Giuliani & G. Semeraro (Eds.), *Studies in Computational Intelligence*, Distributed Systems and Applications of Information Filtering and Retrieval. DART 2012 (pp. 1–25). New York: Springer.

Delmonte, R., & Vincenzo, P. (2011). Opinion mining and sentiment analysis need text understanding. In V. Pallotta, A. Soro & E. Vargiu (Eds.), *Advances in distributed agent-based retrieval tools: Studies in computational intelligence* (Vol. 361, pp. 81–96). New York: Springer.

Delmonte, R., & Tonelli, S. (2010). VENSES ++-UNIVE: Adapting a deep semantic processing system to the identification of null instantiations. *In Proceedings of the 5th International Workshop on Semantic Evaluation* (pp. 296–299). ACL 2010.

Delmonte, R., Tonelli, S., & Tripodi, R. (2009). Semantic processing for text entailment with VENSES. *In Proceedings of Text Analysis Conference (TAC) 2009 Workshop—Notebook Papers and Results* (pp. 453–460). Gaithersburg, MA: NIST.

Delmonte, R. (2009). *Computational linguistic text processing—lexicon, grammar, parsing and anaphora resolution*. New York: Nova Science Publishers.

Delmonte, R. (2008). Semantic and pragmatic computing with GETARUNS. In J. Bos & R. Delmonte (Eds.), *Semantics in text processing (STEP), research in computational semantics* (Vol. 1, pp. 287–298). London: College Publications.

Delmonte, R. (2007). *Computational linguistic text processing—logical form, semantic interpretation, discourse relations and question answering*. New York: Nova Science Publishers.

Delmonte, R., Nicolae, G., Harabagiu, S. (2007). A linguistically-based approach to detect causality relations in unrestricted text. *In Proceedings of MICAI-2007* (pp. 173–185). IEEE Publications.

Delmonte, R., Bristot, A., Piccolino Boniforti, M. A., & Tonelli, S. (2006). Another evaluation of anaphora resolution algorithms and a comparison with GETARUNS' knowledge rich approach. *ROMAND 2006, 11th EACL* (pp. 3–10). Trento: Association for Computational Linguistics.

Delmonte, R., Bristot, A., Piccolino Boniforti, M. A., & Tonelli, S. (2006). Coping with semantic uncertainty with VENSES. In B. Magnini & I. Dagan (Eds.), *Proceedings of the Challenges Workshop—The 2nd PASCAL Recognizing Textual Entailment Challenge* (pp. 86–91).

Delmonte, R., Tonelli, S., Piccolino Boniforti, M. A., Bristot, A., Pianta, E. (2005). VENSES—a linguistically-based system for semantic evaluation. In J. Quiñonero-Candela et al. (Ed.), *Machine learning challenges. Evaluating predictive uncertainty, visual object classification, and recognising textual entailment: First pascal machine learning challenges workshop* (pp. 344–371). Southampton, UK, Revised Selected Papers.

Delmonte, R. (2005). Deep & shallow linguistically based parsing. In A. M. Di Sciullo (Ed.), *UG and external systems* (pp. 335–374). Amsterdam/Philadelphia: John Benjamins.

Delmonte, R. (2004). Evaluating GETARUNS parser with GREVAL test suite. *Proceedings of the ROMAND—20th International Conference on Computational Linguistics—COLING* (pp. 32–41). Geneva.

Delmonte, R. (2002a). Relative clause attachment and anaphora: A case for short binding. *Proceedings of TAG + 6* (pp. 84–89). Venice.

Delmonte, R. (2000b). Generating and parsing clitics with GETARUN. *Proceedings of CLIN'99* (pp. 13–27). Utrech.

Delmonte, R. (2000c) (to appear). Parsing preferences and linguistic strategies. *Proceedings of Workshop Communicating Agents* (p. 15). Bonn: IKP.

Delmonte, R. (2000). Parsing with GETARUN. *Proceedings of TALN2000, 7° confèrence annuel sur le TALN* (pp. 133–146). Lausanne.

Filip, H. (1998). Reduced relatives: Lexical constraint-based analysis. *Proceedings of the Twenty-Fourth Meeting of the Berkeley Linguistic Society* (pp. 1–15).

Fodor, J. (2002). Psycholinguistics cannot escape prosody, Invited Talk, SpeechProsody2002, Aix-en-Provence.

Frazier, L. (1987). Sentence processing. In M. Coltheart (Ed.), *Attention and performance XII*. Hillsdale, N.J: Lawrence Elbaum.

Johan, Bos, & Delmonte, Rodolfo (Eds.). (2008). *Semantics in text processing (STEP), Research in computational semantics* (Vol. 1). London: College Publications.

Hauser, M. D., Chomsky, N., & Tecumesh Fitch, W. (2002). The faculty of language: What it is, who has it, and how did it evolve? *Science, 298*, 1569–1579.

Hindle, D., & Roth, M. (1993). Structural ambiguity and lexical relations. *Computational Linguistics, 19*(1), 103–120.

Kay, M. (1980). Algorithm schemata and data structures in syntactic processing, CSL-80-12, Xerox Corporation, Palo Alto Research Center.

Karlsson, F., & Karttunen, L. (1995). Shallow parsing. In G. B. Varile & A. Zampolli (Eds.), *Survey of the state of the art in human language technology, Language analysis and understanding* (pp. 113–114). Cambridge, England; New York: Cambridge University Press.

Karlsson, F. (2010). Syntactic Recursion and Iteration, In: H. van der Hulst (Ed.), *Recursion and human language* (pp. 43–67). Berlin: De Gruyter Mouton.

Kinsella, A. (2010). Was recursion the key step in the evolution of the human language faculty? In: H. van der Hulst (Ed.), *Recursion and human language* (pp. 179–192). Berlin: De Gruyter Mouton.

Kuhn J. (2013). On the "Spirit of LFG" in current computational linguistics. *In Proceedings of LFG13* (pp. 356–376). CSLI Publications online.

Laury R., & Ono, T. (2010). Recursion in conversation: What speakers of finnish and Japanese know how to do. In H. van der Hulst (Ed.), *Recursion and human language* (pp. 69–92). Berlin: De Gruyter Mouton.

Pereira, F. (1981). Extraposition grammars. *American Journal of Computational Linguistics, 7*(4), 243–256.

Pereira, F. (1995). Sentence modeling and parsing. In G. B. Varile & A. Zampolli (Eds.), *Survey of the state of the art in human language technology, language analysis and understanding* (pp. 113–114). Cambridge, England; New York: Cambridge University Press.

Pereira, F., & Warren, D. (1980). Definite clause grammar for language analysis—a survey of the formalism and a comparison with ATN. *Artificial Intelligence, 13*, 231–278.

Pinker, S., & Jackendoff, R. (2005). The faculty of language: What's special about it?*. *Cognition, 95*, 201–236.

Pritchett, B. L. (1992). *grammatical competence and parsing performance*. Chicago: The University of Chicago Press.

Schubert, L. K. (1984). On parsing preferences. *Proceedings of COLING* (pp. 247–250). Stanford.

Stevenson, S., & Merlo, P. (1997). Lexical structure and parsing complexity. In M. C. MacDonald (Ed.), *Lexical Representations and Sentence Processing* (pp. 349–399). UK: Psychology Press.

Tapanainen, P., & Voutilainen, A. (1994). Tagging accurately—don't guess if you know. *Proceedings of ANLP '94* (pp. 47–52). Stuttgart, Germany.

Tomasello, M. (2006). Acquiring linguistic constructions. In R. Siegler & D. Kuhn (Eds.), *Handbook of child psychology: Cognitive development* (pp. 255–298). New York: Wiley.

Tonelli, R., & Delmonte, R. (2011). Desperately seeking Implicit arguments in text. In S. N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, S. Szpakowicz (Eds.), *Proceedings of the ACL 2011 workshop on relational models of semantics (RELMS'2011)* (pp. 54–62). Portland, USA.

Verhagen, A. (2010). What do you think is the proper place of recursion? Conceptual and empirical issues. In H. van der Hulst (Ed.), *Recursion and human language* (pp. 93–112). Berlin: De Gruyter Mouton.

Author Query Form

Book ID : **323543_1_En**
Chapter No.: **15**

Springer
the language of science

Please ensure you fill out your response to the queries raised below and return this form along with your corrections

Dear Author
During the process of typesetting your chapter, the following queries have arisen. Please check your typeset proof carefully against the queries listed below and mark the necessary changes either directly on the proof/online grid or in the 'Author's response' area provided below

Query Refs.	Details Required	Author's Response
AQ1	References Hulst (2010), Svenonius (2001, 2004), Legate (2002, 2003), Epstein (2004), Townsend and Bever (2001), Chomsky (1998/2000), Delmonte (1999), Delmonte et al. (2010), Hobbs et al. (1992), Altmann (1989), Hauser and Chomsky (2002), Pereira (1983) are cited in text but not provided in the reference list. Please provide references in the list or delete these citations.	
AQ2	The citation Delmonte (2013) has been changed to match the Delmonte (2013a, b) in the reference list. Please check here and in subsequent occurrences, and correct if necessary.	
AQ3	Please check and confirm the inserted citation of Tables 1, 2, 3 are correct. If not, please suggest an alternate citation. Please note that tables should be cited in sequential order in the text.	
AQ4	References Abney (1996), Briscoe (1995), Carroll (2000), Delmonte and Bianchi (1991), Delmonte et al. (2009), Delmonte (2000a, b, c), Frazier 1987b, Filip (1998), Fodor (2002), Hindle and Roth (1993), Kay (1980), Karlsson and Karttunen (1995), Karlsson (2010), Kinsella (2010), Laury and Ono (2010), Pereira (1995), Pereira and Warren (1980), Verhagen (2010) are given in list but not cited in text. Please cite in text or delete from list.	

MARKED PROOF

Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

Instruction to printer	*Textual mark*	*Marginal mark*
Leave unchanged	• • • under matter to remain	
Insert in text the matter indicated in the margin		New matter followed by or
Delete	/ through single character, rule or underline or through all characters to be deleted	or
Substitute character or substitute part of one or more word(s)	/ through letter or through characters	new character / or new characters /
Change to italics	— under matter to be changed	
Change to capitals	under matter to be changed	
Change to small capitals	under matter to be changed	
Change to bold type	under matter to be changed	
Change to bold italic	under matter to be changed	
Change to lower case	Encircle matter to be changed	
Change italic to upright type	(As above)	
Change bold to non-bold type	(As above)	
Insert 'superior' character	/ through character or where required	or under character e.g. or
Insert 'inferior' character	(As above)	over character e.g.
Insert full stop	(As above)	⊙
Insert comma	(As above)	,
Insert single quotation marks	(As above)	or and/or or
Insert double quotation marks	(As above)	or and/or or
Insert hyphen	(As above)	
Start new paragraph		
No new paragraph		
Transpose		
Close up	linking characters	
Insert or substitute space between characters or words	/ through character or where required	
Reduce space between characters or words	between characters or words affected	