Natural-language syntax as procedures for interpretation: the dynamics of ellipsis construal

R.Kempson, E.Gregoromichelaki, W.Meyer-Viol
Philosophy Department
King's College London

M Purver, G.White
School of Electronic Engineering and Computer Science
Queen Mary University of London

R. Cann
Linguistics and English Language
University of Edinburgh

Abstract

In this paper we set out the preliminaries needed for a formal theory of *context*, relative to a linguistic framework in which natural-language syntax is defined as procedures for context-dependent interpretation. Dynamic Syntax provides a formalism where both representations of content and context are defined dynamically and structurally, with time-linear monotonic growth across sequences of partial trees as the core structure-inducing notion. The primary data involve *elliptical fragments*, as these provide less familiar evidence of the requisite concept of context than anaphora, but equally central. As part of our sketch of the framework, we show how apparent anomalies for a time-linear basis for interpretation can be straightforwardly characterised once we adopt a new perspective on *syntax* as the dynamics of transitions between parse-states. We then take this as the basis for providing an integrated account of ellipsis construal. And, as a bonus, we will show how this intrinsically dynamic perspective extends in a seamless way to dialogue exchanges with free shifting of role between speaking and hearing (*split-utterances*). We shall argue that what is required to explain such dialogue phenomena is for contexts, like representations of content, to include not merely partial structures but also the sequence of actions that led to such structures.

1 Preliminaries

Despite extensive research on the context-dependence of natural-language (NL) understanding over the last thirty years, with formal modelling of a wide range of individual phenomena, there has been little attempt to

bring everything together in order to seek an overall concept of context-dependence. In particular, ellipsis has been treated wholly differently from *anaphora*, despite the fact that, like anaphora, ellipsis is a phenomenon which, by definition, exhibits radical context-dependency. An elliptical construction is one "that lacks an element that is, nevertheless, recoverable or inferable from the context" (Wikipedia: *ellipsis*). This characterisation corresponds to the robust folk intuition that, in language use, expressions can be omitted because context fully determines the way the fragment utterance is to be understood. Seen from this point of view, it is reasonable to expect that the phenomenon of ellipsis will provide a basis from which the notion of context needed for language interpretation can be explored; and indeed in this paper we shall use investigation of ellipsis exactly to this end. As (Purver et al., 2006; Cann et al., 2007) argue, if suitably dynamic concepts of structure and context are defined, a unitary account of ellipsis, in all contexts, can be provided making sense of all the distinct aspects of context-dependence exhibited in language use.

2 Ellipsis and the Syntax-Semantics Interface

Current accounts of ellipsis do not in general purport to provide a point of departure for the study of context (though cf. Ginzburg and Cooper, 2004; Fernández, 2006). The consensus is that ellipsis is not a homogeneous phenomenon. Rather, it splits into syntactic, semantic and pragmatic types, with only the last type depending on context for construal.

The general background for both syntactic and semantic accounts is the methodology of conventional grammars which dictates the sentence as the unit of characterisation: the only forms of ellipsis addressed have been those where the ellipsis site can in some sense be analysed sententially – either as a second conjunct of a compound conjunctive form or as an answer to a question with both of these being analysed in sentential terms:

(1) A: Have you seen Mary?
B: Mary? (a) No, I haven't. (b) I have, Bill. (c) Tom too. (d)
B: Have I seen Mary? No, I haven't seen Mary but I have seen Bill; and I have seen Tom too.

Thus (1a) can be understood as an echo of the original question, (1b) as the negative answer *I have not seen Mary*, and so on. Indeed (1) illustrates a number of different ellipsis types. Each of them has been argued to be a separate syntactic phenomenon on the evidence of apparently different structural constraints governing their reconstruction as full sentential forms. However, the ever-accumulating set of phenomena, all labelled ellipsis, do not seem reducible to some general abstract account. Indeed, the general phenomenon, ellipsis, remains puzzling, with apparent conflicting evidence for both semantic and syntactic forms of explanation (see Ginzburg and Cooper, 2004; Merchant, 2007). On the one hand, a semantic explanation

is available for cases where, for a single antecedent form and assigned interpretation, ambiguity nonetheless arises:

(2) John checked over his mistakes, and so did Bill/Bill too.
'Bill checked Bill's mistakes' ("sloppy")
'Bill checked John's mistakes' ("strict")

This is argued to reflect a process of abstraction over some content provided by the antecedent ('John checked over John's mistakes') creating distinct abstracts to apply to the content of the fragment in the elliptical conjunct (a process involving *higher-order unification*): (a) abstraction over solely the subject of the antecedent, or (b) abstracting over the subject and all other references to the individual it denotes. Two resulting predicates can then apply to the elliptical fragment which yields so-called strict/sloppy ambiguities (Dalrymple et al., 1991). Since these abstraction operations affect content (not syntactic structure) this provides the semantic basis for explanation. A second argument for semantic accounts of ellipsis is provided with the almost invariant parallelism of the mode of interpretation between an elliptical second conjunct and its antecedent clause, as in e.g. parallelism of quantifier dependencies (scope):

(3) Every professor got to meet with a visiting government official, and so did every administrator.

Such parallelism of interpretation is said to arise in virtue of variation in the abstraction steps available for the antecedent clause to the ellipsis site. If abstraction applies *prior* to quantifying in the quantified expression within the predicate, then, with only a variable in place of this quantified form in that first conjunct, the result will be an interpretation in which the quantifier simultaneously binds a variable in both conjuncts, hence incorporating the entire conjunction within its scope. If, however, the abstraction operation to create the requisite predicate takes place *after* all quantified expressions in the first conjunct are quantified in, then whatever quantifying terms are contained in the resulting abstract will be interpreted as taking narrow scope with respect to the subject expression with which the created predicate is to be combined. So far so good. But there are three possible interpretations for a sentence such as (3), not just two. The third is where the indefinite is construed as taking wider scope than the subject NP of the conjunct in which it is contained but nevertheless not wide scope with respect to the whole conjunction. In (3) this involves there being two visiting government officials, one visiting the professors, and one visiting the administrators. This third interpretation cannot be captured by the Dalrymple et al. account (an observation due to Mark Steedman), so the semantic account is at best incomplete.

In any case, there is competing evidence that sensitivity to structure is essential to the way in which the elliptical fragment is reconstructed. Ellipsis is highly sensitive to syntactic/morphological requirements set up by the sequence of expressions preceding the ellipsis site:

(4) A: Who did every husband visit? B: His wife.

Moreover, there are cases of ellipsis, so-called *antecedent-contained ellipsis*, that display sensitivity to the very constraints taken to be diagnostic of syntactic phenomena. These are the so-called 'island' constraints which are taken as evidence that at least those forms of ellipsis must be analysed in syntactic terms (Fiengo and May, 1994; Merchant, 2004):

(5) John interviewed every student who Bill already had.

(6) *John interviewed every student who Bill ignored the teacher who already had.

In (6), an instance of *antecedent-contained ellipsis*, the ellipsis site cannot be associated with the relative pronoun (*who*) even though this is possible in (5). This is because a relative-clause boundary intervenes between the relative pronoun and the elliptical element. This type of violation is directly redolent of the island restrictions constraining long-distance dependencies as in e.g. *wh*-questions (*wh* binding also not being possible across a relative-clause boundary). Because such restrictions are taken not to be expressible in semantic terms – the lambda calculus underpinning semantic combinatorics would impose no such structure-particular restriction – they have been taken as evidence for a level of syntactic structure independent of semantics, and a diagnostic of what constitutes a syntactic phenomenon. Hence, so the argument goes, at least some types of ellipsis require syntactic explanation, involving full projection of clausal structure at the ellipsis site with subsequent deletion of phonological material. However, even though structural restrictions can be captured by syntactic reconstructions of ellipsis, this provides no explanation of parallelism effects, which require the definition of independent constraints (see e.g. Fox, 1999).

Over and above the division of ellipsis into semantic and syntactic types, there is the very widespread use of fragments in dialogue. Even though grammatical formalisms have neglected these as "performance data", more recently, as dialogue modelling has developed, this omission is being repaired. Extending the Dalrymple et al. pattern, Ginzburg and Sag (2000), Ginzburg and Cooper (2004) defined multiple types of abstraction mechanisms to reflect distinct identified types of ellipsis, ranging over semantic, syntactic and morphological specifications associated with the antecedent of the elliptical form. But the move made is not straightforward, for, retaining the sentence-based methodology, these fragments are analysed as full sentences, with type-lifting of the fragment not merely in the semantics but also in the syntax. This assumption is problematic if the fragment occurs, as in (21), at a point in the dialogue exchange when no antecedent of appropriate type is available in context to yield the appropriate abstract:

(7)

A: And er they X-rayed me, and took a urine sample, took a blood sample.
A: Er, the doctor
B: Chorlton?
A: Chorlton, mhm, he examined me, erm, he, he said now they were on about a slide [unclear] on my heart. [BNC: KPY 1005-1008]

An additional problem for any such assumption, is that anaphoric and quantificational dependencies can be seamlessly continued across from one speaker to another, as in (4), involving the binding of a pronoun by a quantifying expression. Any syntactic dependency whatsoever can be split between speaker and hearer, with the hearer-turned-speaker continuing an initiated utterance relative to the structural context provided by the first part (see Purver et al. (2009)):

(8) A: Have you read
B: any of your books? Certainly not.

(9) A: Are you OK? Did you burn
B: myself? Fortunately not.

Beyond the syntax/semantics controversy, there are yet further cases where there is no linguistic basis for assigning interpretation to the fragment. Stainton (2006) argues that such cases do not allow any analysis as sentential reconstructions but have to be seen as a speech act that is performed without recourse to a sentential structure:

(10) A (coming out of lift) McWhirter's?
B Second left.

In these, it is non-linguistic aspects of context that determine the way the fragment is understood. What this illustrates is that, from a pre-theoretical point of view, fragments in general can occur whenever the context provides elements relative to which the fragment can be processed in an appropriate way. The context may provide linguistic structure on the basis of which the fragment may yield a propositional content as in (4), in the fragment interruptions of the questions in (8)-(9) and their subsequent fragment replies. But, as in the case of anaphora (e.g. *bridging* phenomena, see Clark (1977)), sometimes both contents and the appropriate context for processing have to be constructible on the fly as in (10). Even further, the fragment expression may have to be interpreted as an extension of a nonpropositional structure given in the context, as in (7). The disparate ways in which elliptical fragments can be understood have been taken as evidence of the so-called "fractal heterogeneity" of ellipsis (Ginzburg and Cooper, 2004) in that its resolution apparently involves cross-cutting constraints across whatever information the grammar manipulates, with morphological, syntactic, semantic, even phonological information yielding multiple bases for ellipsis. In addition, the grammar interacts with a range of dialogue interpretation principles which also contribute to disambiguating the function of each type

of fragment. In the complexity that results from this reconstruction of context as multiple sets of constraints on interpretation across distinct modes of representation, the robustness of the folk intuition is getting lost: how can the context, from which speakers and hearers freely draw, require such complex cross-module constraints and heterogeneity in need of disambiguation online? The challenge of providing a uniform account of ellipsis processing thus remains.

An alternative is to take ellipsis as a phenomenon from which we can glean evidence of the types of information that context records, and on that basis, to explore processes of dynamic update manipulating this information in a unified way. And for this, we turn to Dynamic Syntax, where the dynamics of how information accrues along the time-line of processing is integral to the formalism's structural underpinnings.

3 Dynamic Syntax

Dynamic Syntax (DS) is a model of how interpretation is built up relative to context, reflecting how hearers (and speakers) construct interpretations for strings of words incrementally using linguistic information and context as it becomes available. Crucially, the output of any processing task is a representation of the content of the string uttered in a particular context (not a representation of some hierarchical structure defined over the string, i.e., not a sentence *type* assumed to hold over any linguistic context). NL syntax, on this view, is conceived of as the *process* by which such semantic representations are built up, using instructions associated with words and contextual information to drive the incremental development of the output representation. Since pragmatics may interact at any point with the online syntactic process, output semantic representations may differ for the same string uttered in different contexts.

Formally, DS is a lexicalized grammar using labelled sequences and trees as basic data structures. The labelled sequences are time-linear sequences of words with accompanying phonological, morphological and word-boundary specification; the trees formalize the semantic, functor-argument structure induced from utterances of such sequences in the form of (unordered) trees labeled by terms of a typed lambda calculus and other process-control labels. This emphasis on strings and trees DS shares with other tree-based linguistic theories like Transformational Generative Grammar (in all its guises), Tree Adjoining Grammar, Head-driven Phrase Structure Grammar, and Categorial Grammar. It differs from virtually all these in the way it relates the two data structures. Other theories identify linearity with the yield of ordered trees; in DS, the string determines a sequential 'derivation tree'; the words are interpreted as instruction packages to construct parts of a lambda term in a labelled tree representation. The packages are executed word-by-word from-left-to-right. The terminal nodes of a lambda term produced by a grammatical sentence in some instances stand in one-to-one correspon-

dence with the words of the sentence, but an individual word may also induce sub-structure containing more than one labelled node; and there is no direct relation between the yield of the eventual tree and the sequential order of the string. Terms of only a small, fixed, set of semantic types are used, so no new types (functions) can be constructed (unlike categorial frameworks). Structural underdetermination and update replace concepts of 'movement' (and its analogues in non-transformational frameworks) and function-composition.

The tree is incrementally constructed by tree substitution and addition or extension of labels, but there are two ways to escape the strict linear evaluation order: by *structural underdetermination* and by *underdetermination of labels*; both may lead to delay in choices to be made. Structural underdetermination involves the addition of a sub-term to the tree, an *unfixed node*, whose location in that tree is characterised as merely dominated by a previously constructed term without, as yet, a fully specified hierarchical position. So, for example, in the characterisation of *Mary, John likes* instructions license the introduction of a typed sub-term into the tree as specified by the word *Mary* that cannot yet be given a fixed location in the tree skeleton of the term under construction. Underspecification in the labels dimension is implemented by means of employing *meta-variables*, indicated as **U**, **V**, etc., as temporary labels allowing identification from some larger context given their associated type specification (and any other constraints that may be identified, such as gender, person, and so on).

The parsing process is lexically driven and directed by *requirements*. The main requirement is '?t', the *type requirement* to produce a formula of proposition type t from the word string; but imposition of such requirements for other types induces the creation of new labeled tree structure, providing the basis for expressing how one word may 'subcategorize' for others. As there are only a finite number of types, type requirements can be viewed as *non-terminal symbols* to be rewritten to a terminal symbol (supplied by a word). Tree-growth *actions* are defined as general or lexically triggered options, dividing into two broad types: those for developing a tree-structure for which a number of strategies may be available; and those annotating non-terminal nodes through algorithmic application of β-reduction and its associated type-deduction. These in combination yield a resulting structure, with all nodes properly annotated and no requirements outstanding. Type requirements are not the only form of requirement. Indeed all forms of underspecification are associated with requirements for update, for example, meta-variables co-occur with a requirement for instantiation, represented as $?\exists \mathbf{x}.Fo(\mathbf{x})$. Well-formedness for a string is defined in terms of the possibility of constructing a proper tree rooted in type t with no outstanding requirements left on any node in the tree.

3.1 Tree Growth and LOFT: The Logic of Finite Trees

To round out this summative characterisation, we now sketch the general process of parsing as the induction of a sequence of partial trees, whose input is a one-node tree annotated with only the requirement $?t$ and a pointer $\Diamond$ indicating the node under development, and whose output is a binary branching tree whose nodes reflect the content of some propositional formula. The left-to-right parse of the string *Bob upset Mary* gives rise to the sequence of partial trees shown in (3.1), with the final tree completing the initial requirement.

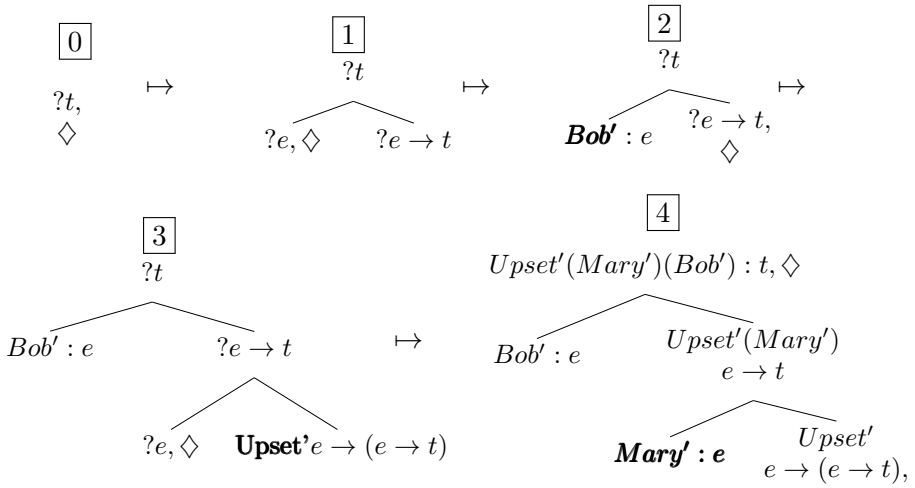


Figure 1: Monotonic tree growth in DS

The parsing task, using both lexical input and information from context, is thus to progressively enrich the input tree to yield a complete output using general tree-growth actions, lexical tree-growth actions, and, when triggered by some lexical item, pragmatic tree-growth actions. As all types of action are defined in the same terms, i.e., as actions that map one partial tree to another, different types of action can interleave at any point. *Decorations* on nodes include content-representing formulae, type specifications, a treenode indicator (with one node distinguished as the rootnode: $Tn(0)$), and requirements for such decorations as imposed by the unfolding process. The primitive types are types e and t as in formal semantics but construed syntactically.[1] Each node must be eventually decorated with a pairing of formula and type specifications written as $\alpha : \phi$ (α the formula labelling ϕ the type). Annotations on non-terminal nodes are induced algorithmically in the final evaluation of the trees, in terms inspired by the Curry-Howard isomorphism and labelled type-deduction. In all cases, the output is a fully annotated (*decorated*) tree whose topnode is a formula value representing a proposition derived by the string of words processed relative to a particular context of utterance.[2]

[1] There are other types, but the list is highly restricted.

[2] We simplify the exposition here: the full presentation of Dynamic Syntax includes

At the heart of the formal characterisation is the (modal) logic of finite trees (LOFT: Blackburn and Meyer-Viol, 1994) which permits addressing any node in the tree from the perspective of any other node using the immediate-dominance modalities $\langle\downarrow\rangle$ and $\langle\uparrow\rangle$ and variations over these. Such operators can be used to indicate nodes that exist already in the tree (e.g. $\langle\downarrow\rangle\alpha$ indicates that there is a daughter of the current node decorated by label α), with variants distinguishing $\downarrow_0/\downarrow_1$ as argument/functor daughter respectively, and Kleene * operations over these to define general dominance relations. Such decorations are used in conjunction with the requirement operator ?, to indicate nodes that, at some stage, must occur on the tree but may not currently do so (e.g. $?\langle\downarrow\rangle\alpha$ indicates that there must, eventually in the derivation, exist a daughter of the current node decorated by label α). Tree-growth is thus defined relative to the imposition and subsequent satisfaction of requirements: $?X$ for any annotation on a node constitutes a constraint on how the subsequent parsing steps must progress, i.e. X must be derived. Hence requirements such as $?t$, $?e$, $?e \rightarrow t$ impose constraints on tree development that formulae of the relevant types must be provided by the end of the parsing process. Constraints on growth may also be modal, e.g. while a decoration $?e$ requires a term to be constructed at the current node, $?\langle\downarrow\rangle e$ requires a daughter node to be so decorated. Although type-requirements are the primary drivers of the syntactic process, in principle, any label may be associated with a requirement. For example, $?\exists\mathbf{x}.Fo(\mathbf{x})$ requires some contentful formula value to decorate a node and is associated with the parsing of pronouns and other anaphoric expressions such as auxiliary verbs in elliptical structures.

An essential ancillary notion to that of tree growth is the concept of *procedure* or *action* for mapping one partial tree to another. These are defined in a language involving such commands as $\texttt{make}(\langle\downarrow\rangle)$, $\texttt{go}(\langle\downarrow\rangle)$, $\texttt{put}(\alpha)$, $\texttt{make}(\langle\downarrow_*\rangle)$, $\langle\texttt{IF}\ldots,\texttt{THEN}\ldots,\texttt{ELSE}\ldots\rangle$ etc. Sets of such actions incorporated in individual packages can be either general *computational rules* or *lexical actions* associated with words contributing content-formulae and other aspects of structure. For example, verbs in English are parsed when the pointer resides at a node decorated with $?e \rightarrow t$ induced by a computational rule. The verb itself contributes not only a concept formula (e.g. $Upset'$) but also creates a new node and moves the pointer to the object node (the transition induced is that of transition 2-3 in Figure 3.1).[3] Parsing of an object will then provide the appropriate formula value for this node. Computational rules can then compositionally determine the combination of those formulae

the assumption that the proposition expressed is relative to a time point given by a term denoting some temporal/modal relation to the time of utterance (Gregoromichelaki, 2006). A further simplification is that names and words with conceptual content are assumed to be in one-to-one correspondence with concepts, with no attempt to address the substantial issues in addressing the context-sensitivity of either.

[3]Formally: $\langle\texttt{IF}\ ?Ty(e \rightarrow t), \texttt{THEN}\ \texttt{make}(\langle\downarrow_1\rangle), \texttt{go}(\langle\downarrow_1\rangle), \texttt{put}(Fo(Upset'), Ty(e \rightarrow (e \rightarrow t))), go\langle\uparrow_1\rangle, \texttt{make}(\langle\downarrow_0\rangle), \texttt{go}(\langle\downarrow_0\rangle), \texttt{put}(?Ty(e)), \texttt{ELSE Abort}\rangle$.

to satisfy the requirements remaining in a strictly bottom-up fashion.

3.2 Semantic underspecification and update

As indicated earlier, pronouns, and indeed elements at ellipsis sites like auxiliaries, project temporary, underspecified formula values which are required (through the injunction $?\exists \mathbf{x}.Fo(\mathbf{x})$) to be instantiated at some point during the parsing process. This interim value is given as a metavariable, e.g., **U**, **V**..., which satisfies a type requirement, allowing the parse to continue, but leaves the content of a node open to be satisfied perhaps later in the derivation. The update for such a metavariable is given by selection of a proper value from context. Thus all context-dependent linguistic elements may allow as an option the assignment of a value from the parsing process subsequent to their original processing. Consider such a delayed resolution:

(11) *It* emerged *that John was wrong.*

In (11), the metavariable projected by *it* in string-initial position licenses the further unfolding of the parse process, allowing the parse of the verb *emerged*. The pointer then returns to the semantic subject node and permits the parse of the post-verbal complement clause whose content satisfies the requirement for determinate content, yielding an output formula $Emerge'(Wrong'(John'))$ (ignoring tense), without any final indication that an expletive pronoun appeared in the uttered string.

The same dynamics applies to quantifying terms. It is wellknown that indefinites can take wide scope over any previously introduced term:

(12) *Every teacher* confirmed that *two students* had written *a report* on *a famous philosopher*. $\forall < \exists < \exists_2 < \exists; \exists < \exists < \forall < \exists_2; \ldots$

But inversion of scope for other quantifiers is available only if an indefinite precedes:

(13) *A nurse* interviewed *every patient*. (ambiguous)
(14) *Most nurses* interviewed *every patient*. (unambiguous)

Within DS, this is addressed through combining underspecification and the selected logic. All noun phrases, even quantified ones, project terms of type e. There is no type-lifting mechanism reversing functor-argument relationships as in generalised quantifier theory: in all predicate-argument arrays constructed within this framework, argument-hood is non-negotiable. Rather, quantifying expressions are analysed in the manner of arbitrary names of predicate-logic natural deduction, as formulated in the *epsilon calculus* of (Hilbert and Bernays, 1939), a conservative, but more expressive, extension of predicate logic. Introduction of such terms in the language is based on the following equivalence:

$$\exists x.F(x) \equiv F(\epsilon x.F(x))$$

This indicates that an existential statement is equivalent to one in which a *witness* of the truth of the statement can appear as an argument of the

statement's predicate. Such a witness appears as an *epsilon term* containing as its *restrictor* the predicate itself and denotes some arbitrary entity satisfying the predicate (if such an entity exists).

Exploiting this equivalence, in DS, such terms are employed because they may carry a record of the context within which they occur inside their restrictor. According to the computational rules defined, an intermediate representation of a sentence such as *A man is waving* will take the form $Wave'(\epsilon, x, Man'(x))$ derived by simple functional application. But this will be eventually algorithmically transformed to a formula where an appropriate epsilon term, abbreviated as a below, appears as the argument of the conjunction of the predicates contributed by the common noun and verb:

$$Man'(a) \wedge Wave'(a)$$

where $$a \;=\; (\epsilon, x, Man'(x) \wedge Wave'(x))$$

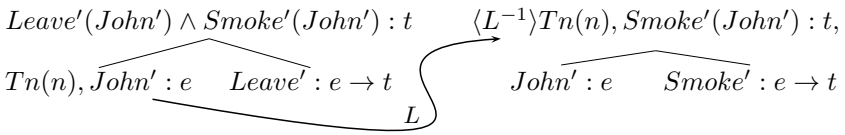
The restrictor of this term contains a record of the propositional structure that gave rise to it so that it provides a suitable antecedent for subsequent cases of *E-type anaphora* (see Kempson et al., 2001). Quantifier scope is not expressed as part of the tree architecture but through scope constraints collected incrementally during the parse process. Multiple quantification of course yields more complex restrictor specifications, but the evaluation algorithm applies to arbitrarily complex combinations of terms and with variation in the connective depending on the type of quantifier involved.

This separation of scope dependencies from the representation structure then allows inverse scope readings for sentences such as (13)-(12) to be treated as a form of context-dependency analogous to anaphoric construal. Scope constraints for indefinites are formulated as partially underspecified, a metavariable indicating availability of choice with respect to source of dependency. This is lexically encoded as $\mathbf{U} < x$ for the variable x associated with the indefinite. This simultaneously allows (a) selection of antecedents as the source of dependency among whatever other terms are already constructed within a domain, and (b) license to delay scope-dependency choice until some other noun phrase has been parsed: it is this license for delay (rather than some unrestricted quantifier storage device) which gives rise to scope inversion as in (12)-(13) but not (14).

3.3 Linking trees through shared terms

For the full array of compound adjunct structures displayed in NL, DS employs a license to build paired, so-called LINK*ed*, trees associated through a LINK modality, $\langle L \rangle$. This device is utilised for allowing incorporation within a tree of information that is to be structurally developed externally to it. Relative clause construal, being one core case of adjunction, involves constructing a LINKed tree bearing a requirement that it contains as a sub-term the formula from the source-node from which the LINK relation is defined:

(15) John, who smokes, left



In the building up of such paired LINKed trees for relative clauses, the requirement imposed for a common term is satisfied by the processing of the relative pronoun which induces an unfixed node annotated with the requisite copy of that term. This is then necessarily constrained to appear within the newly emergent propositional structure (reflecting *island constraints* associated with such structures).

A second pattern is provided by *apposition* structures, in which a sequence of NPs are construed as co-denoting:

(16) A friend of mine, a painter, is outside.

(17) The candidate, a linguist, is outside.

In DS terms, the second NP is defined as extending the term derived by processing the first through construction of a LINKed structure. This involves evaluating such paired terms as a single term incorporating both restrictors: $(\epsilon, x, \phi(x))$ and $(\epsilon, x, \psi(x))$ thus leading to the term: $(\epsilon, x, \phi(x) \wedge \psi(x))$, in (16) yielding $(\epsilon, x, Friend'(x) \wedge Painter'(x))$.

Given this dynamic perspective of progressively building up representations of content, we now examine consequent requirements on a model of *context*. As we shall see, given the strictly incremental approach to the processing of strings, context is definable as a composite record of *what has just taken place*, i.e., the representation of (partial) content so far established in tree form and the actions used to gradually build it up.[4]

3.4 Intra-sentential context

We conceive of each parsing step (the processing of each word in a sentence) as taking place in some context. As already set out, a parsing step is defined in terms of tree expansion: its input is a a partial tree including a pointer as built by the preceding steps so far; its output is another, more fully-specified (although still possibly partial) tree. As natural language is inherently ambiguous, a single sequence of words $w_0 \ldots w_i$ might be associated with more than one possible sequence of parsing actions $a_0 \ldots a_i$, $a'_0 \ldots a'_i$, $a''_0 \ldots a''_i$ etc. The parser state when parsing the next word w_{i+1} may therefore contain multiple (partial) trees T_i, T'_i, T''_i etc., according to the degree of ambiguity exhibited by the string of words or the firing of different parsing actions all of which yield well-formed partial trees, i.e. several independent parsing paths may ensue. However, only one such partial tree will provide the basis for a particular update by a particular word. It is thus important that the context in which any lexical action a_{i+1} is applied corresponds only to the

[4]The formal notion of context in DS also includes the set of words parsed so far but as we will not make use of this here we put it aside for simplicity.

partial tree which a_{i+1} is itself extending. If the role of context is to provide information built by previous interpretations, and the interpretations T_i and T_i' are mutually exclusive alternatives, an extension of T_i must not be able to access information from T_i'. Therefore a minimal model of context in DS consists of the current partial tree that is being extended, and ambiguity of parsing paths and/or interpretation may thus be conceived of as involving multiple, but independent, contexts.

As a grammar formalism, nothing internal to DS definitively determines selection of interpretation in cases of genuine ambiguity: in such cases the grammar must make available all the option required. Nonetheless, because of the in-built left-to-right incrementality, a fine-grained concept of *context* is definable incorporating the monotonically growing information provided not only by previous discourse but also by the evolving parse of the current utterance. Such a context provides a record of (a) the partial terms so far constructed, as well as (b) the actions used in constructing them, thus providing the requisite information for resolution of metavariables as well as (as we shall see below) other forms of underspecification.

4 Ellipsis and context

4.1 Inter-utterance context

When we consider dialogue, our concern with "context" includes what must be provided at speaker-turn boundaries. A new speaker can begin a new sentence, of course, so the default null context is available, in DS terms, the requirement $?t$. Yet a previous complete sentence may provide material enabling a hearer to reconstruct information, as in elliptical utterances, so the context must contain previous (complete) trees. Additionally speakers may continue or collaborate on incomplete utterances spoken by others, as in (7), and this means that we require having incomplete (partial) trees available. We therefore take context after a change of speaker also to contain the current tree (partial or not) from the parser state developed immediately before the change.

For grammar formalisms which define solely the well-formedness of full sentences, this might be problematic, as suitable representations for partial sentences cannot be available without further ado. But with partial trees being well-defined formal objects in the DS representation language, this is straightforward. Moreover, although defined primarily in terms of parsing processes, DS incorporates a parallel account of *generation*. In the DS implementation, generation follows exactly the same steps as parsing, with the added constraint that some desired output tree or *goal* tree (possibly a complete tree as in 4 in Figure 1) constitutes a filter on each transition, so is present from the initial stage (as the content of what the speaker intends to communicate). Hence, parsing/generation processes in DS use interchangeable representations and modelling the distinction between speaking and hearing does not necessitate switching to distinct formal vocabularies or

sets of actions (see Purver and Kempson, 2004, as in this paper we set out parsing derivations only).

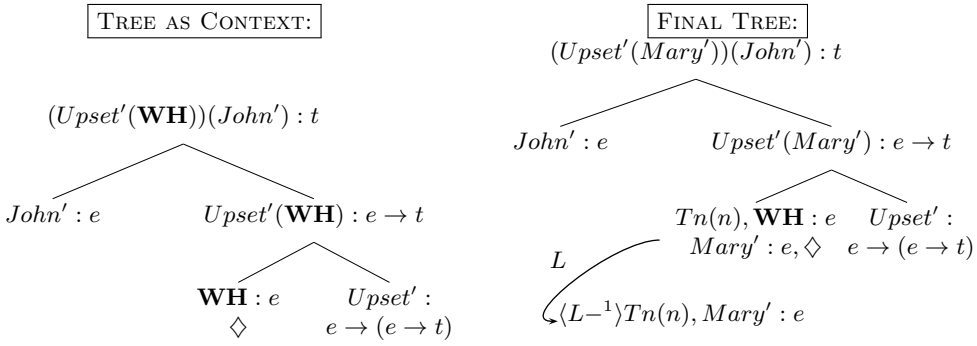
4.2 Access to structure

This perspective on context provides all we need to treat common dialogue fragment phenomena such as short answers, acknowledgements and clarification requests: in all these cases, the fragment updates the existing tree representation in the context provided by the preceding utterance. In question/answer pairs as in (18), the tree produced by parsing a *wh*-question includes underspecification (in DS, *wh*-terms encode a specialised **WH** metavariable).

(18) A: Who did John upset?
B: Mary.

If this tree is taken as present in the context in which B's fragment is parsed (19), we can see that parsing the word *Mary* can directly update this tree, through LINK adjunction, as introduced in section 3.3, to the node decorated with the interrogative metavariable, **WH**. The result is to provide a complete semantic formula and an answer to the question via evaluation of the two formula decorations as being equivalent to $Mary'$, which is by definition a legitimate way to further specify metavariables. Such an update is like the appositive further specification of a noun phrase as in *my friend, the doctor* where a pair of noun phrases is taken to identify the same entity, the only difference being that **WH** adds no further information to the term.

(19) Processing *Mary*:

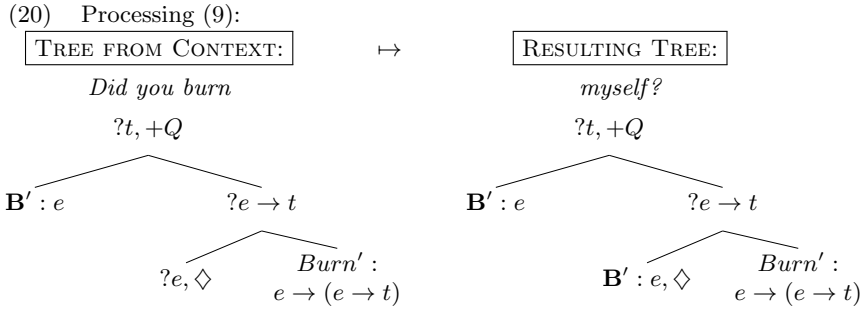


This approach needs no extension for split utterance cases such as (9) in which a reflexive pronoun must be understood relative to what is provided by the first half. Here, the tree in context is partial, as in (20), with the pointer at a node awaiting processing of the object. The actions associated with B's completion of A's utterance as *myself* merely copy the formula at the semantic subject node, just in case that term picks out the current speaker.[5] This is exactly as parsing *yourself* copies the subject formula just in case it picks out the current addressee, a parse of *Did you burn yourself?*

[5] $\langle$IF $?Ty(e)$, THEN $\langle$IF $\langle\uparrow_0\rangle\langle\uparrow_1^*\rangle\langle\downarrow_0\rangle\alpha_{Speaker}$, THEN put$(\alpha)$, ELSE Abort$\rangle$, ELSE Abort$\rangle$.

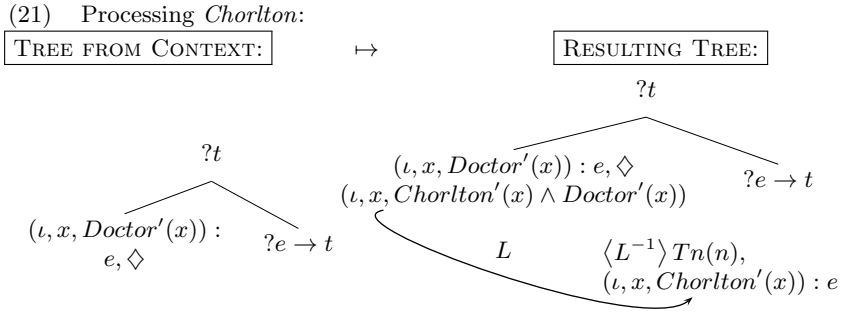
thus yielding exactly the same tree as the resulting tree in (20). There is, on this view, nothing mysterious about shifts in perspective within dialogue:

(20) Processing (9):



Examples such as (7) also follow, given that the contextual trees at speaker change can show a further degree of (temporary) underspecification: partially specified content. The tree constructed by B after processing the utterance-initial *the doctor* is highly partial, but provides a node of type e which B can use as the basis for a clarification request. Such requests are modelled in DS via the same LINK relations used for appositions. Like appositions, clarifications are interpreted as providing or requesting further information on a specific term. Modelling this via a LINKed sub-tree provides a possible update of the original term, much like an answer to a *wh* question provides an extension of the term queried, as we saw in (18), except that here the clarification question provides additional information: the name of the doctor (see Kempson et al., 2009):

(21) Processing *Chorlton*:



As discussed earlier, switch of roles between the two interlocutors does not impede the term-construction process: it is merely the presence of well-defined partial structures in context, as required by the strictly incremental nature of DS parsing, that makes the analysis possible.

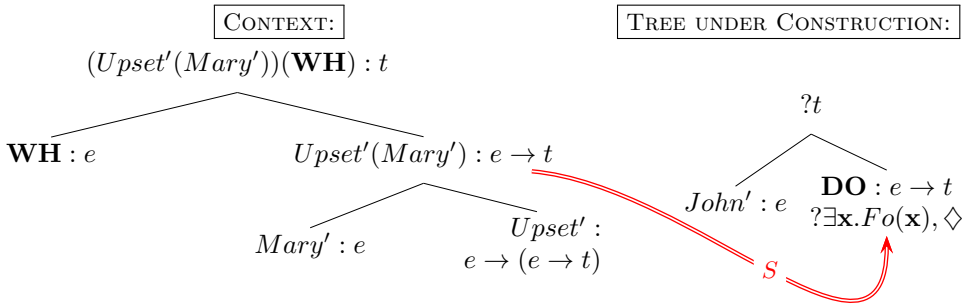
4.3 Access to formulae

Since context provides us with structure so far built – the labelled, partial trees – it also provides us with the logical formulae decorating these trees. As outlined in section 3.2, this is all we need for the analysis of anaphora: anaphoric elements decorate the tree with metavariables, which require for full resolution the presence in context of a suitably typed formula. The same applies to strict forms of VP-ellipsis, as these can also be modelled as

formula underspecification resolved by directly re-using a contextually provided predicate. The only difference between pronoun and ellipsis construal is their logical type assignment, e vs. $(e \rightarrow t)$ (the arrow in the diagram below represents the formal process of *Substitution* of a metavariable by some term in context and **DO** is a specialised metavariable contributed by *did*):

(22) A: Who upset Mary?
B: John did.

Parsing *John did*:



4.4 Access to parsing actions

However, cases of *sloppy* ellipsis must require something else: here the processing of the fragment leads to a different interpretation from that of its antecedent – neither an exact copy of the antecedent content nor an extension of it, but still maintaining some parallelism in the way the interpretation is established. We can see now that, given the DS procedural modelling of interpretation, this parallelism can be suitably expressed in terms that do justice to what constitutes context-dependency in such cases: the *actions* used in parsing the antecedent may now be invoked and re-run at the ellipsis site. This provides a new formula value, but one which is constrained to be built in the same way as the antecedent formula was already constructed:

(23) A: Who hurt himself?
B: John did.

Informally, the DS processing for the question in (23) involves the following actions after parsing of the subject *who*: constructing a two-place predicate as indicated by the verb, introducing an object argument, and then, because this object derives from a reflexive pronoun, it is obligatorily identified with the argument provided as subject (for the formulation of such actions see fn. 3,5). When it comes to processing B's elliptical *did*, the reconstruction we require is essentially the same: the same verbal predicate but this time an object which becomes identified with the subject B has provided, i.e., *John*. This can be achieved precisely by re-running the same set of parsing actions as used when parsing A's utterance – provided that these have been stored as a sequence in context and thus are accessible for re-use, in this case the sequence of actions given by *upset+himself*. This is a possibility that, according to DS, is associated with the resolution of the metavariable

DO contributed by *did*. The effect achieved is the same as the higher-order unification account but without anything beyond what has already been used for the processing of previous linguistic input: the parsing actions themselves stored in context and able to be re-run.

4.5 Context defined

To sum up, we take as context a record of (a) the partial tree under construction, with its semantic annotations, (b) the trees provided by previous utterances, and (c) the sequence of parsing actions used to build (a) and (b). Considering the incremental, action-based nature of DS, this reflects well the state of the parser. Trees provide structural context for the incorporation of dialogue fragments; their decorating formulae provide antecedents for anaphora and strict readings of VP-ellipsis; parsing actions allow reconstruction of the sloppy equivalents. A *context*, then, is a sequence of actions and tree representations contributed by a sequence of words and utterances in a discourse.

Examples like (10) might be taken to motivate an extra dimension of context, provided by no obvious linguistic input but instead by the situation or prior knowledge or belief, i.e. the general cognitive context. Nonetheless, the DS semantic representations are expressed in a language intended to implement the interface among information sources in any modality and this kind of example provides confirmation for this strategy. Any type of information is then by definition represented in the same way as the other aspects of context discussed here, namely, as formulae in tree format present within the contextual tree sequence. Such trees would provide the means by which an utterance of *Second left* may be construed as an answer to the elliptical question *McWhirter's?* which itself induced a tree with top node decorated by the formula: $Located'(McWhirter's, \mathbf{WH})$ obtained by exploiting knowledge representations available as context.

There are consequences to be explored from this perspective, both linguistic and formal. With this preliminary characterisation of context, some of the problems facing other accounts of ellipsis become resolvable. Parallelism effects associated with ellipsis such as (3) are expected to follow. The actions used to establish an interpretation for the first conjunct will form part of the immediate context for the interpretation of the second conjunct containing the ellipsis site. So, in the resolution of the ellipsis, these actions will be replicated yielding parallel results in the scopal interpretation of the indefinite. According to DS, the indefinite in (3) can be interpreted with free choice as its being dependent on the subject, the event term associated with the immediately containing proposition, or the event term for the overall formula.[6] Whatever choice is made in the first conjunct will be necessarily replicated in the second as the same actions need to be repeated. This pattern is expected for scopal dependencies of arbitrary complexity, unlike in

[6]DS semantic representations include event arguments (Gregoromichelaki 2006).

accounts that involve copying of logical form or strings.

On the other hand, the structural restrictiveness associated with *antecedent contained ellipsis* is predicted as it involves the processing of relative clauses. The relative pronoun in English, given its appearance at the left periphery of its clause, in DS is taken to decorate an unfixed node. Such nodes by definition have to be resolved within the local containing tree, and not across a LINK relation (this is the formal reflection of *island constraints* in DS). This independent constraint is respected in (5) but not in (6). In (6) a second relative clause (*who already had*) intervenes between *who* and its argument position. This induces an island violation and explains the inability of *who* to provide an argument for the predicate to be reconstructed from *interviewed* (Kempson et al., 2001; Cann et al., 2005). The island sensitivity of antecedent contained ellipsis in (6) thus follows from this independent constraint on relative clause processing in an entirely expected way.

From this point of view, the notion of context as a record of *previous* parse states remains unchallenged even by the occurrence of expletive pronouns or scope dependencies that are apparently determined non-linearly. In all such cases, the lexical specifications of pronouns/indefinites dictate only partial updates to the node they decorate with metavariables licensing delay – delay of content value assignment for subject expletives and delay of scope-dependency for indefinites. As, in DS, full resolution of underspecified content is not imposed until after the predicate value in some local domain has been determined, the context available for late resolutions of this type will contain terms constructed after the initial, first pass processing of the expletive/indefinite. These terms can then provide values for the underspecified content initially provided. So what superficially appears to be evidence for phenomena that do not conform to the usual context-dependency properties of underspecified expressions reverts to the usual pattern once the concept of context is defined in terms of immediately preceding parse states that can accommodate localised underdetermination.

4.5.1 Formal considerations

Nonetheless, this novel approach on context specification requires a reconsideration of the DS formal mechanism in order to be expressible in the most apposite terms. DS actions for building up structure currently retain a second-class status, being construction mechanisms which are not themselves quantified over. Several questions on the technical level might then be raised regarding the possibility of a neater formalisation.

One approach might be the following. DS relies on the interrelationship of two different structures: the sequential order of the input (or output) string, and the applicative structure of the tree with typed nodes that it thereby built up (or serialised). We can express both structures using a suitable logical framework based on dependent type theory with subtypes and equality: type dependency can be used to express the sequential order,

whereas subtyping can be used to express the applicative structure. To be precise, terms $x : e$, $\phi : e \rightarrow t$ and $\phi x : t$ correspond to triples $\langle s, x, \phi \rangle$ inhabiting, respectively, the types $\langle t, e, \{\phi : e \rightarrow t | \phi x = s\} \rangle$ – types which are expressible in a suitable dependent type theory and whose dependency corresponds to the serial order of the corresponding elements of the input string. Having done this, further questions present themselves. The first is whether the underlying type theory ought to be linear or intuitionistic: that is, whether it should allow explicit contraction and/or weakening. In fact, both seem to have roles: relative pronouns, for example, seem to behave linearly, whereas anaphora (and E-type phenomena generally) are classical. So we would seem to need a mixed linear/intuitionistic system: the validity of a suitable such system is plausible, but needs further investigation.

In a framework like this, we can pose further questions. One concerns the interaction between type extension and subtyping: systems with dependent intersection – such as that of Kopylov (2003) – assimilate type extension to subtyping, and we have an independent motivation for dependent intersection as a means of expressing the contribution of adjuncts (via LINK-structures). The next concerns the possibility of a constructive variant of the ϵ-calculus, which would allow ϵ-terms which were not everywhere defined (and which would provide a semantic basis for the generation of clarification requests). So we can ask about the compatibility of such a system with ϵ-terms (which includes the question as to how we formulate proof rules for such ϵ-terms). Finally, we can raise the following issue: if, in the spirit of Miller (2009), we formalise algorithms using proof search, can we recover the algorithmic side of DS in terms of a search for the proof of a suitable judgement (a proof, it may be, that the input string inhabits a suitable type)? Such a viewpoint could be a very powerful means of relating DS to other type-theoretic linguistic formalisms.

References

Blackburn, P. and Meyer-Viol, W. (1994). Linguistics, logic and finite trees. *Logic Journal of the Interest Group of Pure and Applied Logics*, 2(1):3–29.

Cann, R., Kempson, R., and Marten, L. (2005). *The Dynamics of Language*. Elsevier, Oxford.

Cann, R., Kempson, R., and Purver, M. (2007). Context and well-formedness: the dynamics of ellipsis. *Research on Language and Computation*, 5(3):333–358.

Clark, H. H. (1977). Bridging. In *Thinking: readings in cognitive science*, pages 169–174. Cambridge University Press.

Dalrymple, M., Shieber, S. M., and Pereira, F. C. N. (1991). Ellipsis and higher-order unification. *Linguistics and Philosophy*, 14(4):399–452.

Fernández, R. (2006). *Non-Sentential Utterances in Dialogue: Classification, Resolution and Use*. PhD thesis, King's College London, University of London.

Fiengo, R. and May, R. (1994). *Indices and Identity*. MIT Press, Cambridge, Mass.

Fox, D. (1999). *Economy and Semantic Interpretation*. MIT Press.

Ginzburg, J. and Cooper, R. (2004). Clarification, ellipsis, and the nature of contextual updates in dialogue. *Linguistics and Philosophy*, 27(3):297–365.

Ginzburg, J. and Sag, I. (2000). *Interrogative Investigations: the Form, Meaning and Use of English Interrogatives*. Number 123 in CSLI Lecture Notes. CSLI Publications.

Gregoromichelaki, E. (2006). *Conditionals: A Dynamic Syntax Account*. PhD thesis, King's College London.

Hilbert, D. and Bernays, P. (1939). *Grundlagen der Mathematik II*. Julius Springer, Berlin.

Kempson, R., Gregoromichelaki, E., and Sato, Y. (2009). Incrementality, speaker-hearer switching and the disambiguation challenge. In *Proceedings of SRSL 2009, the 2nd Workshop on Semantic Representation of Spoken Language*, pages 74–81, Athens, Greece. Association for Computational Linguistics.

Kempson, R., Meyer-Viol, W., and Gabbay, D. (2001). *Dynamic Syntax: The Flow of Language Understanding*. Blackwell.

Kopylov, A. (2003). Dependent intersection: A new way of defining records in type theory. In *LICS*, pages 86–95. IEEE Computer Society.

Merchant, J. (2004). Fragments and ellipsis. *Linguistics and Philosophy*, 27:661–738.

Merchant, J. (2007). Three kinds of ellipsis: Syntactic, semantic, pragmatic? Ms University of Chicago.

Miller, D. (2009). Formalizing operational semantic specifications in logic. In *Proceedings of the 17th International Workshop on Functional and (Constraint) Logic Programming (WFLP 2008)*, volume 246, pages 147–165.

Purver, M., Cann, R., and Kempson, R. (2006). Grammars as parsers: Meeting the dialogue challenge. *Research on Language and Computation*, 4(2-3):289–326.

Purver, M., Howes, C., Gregoromichelaki, E., and Healey, P. (2009). Split utterances in dialogue: a corpus study.

Purver, M. and Kempson, R. (2004). Incremental context-based generation for dialogue. In Belz, A., Evans, R., and Piwek, P., editors, *Proceedings of the 3rd International Conference on Natural Language Generation (INLG04)*, number 3123 in Lecture Notes in Artifical Intelligence, pages 151–160, Brockenhurst, UK. Springer.

Stainton, R. (2006). *Words and Thoughts: Subsentences, Ellipsis, and the Philosophy of Language*. Oxford University Press.