Research on Language and Computation (2006) 4:289–326
DOI 10.1007/s11168-006-9007-x

Grammars as Parsers: Meeting the Dialogue Challenge

MATTHEW PURVER[1], RONNIE CANN[2] and RUTH KEMPSON[3]
[1]*Center for the Study of Language and Information, Stanford University, Stanford, CA 94305, USA (E-mail: mpurver@stanford.edu);* [2]*Department of Linguistics, Edinburgh University, Edinburgh EH8 9LW, UK (Email: r.cann@ed.ac.uk);* [3]*Department of Philosophy, King's College London, London WC2R 2LS, UK (E-mail: ruth.kempson@kcl.ac.uk)*

Abstract. Standard grammar formalisms are defined without reflection of the incremental, serial and context-dependent nature of language processing; any incrementality must therefore be reflected by independently defined parsing and/or generation techniques, and context-dependence by separate pragmatic modules. This leads to a poor setup for modelling dialogue, with its rich speaker-hearer interaction and high proportion of context-dependent and apparently grammatically ill-formed utterances. Instead, this paper takes an inherently incremental grammar formalism, Dynamic Syntax (DS) (Kempson et al., 2001), proposes a context-based extension and defines corresponding context-dependent parsing and generation models together with a resulting natural definition of context-dependent well-formedness. These are shown to allow a straightforward model of otherwise problematic dialogue phenomena such as shared utterances, ellipsis and alignment. We conclude that language competence is a capacity for dialogue.

Key words: dynamic syntax, incrementality, parsing, generation, context, anaphora, ellipsis, alignment

1. Introduction

Study of dialogue in informal conversation has been proposed by (Pickering and Garrod, 2004) as the major new challenge facing both linguistic and psycholinguistic theory. Several phenomena common in dialogue pose problems for theoretical and computational linguistics; amongst them *alignment*, *routinization*, *shared utterances* and various *elliptical* constructions. Alignment describes the general tendency of dialogue participants to mirror each other's patterns at many levels, including lexical choice and syntactic structure, even when use of alternative patterns would not affect the semantic content:[1]

(1) A: The nun offering the cowboy a banana.
B: The robber offering the ballerina a book.
??The robber offering a book to the ballerina.

Routinization describes a phenomenon whereby interlocutors converge on agreed interpretations for words or sequences remarkably quickly under test conditions, often without explicit negotiation (Garrod and Anderson, 1987; Garrod and Doherty, 1994). Shared utterances are those in which participants shift between the roles of parser and producer, each participant's utterance being elliptical:[2]

(2) A: That tree has, uh, uh, ?
B: Tentworms.
A: Yeah.

(3) *Daniel*: Why don't you stop mumbling and
Marc: Speak proper like?
Daniel: speak proper?

(4) *Ruth*: What did Alex
Hugh: design for herself? A self-loading washing-machine.

These are especially problematic for approaches in which parsing and generation are seen as separate disconnected processes, in particular if the underlying grammar formalism yields as output the set of well-formed strings.[3] All fragments on this view have to be treated by both the parsing and production devices as in some sense complete, with empty categories postulated that, for the parser, get filled in by the context, and for the producer, do not need to be generated. In the modelling of shared utterances, the system that reflects the initial parser must parse an input which is not a standard constituent, indeed may not be licensed as a well-formed string at all, assigning the presented string a (partial) interpretation, completing that representation despite lack of appropriate input. With a shift to generation mode, the system has then to produce an output from which the previously parsed words and their syntactic form are in some sense taken into account but are not produced. The system reflecting the initial speaker must also be able to integrate these two fragments, but in this case the switch is out of some generation module mapping (representations of) content onto strings, over onto some parsing module, which treats the previously generated string as in some way parsed even though, up to this juncture, it has been characterised by the generation module only. Such invocation of the necessary empty-category devices, whether in the completion of incomplete fragments by the parser, or as not needing to be said by the generator, is a standard enough manoeuvre in accounts of ellipsis;[4] yet it is highly problematic, not least because it involves positing empty categories which not only are not independently required in the grammar formalism, but indeed must not be freely available.[5] The discussion of dialogue in such terms is however hypothetical, as the inter-relation between production and parsing relative to a movement-based grammar formalism remains almost entirely unaddressed.[6]

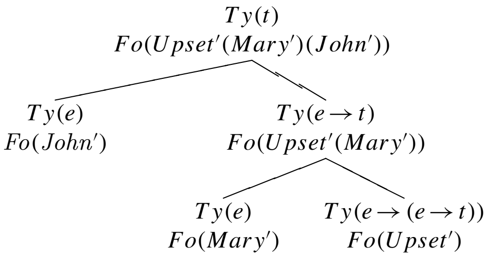
Things are not much better for constraint-based systems which impose a weaker criterion of success on their grammar formalism of merely articulating a set of constraints to be met by strings of the language, allowing sub-sentential strings to be characterised as well-formed within some larger structure that is not (Pullum and Scholz, 2001). But with participant switch being able to take place constituent-internally (as in (3), (4)), the advantage is not so very great. It is in such systems that correlated parsing and generation systems have been defined (e.g. Neumann, 1998). But even though parsing and production devices might be treated as related applications of the same neutrally-defined system of constraints, they must nevertheless be independently defined, with each having to be closed off before the other is invoked, each then having to treat the output of the other as in some sense parsed (or conversely produced), even though it is the *other* mechanism that has just been being activated. And, though some parsing systems are strictly incremental, generation systems are invariably head-driven (e.g. Stone and Doran, 1997), making the generation of utterances such as the first in example (4) especially problematic. Furthermore, in neither case is there reason to expect parallelism effects across such inverse applications of the use-neutral grammar device.

In this paper we respond to the Pickering and Garrod challenge. We start with an inherently incremental parsing-based grammar formalism, Dynamic Syntax (DS) (Kempson et al., 2001), and extend its accounts of parsing and generation (Otsuka and Purver, 2003; Purver and Otsuka, 2003) to define them as context-dependent processes within a suitably structured concept of context which naturally includes the fragments and shared utterances seen so often in dialogue. Finally we show how this context-dependent model can explain a range of alignment patterns as resulting directly from minimisation of effort on the part of the speaker (implemented as minimisation of lexical search in generation). As a coda, we briefly air some of the broader philosophical consequences of this dynamic perspective.

2. Background

DS is a parsing-directed grammar formalism in which a decorated tree structure representing a semantic interpretation for a string is incrementally projected following the left-right sequence of the words. Importantly, this tree is not a model of syntactic structure, but is strictly semantic, being a representation of the predicate-argument structure of the sentence:[7]

John upset Mary.



Grammaticality is defined as parsability, that is, the successful incremental construction of such tree-structure logical forms, using all the information given by the words in sequence. There is no central use-neutral grammar of the kind assumed by most approaches to parsing and/or generation.

The logical forms are lambda terms of the epsilon calculus (Meyer-Viol, 1995), so quantification is expressed through terms of type e whose complexity is reflected in evaluation procedures that apply to propositional formulae once constructed, and not in the tree itself. The analogue of quantifier-storage is the incremental build-up of sequences of scope-dependency constraints between terms under construction: these terms and their associated scope statements are subject to evaluation once a propositional formula of type t has been derived at the topnode of some tree structure. Scope dependency is thus reflected in the internal structure of the terms as finally derived, each reflecting whatever scope dependencies the collection of scope statements dictates.[8]

2.1. Parsing

The central tree-growth process of the model is defined in terms of the procedures whereby such structures are built up; taking the form both of general structure-building principles (*computational actions*) and of specific actions induced by parsing particular lexical items (*lexical actions*). The core of the formal language is the modal tree logic LOFT (Blackburn and Meyer-Viol, 1994), which defines modal operators $\langle\downarrow\rangle$, $\langle\uparrow\rangle$, which are interpreted as indicating daughter and mother relations, respectively, with two subcases $\langle\downarrow_0\rangle$, and $\langle\downarrow_1\rangle$ distinguishing daughters decorated with argument or functor formulae, and two additional operators $\langle L\rangle$, $\langle L^{-1}\rangle$ to license paired *linked* trees (see section 5.1).[9] The actions defined using this language are transition functions between intermediate states, which monotonically extend tree structures and node decorations. The concept of *requirement* is central to this process, $?X$ representing the imposition of a goal to establish X, for any label X. Requirements may thus take the form $?Ty(t)$, $?Ty(e \rightarrow t)$, $?\langle\downarrow_1\rangle Ty(e \rightarrow t)$, $?\exists x Fo(x)$, $?\exists x Tn(x)$, etc. All requirements that are introduced have to be satisfied during the construction process (see figure 1 where parsing and generation sequences are set side by side).

For example, the first action in parsing a sentence is a general computational action (termed INTRODUCTION) which develops the standard initial AXIOM state (here, as in all such partial tree-structures, there is a pointer, $\diamondsuit$, indicating the node under development):

$$?Ty(t), Tn(0), \diamondsuit$$

(i.e. a basic requirement to construct a propositional formula), to

$$?Ty(t), Tn(0), ?\langle\downarrow_0\rangle Ty(e), ?\langle\downarrow_1\rangle Ty(e \to t), \diamondsuit$$

thereby inducing the subgoals of constructing a type e argument (0) node and a type $e \to t$ predicate (1) node, by which a predicate-argument formula can eventually be derived. Words are specified in the lexicon to have lexical actions in similar style, each a sequence of tree-update actions in an $\langle IF..THEN..ELSE\rangle$ format, employing the explicitly procedural predicates `make`, `go`, `put`, defining an ordered (multi-)set of actions. A simple lexical action for a proper name *John* is given as follows:

John	**IF**	$?Ty(e)$
	THEN	`put`$(Ty(e))$;
		`put`$(Fo(John'))$;
		`put`$([\downarrow]\bot)$
	ELSE	ABORT

This entry first checks that there is a requirement $?Ty(e)$ for the correct type at the active node, then adds decorations which specify a semantic formula $Fo(John')$ of this type, and that this is now a terminal node (shown by the modality $[\downarrow]\bot$ "below this node nothing holds"). A subsequent general computational action (THINNING) then removes the now satisfied type requirement. A more complex lexical action for a transitive verb *dislike* takes the following form, first making a new predicate node of type $e \to (e \to t)$, and then an argument node with a requirement for type e (to be filled by parsing the object):

dislike	**IF**	$?Ty(e \to t)$
	THEN	`make`$(\langle\downarrow_1\rangle)$; `go`$(\langle\downarrow_1\rangle)$;
		`put`$(Fo(\lambda x \lambda y.Dislike'(x)(y)))$;
		`put`$(Ty(e \to (e \to t)))$; `put`$([\downarrow]\bot)$;
		`go`$(\langle\uparrow_1\rangle)$;
		`make`$(\langle\downarrow_0\rangle)$; `go`$(\langle\downarrow_0\rangle)$; `put`$(?Ty(e))$
	ELSE	ABORT

This format of lexical specification is general: all lexical items are defined as providing such actions, the concept of lexical content being essentially procedural. These obligatory lexical actions, together with optional computational actions, induce a sequence of partial trees in a monotonic growth relation as each word is consumed in turn.

At every non-final step, input and output tree may be underspecified; and each parameter for tree decoration (values of the predicates Fo, Ty, Tn) is a possible source of underspecification. An example of underspecification of content (i.e. Fo value) is provided by anaphora. In this system, the lexical specification of a pronoun is defined to project a metavariable, together with an accompanying requirement $?\exists x Fo(x)$:[10]

he	**IF**	$?Ty(e)$
	THEN	$\mathtt{put}(Ty(e))$;
		$\mathtt{put}(Fo(\mathbf{U}_{Male'}))$;
		$\mathtt{put}(?\exists x.Fo(x))$;
		$\mathtt{put}(?\langle\uparrow_0\rangle Ty(t))$
	ELSE	ABORT

This requirement must be satisfied by substituting a fully specified Fo value from context as part of the construction process (see section 3 for a formal definition).[11] The additional constraints in the lexical action shown above include case constraints determining relative configurational position in the resulting tree, here $?\langle\uparrow_0\rangle Ty(t)$ (which is equivalent to requiring that this node fill the subject position). However, other than an analogue of the Binding Principles (Chomsky, 1981) determining the local environment in which a value may *not* be provided, there is no constraint on the process determining what *does* provide this value, and, as we shall see, there is more than one way in which this might be achieved.

A more radical form of underspecification, following up the concept of tree-growth dynamics, is provided by allowing tree node relations (Tn values) to be only partially specified, with subsequent update fixing that initial weak specification. Long-distance dependency effects are characterised in these terms: a tree-node with decorations provided by that left-peripheral expression being introduced in a partial tree as "unfixed", the relation of the newly introduced node to the node n from which it is introduced specified only as a constraint on some fixed extension (following D-Tree grammar formalisms (Marcus, 1987)):[12]

$$\langle\uparrow_*\rangle Tn(n), ?\exists x Tn(x)$$

As with other requirements, such underspecification of tree-relation must get resolved within an individual tree constructed as part of the left-to-right construction process by a general computational action MERGE; this identifies an existing node with requirement $?Ty(X)$ with a compatible dominated but as yet unfixed node decorated with some formula α of that type $Ty(X)$.

The closing stages of tree decoration, once tree node relations in a tree are fixed and all terminal node decorations fully determined, involve a modal form of type deduction progressively compiling decorations on mother nodes reflecting functional application of formulae on their daughter nodes. Once all requirements are satisfied and all partiality and underspecification is resolved, trees are *complete* (i.e. a topnode formula of type t is derived), parsing is successful, and the input string is said to be grammatical. Provisionally, we say that a string is well-formed just in case it can be parsed using the computational rules of the system and lexical actions of each word in turn to produce a propositional tree that contains no outstanding requirements, a concept we return to in due course. The analogy of the account of long-distance dependency to anaphora resolution is deliberate: in this characterisation of long-distance dependency, the concepts of underspecification and update are extended from semantics/pragmatics to syntax, and expressed in similar formal terms. The immediate advantage of this perspective is the anticipation of feeding relations between anaphora construal, quantifier construal, and structural processes (Cann et al., 2005).

To characterise complex adjunction structures, such as relative clauses, pairs of trees are defined, with a transition from a node in one tree to the top node as the first in an emergent *linked* tree, with requirements imposed on how this new tree is to be developed, a computational action which feeds into other transitions as defined. Thus, in relative clauses, a *linked* tree is introduced requiring a copy of the formula at the node from which the LINK transition is built, with this copy being provided at an unfixed node by the relative pronoun, whose position in that structure is subsequently resolved in the regular way. Many syntactic phenomena can be explained in terms such as these (Kempson et al., 2001; Kempson and Meyer-Viol, 2002; Marten, 2002; Kempson et al., 2003; Cann, 2005; Cann et al., 2005). For the purposes of the current paper, the important point is that the process is monotonic: the parser state at any point contains all the partial trees which have been produced by the portion of the string so far consumed and which remain candidates for completion.

2.2. Generation

With the base formalism set out in a parsing perspective, we can define a generation system reflecting production that applies the very same parsing mechanism, as we shall see, leading to tight coordination between parsing and production.[13] Our point of departure is (Otsuka and Purver, 2003; Purver and Otsuka, 2003), which gives an initial method of context-independent tactical generation in which an output string is produced according to an input semantic tree, the *goal tree*. The generator incrementally produces a set of corresponding output strings and their associated partial trees (again, on a left-to-right, word-by-word basis) by following standard parsing routines and using the goal tree as a subsumption check. At each stage, partial strings and trees are tentatively extended using some word/action pair from the lexicon; only those candidates which subsume the goal tree are kept, and the process succeeds when a complete tree identical to the goal tree is produced (see Figure 1). Generation and parsing thus use the same tree representations and tree-building actions throughout.

In building n-tuples of trees corresponding to predicate-argument structures, the system is similar to LTAG formalisms (Joshi and Kulick, 1997). However, unlike LTAG systems (e.g. (Stone and Doran, 1997)), both parsing and generation are not head-driven, but word-by-word incremental. This has the advantage of allowing fully incremental models for all languages, matching psycholinguistic observations (Ferreira, 1996) irrespective of the position in the clausal sequence of the verb.

Our current model (and implementation) takes these basic parsing and generation models as starting points, but modifies them significantly. Our main departure is the incorporation of a model of context as a basic part of both parsing and generation, to which we turn next.[14]

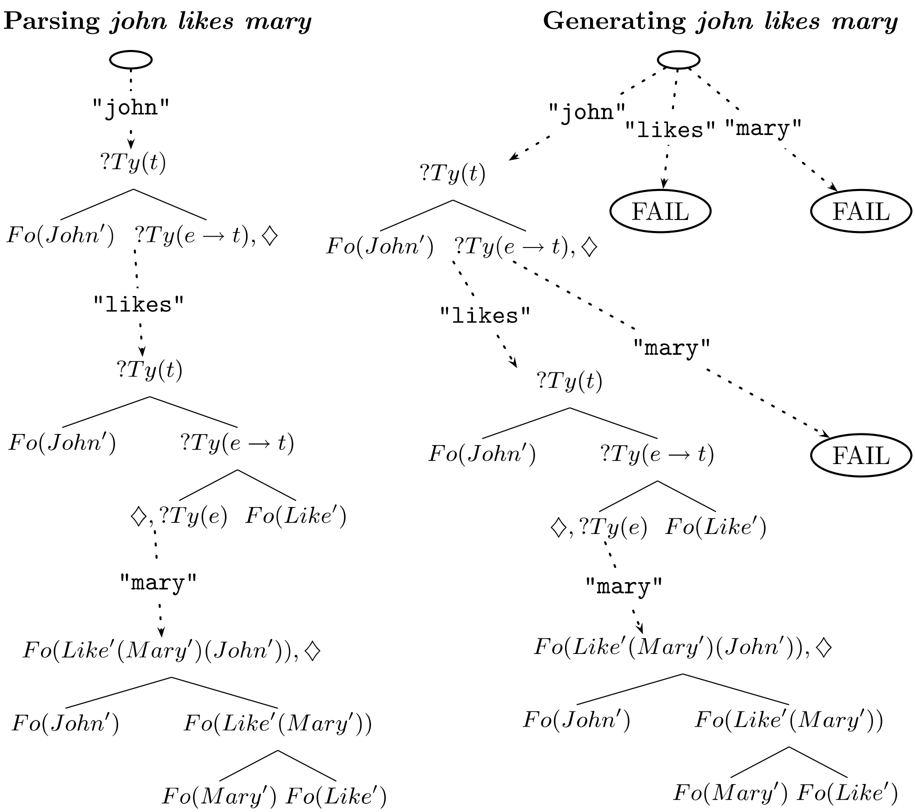


Figure 1. Parsing/generating *John likes Mary*

3. Modelling Context for Parsing and Generation

The basic definitions of parsing and generation as set out in (Kempson et al., 2001; Otsuka and Purver, 2003; Purver and Otsuka, 2003) assume some notion of context but give no formal model or implementation. In this section we define such a model and extend the parsing and generation definitions correspondingly, together with the definition of grammatical well-formedness that they encapsulate.

Standard formal models of context (e.g. (Kamp and Reyle, 1993; Heim, 1982; Stalnaker, 1978)) concern semantic representations, and in our case an obvious minimal requirement is that a model of context must include the semantic tree representations that have been produced. As shown in section 4, such a model is, when combined with the inherent interdependence of the DS parsing and generation models, powerful enough to give a suitable analysis for shared utterances and for certain elliptical fragments. However, computational models have often extended this notion of context to include information about surface strings, particularly to enable the suitable use in generation of e.g. information structure and subsequent clarificational dialogue (van Deemter and Odijk, 1997; Stone, 2003). Furthermore, some phenomena (in particular clarification requests which ask about a speaker's intended meaning or reference) have been taken to motivate the inclusion of phonological and syntactic information (Ginzburg and Sag, 2000; Ginzburg and Cooper, 2004). Given the dynamic perspective adopted here, with natural language syntax interpreted as a process rather than a system of representation, it is the transition from one partial tree to another that is central; and the procedural concept of action transforming one structure into another is fundamental. Accordingly, a natural move is to extend the concept of context to incorporate a record, not just of semantic structure, but also of the *actions* used to construct that structure from some uttered string.

Immediate evidence in support of this move is the basis it promises to provide for explaining ellipsis construal, while reflecting the pre-theoretic observation that elliptical forms are interpreted directly from the context, whether given strict or sloppy interpretations (in (6), *'Tom worried about John's sister'* and *'Tom worried about Tom's sister'* respectively):

(6) John worried about his sister, and Tom did too.

On movement accounts of ellipsis, in which there is full replication of syntactic structure at the ellipsis site, which fails to get realised (by application of PF deletion: see (Fiengo and May, 1994; Merchant, 2001) among others), this common-sense observation is not reflected in the analysis at all. On orthodox semantic accounts, ellipsis construal is analysed as part of the semantic evaluation process and in some sense context-dependent, albeit grammar-internally (Dalrymple et al., 1991; Shieber et al., 1996; and others following). However, these accounts do not seem to match the pre-theoretic intuition either, as ellipsis construal involves an operation that manipulates the context. Possible predicates are constructed from the antecedent clause by abstraction (restricted to the subject) over propositional contents: the ellipsis site is interpreted by application of that predicate to the presented fragment. In the sloppy interpretation of example (6), a predicate $\lambda x[Worry\text{-}about'(x, Sister\text{-}of'(x))]$ must be created; in the strict interpretation, a predicate $\lambda x[Worry\text{-}about'(x, Sister\text{-}of'(John'))]$.[15] To replicate this form of account in the DS framework by an equivalent structural process would be problematic, though possible, conflicting with the otherwise upwardly monotonic tree-growth process of construal.

However, once previous actions are included in the context, both interpretations of the second conjunct can be analysed as straightforwardly picked up from context (in some sense to be made precise below). Strict readings can be analysed just as pronouns are in section 2: typed metavariables are projected which must take their values from some tree node in context. As parsing the first conjunct in (6) must have created a node labelled with the semantic predicate *'worry about John's sister'*, this can be re-used directly in the second conjunct. Sloppy readings can be analysed in terms of re-running actions used to derive a previously constructed tree: parsing the substring *worried about his sister* in the first conjunct must have involved a series of actions which (a) built a predicate node $Worry\text{-}about'$; (b) added an object argument below it which contained (among other things) a metavariable projected from the pronoun *his*; and (c) resolved the value of this metavariable to the contextual $John'$. Re-running this sequence in the new context of the second conjunct, now including Tom', allows the metavariable this time to be resolved differently, giving the sloppy reading. This approach will therefore be able to reflect the strict vs. sloppy ambiguity without any operation on the context, but simply using whatever the context already provides.

3.1. Parsing in context

With this preliminary justification for incorporating actions within the specification of context, we now define what constitutes a parser state. The original (Kempson et al., 2001) model takes a parser state to be a set of (partial) trees reflecting semantic content (in this sense *semantic trees*). We now extend this so that a parser state P is a set of triples $\langle T, W, A\rangle$, where T is a (possibly partial) semantic tree, W the sequence of words and A the sequence of lexical and computational actions that have been used in building it. Context can now be defined in these terms. At any point in the parsing process, the context $\mathcal{C}$ for a particular partial tree T in the set P can be taken to consist of:

(a) a set of triples $P' = \{\ldots, \langle T_i, W_i, A_i \rangle, \ldots\}$ resulting from the previous sentence(s); and
(b) the triple $\langle T, W, A \rangle$ itself.

Discourse-initially, the set P' will be empty, and the context will therefore be identical to the standard initial parser state, the singleton set P_0 containing only a single triple $\langle T_0, \emptyset, \emptyset \rangle$ (where T_0 is the basic Axiom = $\{?Ty(t), \Diamond\}$, and the word and action sequences are empty). As words are consumed from the input string and the corresponding actions produce multiple possible partial trees, together with their corresponding word and action sequences, the parser state set will expand to contain multiple triples; note that the context $\mathcal{C}$ available to any tree will still be restricted to its current triple (as P' is empty). Once parsing is complete, we use the final set P_1 to define the new starting state (and context) for the next sentence as $P_1 \cup P_0$ (i.e. P_1 with the addition of the triple containing the basic axiom).[16]

Note that we take this definition to provide the *minimal* context available to any particular tree. There is no doubt that context can be extended beyond this by a participant's general cognitive processing – further propositional structures might be introduced by inference or by representing aspects of the visual or attentional situation. As this extension of context is not linguistically controlled, we do not attempt to model it here – but we assume that it will always be available to provide representations of, and information about, for example, the current speaker and hearer (for resolution of personal pronouns) and other salient entities (for deixis). Given that the tree representations are inhabited by concepts and not by words, we take such information to be expressible in the same tree-based format.

Note also that the simple protocol defined here will monotonically extend the contextual set of triples, keeping all constructs for an arbitrarily long time, all equally salient. A fuller model would take account not only of salience and recency issues, but also of the psycholinguistic observation that higher-level 'syntactic' information (in our case, actions) might be expected to decay relatively fast, with propositional information (trees) remaining accessible for longer (Fletcher, 1994). In particular, we assume that partial trees representing unsuccessful parse strategies will be discarded fast in the presence of alternative successful complete trees, and that in general the only partial trees in context will come from immediately preceding utterances. However, as decisions on when particular alternatives should be accepted or discarded must depend on grammar-external processes such as indirect answerhood, inference and consistency checking, we do not attempt to model it here: see e.g. (Ginzburg, forthcoming).

3.2. Re-use of terms (substitution)

Given a definition of the context $\mathcal{C}$ available to any tree T, we can now define the rule of Substitution that we took earlier to provide a fully specified semantic formula for the underspecified metavariable induced by a pronoun:[17]

$$\textsc{Substitution}\ \left|\ \begin{array}{lll} \textbf{IF} & Ty(X), ?\exists x.Fo(x), & \\ & \langle T, W, A\rangle \in \mathcal{C}, & \\ & \{Ty(X), Fo(Y)\} \in T & \\ \textbf{THEN} & \textbf{IF} & \uparrow_0\uparrow_*^1\downarrow_0 Ty(X), Fo(Y) \\ & \textbf{THEN} & \text{ABORT} \\ & \textbf{ELSE} & \mathtt{put}(Fo(Y)) \\ \textbf{ELSE} & \text{ABORT} & \end{array}\right.$$

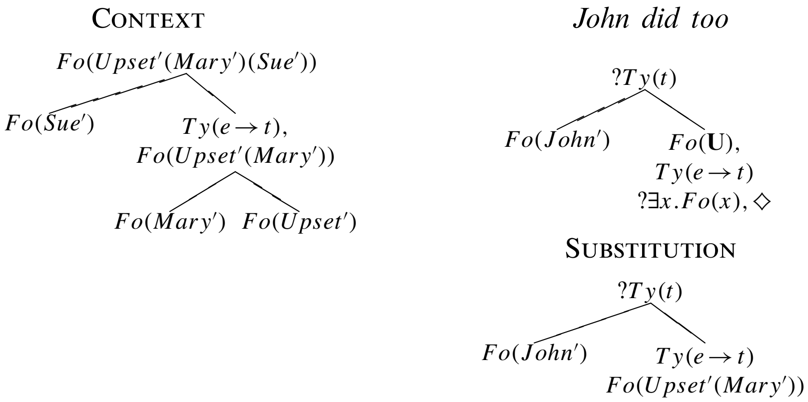
As set out here, X and Y are placeholders which range over types and formula values respectively. So, this action checks for an antecedent of the correct type in context (and ensures that it does not appear in a relative position in the tree which would violate the locality restrictions on non-reflexive pronouns – the $\uparrow_0\uparrow_*^1\downarrow_0$ test), and uses it to provide a fixed $Fo(Y)$ value – which satisfies the requirement $?\exists x.Fo(x)$ originally induced by parsing the pronoun (see section 2.1 above). Formally, this satisfaction allows the requirement to be removed by the standard process of Thinning, a rule that deletes requirements in the presence of a satisfying decoration. Thus it is only in a suitable context that all requirements can be satisfied – and therefore in which a string including a pronoun can lead to a complete tree and be said to be grammatically well-formed.

Exactly the same analysis can be applied to strict readings of VP ellipsis. The lexical entries for elliptical auxiliaries are defined in similar terms to those for pronouns: a metavariable is projected, together with a requirement for a fully specified formula. The only difference here is the type $e \to t$:[18]

$$\textit{do}\ \left|\ \begin{array}{ll} \textbf{IF} & ?Ty(e \to t) \\ \textbf{THEN} & \mathtt{put}(Ty(e \to t)); \\ & \mathtt{put}(Fo(\mathbf{U})); \\ & \mathtt{put}(?\exists x.Fo(x)); \\ \textbf{ELSE} & \text{ABORT} \end{array}\right.$$

Again, Substitution can apply (as it is type-general) as long as the context contains a tree with a suitable fixed Fo value of the correct type (e.g. $Fo(Upset'(Mary'))$), to yield a well-formed complete tree.[19]

(7) Sue upset Mary. John did too.



Given this account, the utterance of a pronoun (or ellipsis) where a preceding sentence contains two possible antecedents will be ambiguous – SUBSTITUTION can be applied in two different ways to derive two different logical forms. We take this to be correct. The resulting ambiguity presumably has to be resolved at some point by evaluation of inferential potential (minimally passing a consistency check). In this, the account reflects relevance considerations (Sperber and Wilson, 1995).

3.3. GENERATION IN CONTEXT

With these definitions to hand, we can proceed to the definition of a generator state. A generator state G is a pair (T_g, X) of a goal tree T_g and a set X of pairs (S, P), where S is a candidate partial string and P is the associated parser state (a set of $\langle T, W, A\rangle$ triples). Discourse-initially, the set X will contain only one pair, of an empty candidate string and the standard initial parser state, $(\emptyset, P_0)$. As generation progresses, multiple pairs are produced as candidiate partial strings S are considered, each with their own associated parser state P. In generation, the context $\mathcal{C}$ for any partial tree T in a state P is defined exactly as for parsing: the set of triples $P' = \{\ldots, \langle T_i, W_i, A_i\rangle, \ldots\}$; and the current triple $\langle T, W, A\rangle$. Once generation is complete, the state P_1 paired with the chosen string S_1 is taken to form the new context for the next sentence $P_1 \cup P_0$ (just as with parsing), hand-in-hand with the new initial generator state $X_1 = (\emptyset, P_1 \cup P_0)$.

Note here the close relationship between the parsing and generation processes. They share the same basic component of their state (a parser state P, a set of tree/word-sequence/action-sequence triples – the generator state merely adds to this (partial) candidate strings and a goal tree), and they share the same representation of context. In addition, as both processes are strictly incremental, there is no requirement that their initial states be empty or contain only complete trees – they can in theory start from *any* parser or generator state. Switching between the two processes, even in mid-sentence, therefore becomes straightforward, as we will show below in our analysis of shared utterances (section 4.2).

Note also that as the processes *necessarily* use the same parsing actions, they must make parallel use of context. Thus the generation of *He smiled* in *John came in. He smiled* is licensed not simply because the metavariable provided by *He* allows its partial tree to (trivially) subsume the goal tree, but because, following the parsing dynamics, a value for this metavariable can be identified from context. The parse of the antecedent string provides such a value $Fo(John')$ by SUBSTITUTION which (less trivially) allows subsumption. In less suitable contexts, the requirement for such a value for the pronoun could not be satisfied and generation could not be complete.

This constraint of using context for generation as well as for parsing matches the general methodology of reflecting the ongoing dynamics of natural-language processing, and indeed it matches recent psycholinguistic results. Parsing has been known for some time to be incremental (Crocker et al., 2000; Phillips, 2003), and there is now increasing evidence of the incremental nature of natural language production (Ferreira, 1996; Aoshima et al., 2004).

4. Re-use of Structure

With context defined in terms that match that of a parser state, we now expect that context-dependent phenomena may pick up on any aspect of such states, not merely, that is, on formula values but also on structure in more general terms. The framework indeed commits us to expecting that structure provides a context for subsequent updates, given the concept of *linked* trees already defined. In this section we show how this structural re-use allows us a straightforward analysis of bare answers and shared utterances.

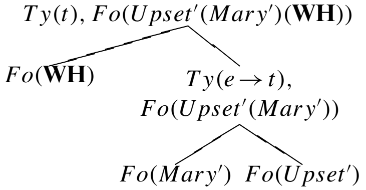
4.1. BARE ANSWERS

The first such case is the phenomenon of fragment answers to questions. As pointed out in the previous section, the parsing and generation processes are both fully incremental, and can start from any state (not just the basic axiom state P_0). The initial state (and context) for either process is formed by combining the basic axiom state P_0 with the final state from the previous utterance P_1. Just as one process can re-use actions saved in context by another, it is therefore straightforward for one to re-use the (possibly partial) trees produced by another as points from which its parsing actions can proceed. This provides a basis for characterising the bare answers common in question-answer pairs. The question tree in context provides the initial structure, and the answer fragment merely serves to provide the closing stages of building up a propositional formula.

The parsing or production of the question provides an open structure. Following (Kempson et al., 2001),[20] we analyse *wh*-expressions as providing a particular form of metavariable. In consequence, in a question such as (8), parsing *A*'s *wh*-question yields a type-complete but open formula as shown in (8′).

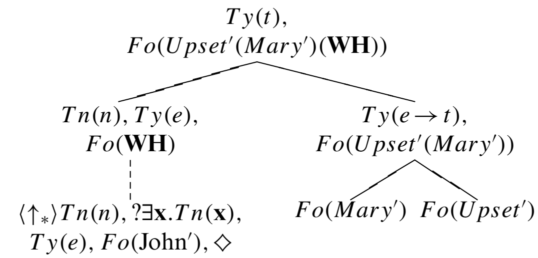
(8) A: Who upset Mary?
B: John.

(8′)



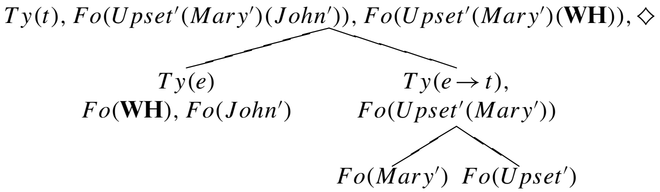
In using the structure provided by that context, we presume that *wh*-questions provide the license to move the pointer down the tree to the terminal **WH**-decorated node.[21] From there, a standard move in DS is to introduce an unfixed node by a process of LATE*ADJUNCTION which, from a node decorated with a metavariable of type *X* introduces an unfixed node requiring a decoration of that type.[22]

(8″)



With such a sequence of actions, the tree will accordingly be finally compiled to give the desired $Fo(Upset'(Mary')(John'))$ at the root node:

(8‴)



This procedure is entirely commensurate with the general monotonicity principle: the term $John'$ stands in a licensed growth relation from the original metavariable **WH**, as does $Fo(Upset'(Mary')(John')$ from

$Fo(Upset'(Mary')(\mathbf{WH}))$, so nothing precludes such pairs of formulae from decorating the same node.

Although space precludes a full exposition of quantification, this analysis is buttressed by the explanation it provides for so-called functional questions:

(9) A: Who did every student ignore?
B: Their supervisor.

Standardly these are said to involve a distinct type of *wh*-question enforcing a particular kind of answer (Higginbotham and May, 1981; Groenendijk and Stokhof, 1984; Ginzburg and Sag, 2000), but on this analysis no such distinction is required. The answer updates the structure provided by the question, updating the **WH**-term with an epsilon term; this can take narrow scope with respect to the subject during scope evaluation of the overall structure in the regular way, once the propositional formula is completed.

4.2. Shared utterances

As stated, it is perfectly possible for the initial state and context for either parser or generator to contain only partial trees. In other words, parsing and generation can use a radically incomplete previous utterance as context, with both processes thereby building up the same structure. The modelling of continuations in shared utterances so characteristic of dialogue is therefore straightforward. There are two aspects to such shared utterances: the modelling of continuations, and the phenomenon of transition between speaker and hearer roles. First, the modelling of the incomplete tree provided by the *non*-well-formed first part can be used directly as input to the parsing/generation of the continuation. In the case that this eventually leads to a complete tree with no outstanding requirements, the continuation can be considered well-formed. Of course, this will only happen in a very restricted set of contexts where the partial tree provides the correct type requirements and tree properties.

(10) *Ruth*: What did Alex ...
Hugh: Design? A kaleidoscope.

Secondly, there is the shift of roles. Here the tight coordination of parsing and generation in context comes into its own. As parsing and generation share the same lexical entries, the same context and the same semantic tree representations, the switch of speaker/hearer roles also becomes straightforward.[23] The phenomenon of shared utterances therefore falls into place as an entirely expected consequence of our context-dependent parsing and generation definitions.[24]

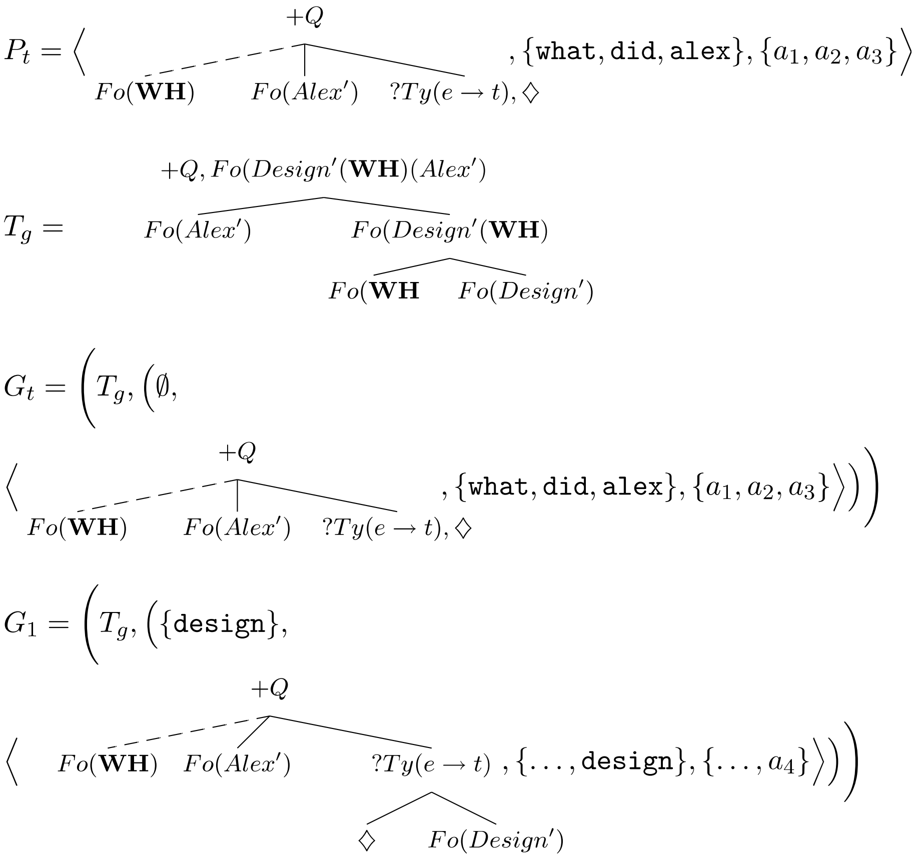


Figure 2. Transition from hearer to speaker: *What did Alex ... / ... design?*

We take first the transition from Hearer to Speaker. Normally, the generation process begins with the initial generator state as defined above: $(T_g, \{(\emptyset, P_0)\})$, where T_g is the goal tree, $\emptyset$ the empty candidate string, P_0 the standard initial "empty" parser state $\{\langle T_0, \emptyset, \emptyset\rangle\}$. As long as a suitable goal tree T_g is available to guide generation, the only change required to generate a continuation from a heard partial string is to replace P_0 with the parser state (a set of triples $\langle T, W, A\rangle$) as produced from that partial string: we call this the *transition state* P_t. The initial hearer A therefore parses as usual until transition, then given a suitable goal tree T_g, forms a transition generator state $G_t = (T_g, \{(\emptyset, P_t)\})$, from which generation can begin directly – see Figure 2 as a display of this process for example (10).[25] Note that the context does not change between processes modulo information about identity of current speaker and addressee.

For generation to begin from this transition state, the new goal tree T_g must be subsumed by at least one of the partial trees in P_t (i.e. those built so far by the parser). Constructing T_g prior to the generation task will often be a complex process involving inference and/or abduction over context and world/domain knowledge – Poesio and Rieser (2003) give some idea as to how this inference might be possible – for now, we make the simplifying assumption that a suitable tree can be made available.

We turn now to the transition from Speaker to Hearer. At the point of transition, the initial speaker B's generator state G'_t contains the pair (S_t, P'_t), where S_t is the partial string output so far, and P'_t is the corresponding parser state (the transition state for B).[26] In order for B to interpret A's continuation, B need only use P'_t as the initial parser state which is extended as the string produced by A is consumed.

As there will usually be multiple possible partial trees at the transition point, A may continue in a way that does not correspond to B's initial intentions – i.e. in a way that does not match B's initial goal tree. For B to be able to understand such continuations, the generation process must preserve all possible partial parse trees (just as the parsing process does), whether they subsume the goal tree or not, as long as at least one tree in the current state *does* subsume the goal tree. A generator state must therefore rule out only pairs (S, P) for which P contains no trees which subsume the goal tree, rather than thinning the set P directly via the subsumption check as proposed by (Otsuka and Purver, 2003).

There are a number of transition effects. Just as with alignment, the change in reference of the indexicals *I* and *you* across the speaker/hearer transition in (11) emerges straightforwardly from the nature of their lexical actions, with their use at any point involving reference to the speaker or addressee at the time of use:

(11) A: I think you should read ...
B: Your latest chapter. OK, I will.

We also expect that there is no constraint on when in the utterance the transition point can occur, as might be the case in head-driven approaches where transition prior to the sentential head would be problematic. So the occurrence of (4) (repeated here), with in addition the anaphor requiring identification with some locally available term is straightforward, despite the lack of recoverable head from the initial fragment, and lack of antecedent in the completing fragment:

(4) *Ruth*: What did Alex
Hugh: design for herself? A self-loading washing-machine.

In addition, as quantifier scope-dependency constraints form part of the contextual tree under construction and are not evaluated until a complete type t formula has been derived, dependencies between the portions either side of transition are unaffected, even when some quantifying expression is taken to be dependent on a quantifying term introduced after the role switch:

(12) A: We must make sure a nurse ...
B: Sees every patient. Absolutely.

This latter case turns on the (Kempson et al., 2001) account of quantification, in which indefinites are exceptional in projecting a metavariable in their scope-dependency statement allowing choice of term on which to be construed as dependent on some term subsequently constructed.[27]

5. Re-use of Actions

Given our inclusion of a record of actions in the specification of context, structural re-use is not the only strategy now available to our context-dependent parsing and generation processes. Our preliminary justification for incorporating this record was the accounts of VP ellipsis and sloppy pronoun construal that it allows, and in this section we set these out.

5.1. VP ellipsis

In order to formalise this approach, we need two things. The first is an equivalent to the SUBSTITUTION rule that allows us to provide fully specified values for metavariables by re-use of actions, rather than by re-use of semantic formulae. This we term REGENERATION:

$$\text{REGENERATION}\quad\left|\begin{array}{ll}\textbf{IF} & Ty(X), ?\exists x.Fo(x),\\ & \langle T, W, A\rangle \in \mathcal{C},\\ & \langle a_i, \ldots, a_{i+n}\rangle \sqsubseteq A,\\ & a_i = \langle \textbf{IF}\,\phi_1, \textbf{THEN}\,\phi_2, \textbf{ELSE}\,\text{ABORT}\rangle,\\ & ?Ty(X) \in \phi_1,\\ \textbf{THEN} & \texttt{do}(\langle a_i, \ldots, a_{i+n}\rangle)\\ \textbf{ELSE} & \text{ABORT}\end{array}\right.$$

Simply stated, the rule of REGENERATION enables the parser (and generator) to take a sequence of actions from context and re-use them, provided that they were triggered by the same type-requirement as is imposed on the node currently under development.[28] Any such re-use of actions from context will be successful if and only if the result of applying these actions in the new context is suitable, i.e. yields an output in which all requirements are now satisfied, or which the actions of any immediately subsequent lexical expression can take as input to eventually lead to a complete tree.[29] This rule merely allows any sequence of actions to be re-used, given an appropriate type matching, without constraint on the end-point of the sequence.[30]

The second addition is a variant of SUBSTITUTION which, analogously, recovers actions rather than a formula value. Recall that, once having checked the appropriate pre-conditions for there to be some $Fo(Y)$ in a tree in context, the actions defined by SUBSTITUTION were simply to decorate the current node with the value of Y so found, in (6) this being $\texttt{put}(Fo(John'))$. Re-using this action in a new context will make no difference to the result – the formula $Fo(John')$ will be added. The problem posed by sloppy cases is that this is not what we want: we need pronouns to be resolved differently when their actions are re-run as part of an elliptical reconstruction process. We therefore define an alternative LOCAL-SUBSTITUTION rule, one that reflects the saving of actions by checking the modal tree relation between the current node and a putative antecedent:

LOCAL-SUBSTITUTION	**IF**	$Ty(X), ?\exists x.Fo(x),$	
		$\langle Y\rangle Ty(X), Fo(\alpha)$	
	THEN	**IF**	$\uparrow_0\uparrow_*^1\downarrow_0 Ty(X), Fo(\alpha)$
		THEN	ABORT
		ELSE	$\texttt{put}(Fo(\alpha))$
	ELSE	ABORT	

In this definition, X is a placeholder for a type as before, but Y is now a placeholder which ranges over the possible tree modalities $\{\uparrow, \downarrow, \ldots\}$: its value will be the modality describing the relative tree relation between the current node and the antecedent. As such modal relations can only hold between nodes in the same overall tree, this restricts this rule to antecedents along a tree-definable path. When the action is saved in context, X and Y must become fixed with the appropriate values (as with the semantic formula placeholder in SUBSTITUTION before). However, we now take α to be a rule-language metavariable which persists in context, rather than becoming fixed with the value it takes on first application (we will indicate this by use of Greek letters from now on). Re-application of this rule in a new context will therefore force the same type and relative tree address of the antecedent, but not the same semantic formula label.

With these formulations, we can provide a sloppy analysis of (6), repeated here:

(6) John worried about his sister, and Tom did too.

In the first sentence, given that *his* and its antecedent *John* are in the same tree, the parser can use the LOCAL-SUBSTITUTION rule to provide a value, instantiating the value for the modality $\langle Y\rangle$, as a path from the determiner-internal pronoun to the subject node; in this case, $\langle\uparrow_0\uparrow_0\uparrow_1\downarrow_0\rangle$. The relevant actions used are therefore as follows (slightly abbreviated):[31]

	IF	$?Ty(e \rightarrow t)$
	THEN	$\mathtt{make}(\langle\downarrow_1\rangle); \mathtt{go}(\langle\downarrow_1\rangle); \mathtt{put}(Ty(e \rightarrow (e \rightarrow t)));$
a_i (*worry about*)		$\mathtt{put}(Fo(Worry\text{-}about')); \mathtt{go}(\langle\uparrow_1\rangle);$
		$\mathtt{make}(\langle\downarrow_0\rangle); \mathtt{go}(\langle\downarrow_0\rangle); \mathtt{put}(?Ty(e))$
	ELSE	ABORT
	IF	$?Ty(e)$
	THEN	$\mathtt{make}(\langle\downarrow_1\rangle); \mathtt{go}(\langle\downarrow_1\rangle);$
		$\mathtt{put}(\lambda P.\epsilon, P); \mathtt{go}(\langle\uparrow_1\rangle);$
		$\mathtt{make}(\langle\downarrow_0\rangle\langle\downarrow_0\rangle); \mathtt{go}(\langle\downarrow_0\rangle\langle\downarrow_0\rangle); \mathtt{put}(Fo(\mathbf{U}));$
a_{i+1} (*his*)		$\mathtt{put}(?\exists x.Fo(x)); \mathtt{put}(Ty(e)); \mathtt{go}(\langle\uparrow_0\rangle);$
		$\mathtt{make}(\downarrow_1\downarrow_0); \mathtt{go}(\downarrow_1\downarrow_0); \mathtt{freshput}(\mathbf{x});$
		$\mathtt{go}(\langle\uparrow_0\rangle); \mathtt{make}(\langle\downarrow_1\rangle); \mathtt{go}(\langle\downarrow_1\rangle);$
	ELSE	ABORT
...	...	
	IF	$Ty(X = e), ?\exists x.Fo(x),$
		$\langle\uparrow_0\uparrow_0\uparrow_1\downarrow_0\rangle(Ty(e), Fo(\alpha))$
		(succeeds with $\alpha = John'$)
a_{i+j} (LOC-SUBST)	**THEN**	**IF** $\uparrow_0\uparrow_*^1\downarrow_0 Ty(e), Fo(\alpha)$
		THEN ABORT
		ELSE $\mathtt{put}(Fo(\alpha))$
	ELSE	ABORT

The action a_i for *worry about* introduces a predicate node $Worry\text{-}about'$, also introducing its argument node with a requirement for an object $Ty(e)$ term; the actions for *his* build an epsilon-term subtree, including a metavariable **U** which is required to be given a fixed formula; the LOCAL-SUBSTITUTION computational action resolved that metavariable. In this first application, the only available and consistent $Ty(e)$ antecedent node in the current partial tree is that for the subject, which is at $\langle\uparrow_0\uparrow_0\uparrow_1\downarrow_0\rangle$ from the current node; the formula $John'$ is therefore copied from there. The actions for *mother* (not shown) then provide the required predicate to complete the epsilon term.

The second (elliptical) sentence is initially parsed as before, with the lexical actions of *did* projecting a metavariable. Now, REGENERATION allows us to retrieve the actions shown above. Clearly, when these actions are re-applied in the new context of the elliptical utterance, an identical structure will be built modulo the fact that the LOCAL-SUBSTITUTION action will now pick up the local antecedent Tom' in the current partial tree, as this now decorates the node related to the current node by the modality $\langle\uparrow_0\uparrow_0\uparrow_1\downarrow_0\rangle$. We thus obtain the sloppy reading.[32]

Positing two substitution processes is, interestingly, not equivalent to a specification of lexical ambiguity in the pronominal or elliptical expressions themselves. Metavariables projected by pronouns are simply place-holders to be provided a value. However, just as in parsing, there is more than one strategy for providing such a value; tracing paths through nodes in a tree is one strategy provided the tree itself remains available in context.[33,34]

With VP ellipsis being a well-documented phenomenon, there is a regular array of data against which to check the proposed account. First there are the mixed readings; and here it is the locality restriction on LOCAL-SUBSTITUTION which ensures the correct results. In readings where the antecedent sentence contains a pronoun that is initially resolved without any reference to a tree relation between metavariable and antecedent, LOCAL-SUBSTITUTION will be inapplicable, and SUBSTITUTION must be used. This will cause the fixed resolved value to be saved in the action in context, with any subsequent elliptical sentence re-running these actions therefore correctly picking up the same fixed value:

(13) A: John is a waste of space.
B: But Tom let him share his ice-cream all the same.
A: Susan did too.

Here, we have a mixed reading available: the role played by *his* may be resolved sloppily (*A* may be saying that Susan let John share Susan's ice-cream) – but the value for *him* cannot change (*A* cannot be saying that Susan let Tom share anything). This is ensured by the fact that *B*'s utterance must use SUBSTITUTION to resolve *him* (as the antecedent is tree-external) but can use LOCAL-SUBSTITUTION to resolve *his*, a sequence of actions which A's second utterance replicates.

As the REGENERATION rule can apply to any type, an incidental bonus of this analysis is that we expect that pronouns could be resolved by re-using contextual actions. This gives us a handle on the pronouns originally identified as "sloppy" (Karttunen, 1976), where the value assigned to the pronoun must retain the fact that what its interpretation is based on contains a pronoun itself:

(14) John keeps the money he gets safely locked away, but Bill keeps it in a cupboard.

No special stipulation is needed to anticipate this effect: using REGENERATION to provide the value for the metavariable associated with *it* means re-running the actions originally used to parse the antecedent expression – these include introducing a new metavariable for *he* and resolving it via LOCAL-SUBSTITUTION, and repeating this will provide the desired values, this time resolving *he* to the new subject *Bill*.

There are, then, those cases which raise complications for the abstraction account (Dalrymple et al., 1991), because what appears to be doing the binding is a term contained within the subject, not the subject itself:

(15) A: A policeman who arrested Bill read him his rights.
B: The policeman who arrested Tom did too.

Here, our action re-use account provides the necessary reading without modification. The actions associated with *A*'s use of *read him his rights* in example (15) include the projection of metavariables associated with *him* and *his*, and their resolution along the path of nodes to that decorated by the term associated with *Bill*. Resolving the metavariable projected by the auxiliary *did* in *B*'s elliptical utterance via REGENERATION allows these resolutions by LOCAL-SUBSTITUTION to take a different, newly available antecedent Tom', providing the sloppy interpretation. The fact that this antecedent is provided by an expression inside a relative clause modifying the subject makes no difference, as the tree relation recorded in application of LOCAL-SUBSTITUTION is recoverable.

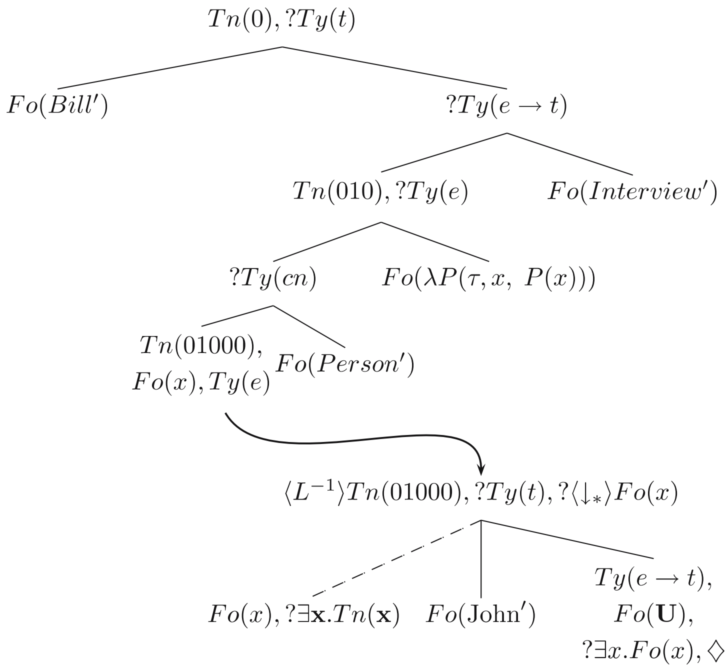
A further bonus for our analysis is that it provides a straightforward solution to the puzzle presented by antecedent-contained ellipsis:

(16) Bill interviewed everyone that John did.

This construction is problematic for many accounts, as the elliptical fragment is apparently contained within the expression from which its interpretation has to be built up, threatening circularity (Fiengo and May, 1994). It falls naturally into place, though, on an action re-use account. In DS, recall, adjunct structures for relative clause construal are constructed as paired *linked* structures. Such structures may be constructed in tandem, often with anaphoric-style links between them (Kempson et al., 2001; Kempson and Myer-Viol, 2002) with evaluation rules then determining that these independent structures, once completed, are compiled together to yield conjunctive propositional formulae in tree format. What the parsing of *everyone that John did* in example (16) provides is firstly a partial τ-term under construction got by parsing *everyone*,[35] with its fresh variable x; and secondly a linked tree structure with a copy of that variable x at an unfixed node. This independent linked tree will itself contain a subject node decorated by $Fo(John')$, and then a predicate node decorated by a $Ty(e \rightarrow t)$ metavariable projected by the elliptical *did* as usual.

In a case such as this, some constituents are already provided for the building up of a value for the predicate metavariable, so we can provide a complete fixed $Ty(e \rightarrow t)$ value by re-using only a single action from

(16′)



the first half of the parse process, viz. that associated with the single word *interviewed*; this constructs a new pair of a $Ty(e \rightarrow (e \rightarrow t))$ predicate node and a $?Ty(e)$ argument node, with which the already present unfixed node decorated with x can be MERGED. This leads to successful completion of the linked structure, thereby building up a composite restrictor for the τ-term already introduced. The final result is, as required:

$$S < x \; Fo(Interview'(\tau, x, (Person'(x) \wedge Interview'(x)(John')))(Bill'))$$

In fact, this analysis also applies to yield parallelism effects in scoping (Hirschbühler, 1982; Shieber et al., 1996). In (17), the subject can receive wide or narrow scope in *A*'s utterance, but must be given the same scope in *B*'s:

(17) A: A nurse interviewed every patient.
B: An orderly did too.

Although space precludes a full exposition here, as noted above, scope construal in DS relies on the evaluation of scope statements with initially underspecified arguments which, like pronouns, are encoded as metavariables and must be resolved via general computational actions. Using REGENERATION to resolve the elliptical *did* in *B*'s utterance will then re-use the actions used in interpreting *A*'s utterance, and the new subject will receive the same scope as was assigned previously.

5.2. FRAGMENT ELLIPSIS

This action re-use strategy applies equally to elliptical constructions other than VP ellipsis. With fragment construal, the term constructed appears to require insertion into arbitrary points of the replicated structure, as in (18), which allows interpretations with the fragment construed as either subject or object of *interview*:[36]

(18) A: John interviewed Mary in hospital.
B: Bill too.

To achieve this effect, we again appeal to the concept of re-using actions from context. First, the fragment *Bill* is parsed, decorating an unfixed node of type e (as with short answers, see above). A metavariable, this time of type t, is then projected by the lexical actions of *too*, whenever a complete propositional formula has not already been constructed:

$$
\textit{too}\quad\left|\begin{array}{lll}
\textbf{IF} & Ty(t), & \\
\textbf{THEN} & 1 & \\
\textbf{ELSE} & \textbf{IF} & ?Ty(t), \\
 & \textbf{THEN} & \texttt{put}(Ty(t)); \\
 & & \texttt{put}(Fo(\mathbf{U})); \\
 & & \texttt{put}(?\exists x.Fo(x)); \\
 & & \textsc{Regeneration}; \\
 & & \textbf{IF}\ Tn(0), Ty(t)\ \textbf{THEN}\ 1\ \textbf{ELSE}\ \text{ABORT} \\
 & \textbf{ELSE} & \text{ABORT}
\end{array}\right.
$$

In contrast to other similar anaphoric specifications, note that *too* enforces the completion of a propositional end result at the top node. This excludes the partial use of actions licensing strings after *too* has been parsed:[37]

(19) *I persuaded Harry to visit Mary in hospital; and Bill too Sue.

This metavariable now licenses the use of REGENERATION to re-run actions to construct a complete $Ty(t)$ formula. However, in order to allow the unfixed node to be merged into the appropriate position, we require an alternative version, allowing a compatible section of the contextual action sequence to be replaced by the standard MERGE operation:[38]

REGENERATION'	**IF**	$Ty(X), ?\exists x.Fo(x), \langle U, D\rangle(Ty(Y), ?Tn(Z))$
		$\langle T, W, A\rangle \in \mathcal{C}$,
		$\langle a_i, \ldots, a_j, \ldots a_k, \ldots, a_m\rangle \sqsubseteq A$,
		$a_i = \langle \textbf{IF}\,\phi_1, \textbf{THEN}\,\phi_2, \textbf{ELSE}\,\text{ABORT}\rangle$,
		$?Ty(X) \in \phi_1$,
		$a_j = \langle \textbf{IF}\,\theta_1, \textbf{THEN}\,\theta_2, \textbf{ELSE}\,\text{ABORT}\rangle$,
		$?Ty(Y) \in \theta_1$,
	THEN	$\texttt{do}(\langle a_i, \ldots, a_{j-1}\rangle)$,
		MERGE,
		$\texttt{do}(\langle a_k, \ldots, a_m\rangle)$,
	ELSE	ABORT

Here, we are re-running a sequence of actions associated with the construction of some propositional tree in the context up to a certain point. That point is reached when there is some action a_j triggered by a requirement $?Ty(X)$ for a decoration of the type $Ty(X)$ of the unfixed node. Instead of undertaking the action a_j, the unfixed node is merged at that point and then some consistent remaining subsequence of the actions $a_k, \ldots, a_m$ is run. Ultimately a propositional structure of $Ty(t)$ must be completed to remove the $?Ty(t)$ requirement; it will be identical to the original except for the decoration on the $Ty(X)$ node.

In (18), there are two possible points at which the sequence can be broken and the unfixed node merged in: either when the pointer is at the subject node or when it is at the object node. In the former case, it is the sequence of actions associated with parsing *John* that is replaced by the MERGE action; in the latter, those associated with parsing *Mary*.

At this juncture, there is an advantage in having presumed that the fact that VP ellipsis involves apparent binding of the subject position is a consequence of parsing the words in the fragment, and not a consequence of individuation of any particular argument node in the tracing of a path by LOCAL-SUBSTITUTION. For now, by REGENERATION and LOCAL-SUBSTITUTION, we can correctly anticipate the availability of sloppy readings from non-subject positions:

(20) I gave John comments on his paper, and Mary too.

In all these cases of ellipsis and anaphora construal, the trigger for contextual re-use is provided either by the projection of a metavariable by lexical action (whether by a pronoun, or an auxiliary) or by trigger for re-use of actions by a lexical item such as *too*); and the resolved value is provided simply by using whatever is already available in the context. While we have phrased most of our explanations in parsing terms, note that generation of these constructions is licensed in exactly the same way: projection of the metavariables leads to an update of the tree under construction which subsumes the intended goal tree, and the associated requirement forces some contextual enrichment to be used to fully specify the value. Literally, nothing more needs to be said. This result, as before, is a direct consequence of having defined generation in terms of the incremental actions used in parsing.

6. Alignment as Action Re-use – Minimizing Lexical Search

The previous three sections have shown how given a context that incorporates both actions and structure, we can extend our notion of grammatical well-formedness to include the anaphoric and elliptical phenomena so common in dialogue. This section briefly explores a corollary of this model: that besides this it provides a basis for explaining psycholinguistically-observed dialogue alignment preferences.

This assumption that context incorporates actions that can be re-used appears to have the major bonus of allowing a speaker to minimize what constitutes on our account the major task of production: the search through the lexicon for the appropriate lexical items in order. The generation process outlined in section 2.2, being word-by-word incremental, must search through the lexicon for potential lexical items *at each step*, checking for their suitability in extending the current partial tree structure so that it subsumes the goal tree.

We can see how reuse of context allows this task to be drastically reduced. At an ellipsis site, if there are actions in context that induce an appropriate parse tree, words won't have to be searched for, checked and produced: the goal can be achieved directly using context. As long as we can make an (intuitively reasonable) assumption that pronouns and elliptical auxiliaries are stored in a way that ensures their easy retrieval, their use therefore allows search to be bypassed.[39] This will lead to an obvious saving in cognitive processing costs, a well-known desideratum in cognitive accounts of processing (Sperber and Wilson, 1995, among others); and this perhaps gives us some insight into why pronouns and ellipsis are so common in dialogue.

Just as ellipsis and pronoun construal will provide a saving in processing, so too can the re-use of actions in context by constraining the generation process to search for words and their associated actions in the contextual parser state *before* looking in the lexicon. If suitable words and actions are found, the path of least effort will be to repeat words and structures already used.[40] And this, of course, is the alignment phenomenon itemized by Pickering and Garrod (2004) as part of the challenge which dialogue modelling poses.

Clearly, re-use of words and their associated actions in generation will surface as lexical alignment. More precisely, if there is some action $a \in A$

from some $\langle T, W, A\rangle \in C$ suitable for extending the current tree, a can be re-used, generating the word w which occupies the corresponding position in W. The result is repetition of w rather than choosing an alternative but as yet unused word from the lexicon. Note that as the parsing and generation processes work from the same context and use the same action definitions, previously parsed actions will directly cause repetition in generation without any transfer of representations being required.

Note also that as with REGENERATION before, re-use of the actions is importantly distinct from re-use of the trees they construct, and the terms that decorate them. Re-using the actions associated with an indefinite will introduce a new variable (as the action requires), rather than re-introducing the same term; and re-using pronoun actions will decorate a node with a new metavariable, rather than merely copying the previous resolved value. Re-use of indexicals such as *I* and *you* is therefore unproblematic, as their actions will require values to be assigned according to the current speaker and addressee, rather than copying values from previous uses.

Apparent alignment of *syntactic* structure also follows in virtue of the procedural action-based specification of lexical content. (Branigan et al., 2000) showed that syntactic structure tends to be preserved, with semantically equivalent double-object forms *give the cowboy a book* or full PP forms *give a book to the cowboy* being chosen depending on previous use. Most frameworks would have to reflect this via activation of syntactic rules, or perhaps preferences defined over parallelisms with syntactic trees in context, both of which seem problematic. In DS, though, this type of alternation is reflected not as a difference in the output of parsing (the semantic tree structure) but as a difference in the lexical actions used during parsing to build up this output: a word such as *give* has two possible lexical actions $give'$ and $give''$ corresponding to the two alternative forms (see Figure 3 below).

A previous use will cause either $give'$ or $give''$ to be present in A; re-use of this action will cause the same form to be repeated.[41] Similarly, (Cleand and Pickering, 2003) observed the repetition between participants of adjective structures as attributive or in a predicative relative-clause (*a green book* vs. *a book which is green*. These can be distinguished in DS by analyses that differ in the construction of a linked tree structure before the head noun (by lexical actions associated with attributive adjectives) or after the head (by actions associated with a relative pronoun); and re-use of these actions will cause repetition of form. So again two distinct tree-building strategies, despite producing the same logical form, nevertheless lead us to expect parallelism following the sequence of actions already in context.

The same approach can be applied for the parser, with contextual re-use of actions bypassing the need to test all possible actions associated in the lexicon with a particular word. A similar definition holds: for a word w presented

$$
give' \left| \begin{array}{ll}
\textbf{IF} & ?Ty(e \rightarrow t) \\
\textbf{THEN} & \text{(Double Object)} \\
 & \texttt{make}(\langle\downarrow_0\rangle); \texttt{go}(\langle\downarrow_0\rangle); \texttt{put}(?Ty(e)); \\
 & \texttt{go}(\langle\uparrow_0\rangle); \texttt{make}(\langle\downarrow_1\rangle); \texttt{go}(\langle\downarrow_1\rangle); \\
 & \texttt{put}(?Ty(e \rightarrow (e \rightarrow t))); \texttt{make}(\langle\downarrow_1\rangle); \texttt{go}(\langle\downarrow_1\rangle); \\
 & \texttt{put}(Ty(e \rightarrow (e \rightarrow (e \rightarrow t))), Fo(Give'), [\downarrow]\bot); \\
 & \texttt{go}(\langle\uparrow_1\rangle); \texttt{make}(\langle\downarrow_0\rangle); \texttt{go}(\langle\downarrow_0\rangle); \texttt{put}(?Ty(e)) \\
\textbf{ELSE} & \text{ABORT}
\end{array} \right.
$$

$$
give'' \left| \begin{array}{ll}
\textbf{IF} & ?Ty(e \rightarrow t) \\
\textbf{THEN} & \text{(Indirect Object)} \\
 & \texttt{make}(\langle\downarrow_1\rangle); \texttt{go}(\langle\downarrow_1\rangle); \texttt{put}(?Ty(e \rightarrow (e \rightarrow t))); \\
 & \texttt{make}(\langle\downarrow_1\rangle); \texttt{go}(\langle\downarrow_1\rangle); \\
 & \texttt{put}(Ty(e \rightarrow (e \rightarrow (e \rightarrow t))), Fo(Give'), [\downarrow]\bot); \\
 & \texttt{go}(\langle\uparrow_1\rangle); \texttt{make}(\langle\downarrow_0\rangle); \texttt{go}(\langle\downarrow_0\rangle); \texttt{put}(?Ty(e)) \\
 & \texttt{go}(\langle\uparrow_0\rangle\langle\uparrow_1\rangle); \texttt{make}(\langle\downarrow_0\rangle); \texttt{go}(\langle\downarrow_0\rangle); \texttt{put}(?Ty(e)) \\
\textbf{ELSE} & \text{ABORT}
\end{array} \right.
$$

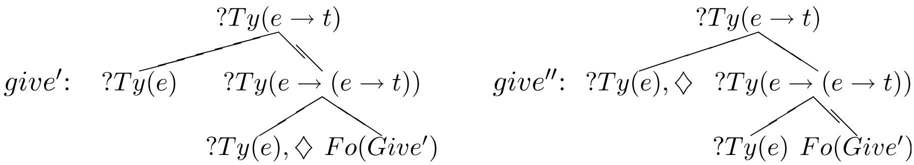


Figure 3. Output of alternative lexical actions for *give*.

as input, if $w \in W$ where $\langle T, W, A\rangle \in \mathcal{C}$ then the corresponding action a in A can be used without consulting the lexicon. Words will therefore tend to be interpreted as having the same sense or reference as before, modelling the *semantic* alignment described by (Garrod and Anderson, 1987).

These characterisations can also be extended to sequences of words – a sub-sequence $(a_1; a_2; \ldots ; a_n) \in A$ can be re-used by a generator, producing the corresponding word sequence $(w_1; w_2; \ldots ; w_n) \in W$; and similarly the sub-sequence of words $(w_1; w_2; \ldots ; w_n) \in W$ will cause the parser to use the corresponding action sequence $(a_1; a_2; \ldots ; a_n) \in A$. This will result in sequences or phrases being repeatedly associated by both parser and generator with the same sense or reference, leading to what Pickering and Garrod (2004) call *routinization* (construction and re-use of word sequences with consistent meanings).

It is notable that these various patterns of alignment, said by Pickering and Garrod (2004) to involve alignment across different levels, hence buttressing the existence of such distinct levels in the grammar (Jackendoff, 2002), are all expressible here as re-use of actions and sequences of actions. This result is achieved since context, content and lexical actions are all defined in terms of the same tree configurations. It is also notable that this analysis requires no higher-level hypotheses about the interlocutor. The parallelism across speakers seen in alignment might seem to many to necessitate high-level decisions to copy what has been done before (perhaps as a way of ensuring success in the communication), necessitating a considerable gap in complexity between what the grammar associates with some uttered string and the structures that speaker and hearer have to be taken to be manipulating.[42] However, in action re-use we have available a simpler and lower-level explanation: that an interlocutor will use whatever (s)he has in her own context to minimise the task at hand. The cross-feeding between parsing and production processes is ensured simply by defining them in terms of the same structures and structure-building actions.

7. A Grammar Formalism for Dialogue?

In this paper we have set out a grammar formalism which enables a particularly tight coupling of parsing and generation processes, both involving the same intrinsically dynamic structure-building formalism, and both building up on structures in context of the same formal type. This has made possible a model of dialogue that constitutes a direct response to the challenge set out by Pickering and Garrod (2004), capturing, as it does, shared utterances, ellipsis and alignment phenomena. A prototype system has been implemented in Prolog which reflects the model given here, demonstrating all the above phenomena in simple dialogue sequences.

One of the striking properties of this model is that the accounts of parsing and generation have not involved any articulation of higher-level propositional attitudes between speaker and hearer. Bringing out the broader significance of the role of higher-level reasoning about the other person's mental states in discourse is not our primary focus here, but the very fact that alignment, prevalent in dialogue, is naturally explained on the assumption that what is being checked as the context is merely the actions leading to the predicate-argument arrays representing context expressed by the utterance, strongly suggests that the processes involved in dialogue exchange do not of necessity involve higher-order hypotheses about the other person's mental states (contra both the proto-Gricean and relevance-theoretic accounts (Levinson, 2000; Sperber and Wilson, 1995, etc.)). Equally, the ease with which split utterance phenomena are reflected in this model turn on there being no such essential higher-order hypotheses. The consequences of this are far-reaching, since almost all accounts of language-understanding presume that such higher-level reflections are the bed-rock on which all communication is based (Sperber and Wilson, 1995; Levinson, 2000; Clark, 1996), though for isolated exceptions, see (Millikan, 2004; Breheny, 2005).

For grammar-formalism design also, the results are important. In so far as principles that determine such alignment effects also determine distributional patterns of individual languages and more broadly language in general, this suggests that formal models of language should be articulating a much more direct relationship between natural language expressions and the dynamics of their real-time use in context. One might ask now what significance there is to having defined a grammar formalism that provides a basis for dialogue-modelling. First, these principles provide no basis for any concept of sentence-meaning *sans* context, for the entire construction process has been made context-relative. What the grammar formalism provides is a set of processes which, in their implementation in any sequence of tree-growth updates, allow intercalation with processes such as substitution which are constrained by general cognitive principles, and so are essentially grammar-external. These processes are presumed to be available in the same way to both hearer and speaker in combination with whatever general cognitive principles that determine overall language use. In so far as this provides an articulation of those properties which are intrinsic to language, we have a grammar formalism of that language, albeit defined in an essentially procedural mode.

Secondly, on shifting the focus to dialogue, in so far as these principles also provide an articulation of dialogue processes, we do indeed have a grammar formalism directly applicable to dialogue modelling. There is nevertheless a significant shift of emphasis in these characterisations of language and dialogue; and this is because a full explanation of dialogue is transparently more than a full explanation of structural properties of a language, just as a full explanation of language use is transparently more than structural explanation of a single language. So, in articulating the system of principles which the model makes available, we have had nothing to say about the full complexities that determine language use, the process of how interlocutors decide what they want to say, or the rich array of ways in which they can indirectly interpret what they hear; and we have also had nothing to say about the process of how, in deciding to interrupt someone, an individual may decide that the sentence started by their interlocutor can be appropriately completed by themselves. The processes underpinning these phenomena doubtless involve higher order hypotheses about other people's mental states that make language the rich vehicle for communication that it is. These we take to be part of the remit of a theory of pragmatics which we do not provide (see e.g. (Sperber and Wilson, 1995)). Nonetheless what we are advocating is that the principles which underpin the explanation of intrinsic structural properties of language are the very same principles that underpin dialogue: both are explained in terms of how individuals process language in real time in the minimal context provided by their own previous processing. The human capacity for language, in this sense knowledge of language, is, on this view, the possession of a capacity that makes dialogue possible.

Acknowledgements

This paper is indebted to Wilfried Meyer-Viol for earlier work defining the formal underpinnings of the DS framework, though he should not be held to account for any of the views expressed here. The account provided also builds on previous work on generation with Masayuki Otsuka. Our thanks go to both, for their input in early stages of this work. We also thank Matthew Stone for suggesting to us the initial idea that the generation mechanism should directly adopt the parsing mechanism, rather than some inverse form of application. Thanks are also due to the reviewers of this journal, and to many others who have provided helpful comments. This work was supported by the ESRC (RES-000-22-0355) and (in the case of the third author) the Leverhulme Trust for a five-year research professorship.

Notes

[1] Example (1) is modelled after Branigan et al. (2000)s experiment in which participants describe pictures of simple situations to each other – see section 6.

[2] Example (2) is taken from (Clark and Wilkes-Gibbs, 1986), example (3) from the BNC, file KNY (sentences 315–317).

[3] Although see (Poesio and Rieser, 2003) for an initial DRT-based approach.

[4] See (Stanley, 2000; Ludlow, 2005), (Schlenker, 2003) who gives an account of pseudo-cleft constructions in terms of full question and full answer pairs, and (Carston, 2002) who sets aside ellipsis as falling outside a pragmatic account, on the basis of its syntactic characterisation.

[5] See (Stainton, 2005) for the same criticism of (Stanley, 2000) and (Ludlow, 2005), though Stainton only takes his arguments to apply to a restricted set of fragments.

[6] Until very recently, the parsing and production research communities have been totally disjoint, the former developing parsing systems compatible with some use-neutral grammar-formalism, the latter largely focussing on the utterance of words in isolation. See however (Cutler, 2002; Ferreira, 1996; Phillips, 2003).

[7] Fo is a predicate that takes a logical formula as value, Ty a predicate that takes logical types as values, Tn a predicate that takes tree-node addresses as values.

[8] We do not explore determination of scope dependencies in any depth in this paper – see (Kempson et al., 2001) chapter 7.

[9] From node n, $\langle\downarrow\rangle X$ denotes 'X holds at a daughter of n'; $\langle\downarrow_0\rangle X$ 'X holds at an argument daughter of n', $\langle\downarrow_1\rangle X$ 'X holds at a functor daughter of n', $\langle\uparrow\rangle X$ denotes 'X holds at the mother of n'.

[10] Though model-theoretic characterisations of anaphora construal have been predominant in the literature, there are also proof-theoretic accounts (Ranta, 1994; Fernando, 2002; Piwek, 1998), to which this account is allied.

[11] The specification of the metavariable as $\mathbf{U}_{Male'}$ here expresses a (presuppositional) constraint restricting potential substituends to the correct gender; see also footnote 12 below. Other constraints, e.g. restriction to finite clauses, we ignore here: see (Cann et al., 2005).

[12] In this, the system is like LFG, modelling long-distance dependency in the same terms as the LFG concept of *functional uncertainty* (Kaplan and Zaenen, 1989), differing from

that concept in the dynamics of update internal to the construction of a single tree, with relative clauses and other strong islands modelled as paired *linked* trees.

[13] In defining a model of generation to match the account of parsing, we only seek to model the mapping from some selected tree onto a linear string: in other words, a model of *tactical* generation (Dale, 1992), rather than *strategic* generation, the process of defining that selected tree in the first place.

[14] There are other modifications (in particular to the details of (Otsuka and Purver, 2003)'s proposed model of generation, to ensure strict incrementality and reflect advances in the parsing model), but as they do not concern the contextual dependence they will not be discussed here, and the reader may take these basic models as applicable without disadvantage.

[15] In cases where the semantic representation assigned to the antecedent may under-determine interpretation (as in the glue language characterisation of mixed quantification of (Crouch and van Genabith, 1999)), possible construals of the ellipsis site may be matched against that of the putative abstraction on the antecedent, with an evaluation metric which selects paired interpretations of antecedent and elliptical form that display the most parallelism.

[16] In the case that the sentence is unambiguous, or all ambiguity can be removed by inference etc., the final state P_1 will be a single triple $\langle T_1, W_1, A_1 \rangle$. We will sometimes simplify examples below in this way for ease of exposition, but it does not necessarily have to be the case.

[17] We ignore here gender specification, though we take this to be a condition on Action, not reflected in the update action.

[18] In requiring a variable for which a value is provided by the process of construal, this analysis might seem to be allied to the account of ellipsis as essentially involving reconstruction of syntactically explicit variables (Stanley, 2000). Indeed it does, with one critical difference, that what is reconstructed is not a linguistic expression, but a representation of its content.

[19] For simplicity, we ignore the contribution of the word *too* here, which we take to be associated with a test that a complete propositional value has been constructed (see below).

[20] Kempson et al. present detailed arguments that the supposed scope-taking properties of *wh*-question words do not provide evidence that *wh*-expressions are quantified expressions. Cann et al. (2005) suggest an alternative characterisation of *wh*-expressions as incomplete epsilon terms, lacking a scope statement, but here we retain the simpler characterisation.

[21] This is a feature-specific extension of a process of ANTICIPATION which moves the pointer down from a type-requiring node to a node needing further development (Cann et al., 2005).

[22] This mechanism is independently used to analyse other phenomena such as expletives (Cann et al., 2005).

[23] We have little to say about exactly *when* transitions occur (although presumably both speaker pauses and the availability to the hearer of a possible goal tree both play a part), or about how the hearer decides what the completing goal tree should be. We are interested here in characterizing the incremental parsing and resulting well-formedness of shared utterances. Just as we have nothing to say about the strategic generation of goal trees, but only about the tactical incremental generation of the corresponding partial tree and string, we have nothing to say here about the timing or inference methods involved in goal tree generation.

[24] This follows the informal outline of an analysis given by Otsuka and Purver (2003).

[25] Figure 2 contains several simplifications to aid readability (in particular ignoring the contribution of the auxiliary – see (Cann et al., 2005) for details), both to tree structure

details and by showing parser/generator states as single triples/pairs rather than sets thereof.

[26] Of course, if both *A* and *B* share the same lexical entries and communication is perfect, $P_t = P_t'$, but we do not have to assume that this is the case.

[27] There is a further effect associated with transition which we do not attempt to model here. Our analysis shows how the continuation can build a complete proposition or question (in (4) above, the question *'What did Alex design for herself?'*). However, many cases (especially those like (4) with interrogative continuations) can be seen not as asking this question directly, but as asking whether the previous speaker was asking it (*'Is it the question* ***'what did Alex design for herself'*** *that you are asking?'*). As we are not including a level of illocutionary force in our analysis, this distinction must fall outside our grammatical model; but see (Ginzburg et al., 2003) for a possible approach.

[28] We use the $\sqsubseteq$ symbol to represent the subsequence relation.

[29] As phrased here, this rule can only be triggered by the presence of a metavariable with an unsatisfied requirement for a fixed value (just as with SUBSTITUTION). It may, however, be that the rule should be generalised to any $?Ty(X)$ triggering context. This will allow gapping examples to be handled straightforwardly without the need for any extra machinery. Although this may seem to open the floodgates for arbitrary generation of semantic structure, the use of such a procedure will be constrained by both context and prosody. We will leave this possibility aside for now.

[30] This characterisation will also apply directly to pseudo-gapping, as the actions of parsing the verb *interviewed* in the first conjunct in (i) can be used to extend the tree to provide the appropriate structure for parsing the final noun phrase *Mandela*:
(i) John interviewed Clinton, and I did Mandela.

[31] We take *his* to project the combination of an epsilon operator introducing a two place predicate, of whose arguments, the higher is a metavariable of type *e*, the lower, the variable that the operator binds. The characterisation of *his* simplistically conflates the pronominal sub-entry and the sub-entry for the genitive, a conflation which leaves the pointer at not strictly the right node. In a stricter specification, in which the contained noun-phrase expression is analysed as parsed as a node locally unfixed to the $Ty(e)$-requiring node, subsequently resolved to yield a structure identical to that of *mother of him*, this problem doesn't arise. But we leave such complications aside here. See (Cann et al., 2005) for justification of the construction of locally unfixed nodes.

[32] This characterisation forces an exactly parallel tree relation between the regenerated pronoun and its antecedent in the two sentences. This could be weakened to allow for non-identical structure, but we take any lack of parallelism to seriously jeopardise the availability of sloppy readings, as in (i), and therefore we do not make that move here:
(i) The teacher who spoke to Bill about his problems reported them to the head, and the man who Sue tells me had spoken to Tom did too. Note also that versions with subject-auxiliary inversion *"and so did Tom"* require the lexical actions of the auxiliary to provide an underspecified subject node; we skip this refinement here, but see (Cann et al., 2005).

[33] One might view the distinction between the two forms of SUBSTITUTION as the parser's pragmatic choice of whether to take the (strict) choice of substituend as critical, or the (sloppy) local node relation (an option not even available if the substituend is taken from a tree which is not part of the current tree, as no such relation can be defined). We take the two separate rule specifications as simply making this choice explicit.

[34] We assume, without full analysis, that presuppositional constraints on pronoun resolution (gender and speaker/addressee identity) are taken not to persist into context, in order to allow the LOCAL-SUBSTITUTION actions for resolving *his* and *my* in (i)–(ii) below to be

re-run, picking up on the different parallel antecedents even though they do not fit the initial gender or speaker-identity requirements:
(i) A: John left his socks in the washing machine. B: Susan did too.
(ii) A: I left my socks in the washing machine. B: I did too.
[35] $(\tau, x, ...)$ is the epsilon calculus analogue to the universal quantifier.
[36] This process has been dubbed "stripping". See (Reinhart, 1991; Kempson, 1995) for discussion of its strong-island sensitivity.
[37] This characterisation also correctly precludes the use of *too* in pseudo-gapping:
(i) John interviewed Clinton, and Bill did too Mandela.
[38] Use of this strategy is not restricted to *too*; we take it to be a generally available alternative, licensing examples such as:
(ii) Sue persuaded Harry to visit his mother in hospital, and Mary did Tom.
[39] In our implementation, such anaphoric lexical items are simply considered part of the discourse-initial context; other approaches would be possible.
[40] Our assumption of general lexicon search is naive, but even assuming a more efficient strategy (e.g. by activating only certain subfields of the lexicon based on the semantic formulae and structure of the goal tree) searching through the immediate context will still minimise the effort required.
[41] Branigan et al. (2000) also showed that this effect also occurs in a weaker form when the second sentence contains a lexically distinct verb with the same alternation in form (e.g. *hand, pas*). In this the actions might still be being presumed to be recovered from the context, with the lexicon-search task being reduced to finding some predicate complementary to $Give'$, $Hand'$ $Pass'$, etc.
[42] This is the well-known problem of mutual knowledge of proto-Griceans (Smith, 1982, and other references subsequently), a problem also facing relevance-theorists with the analogous concept of mutual manifestness (Sperber and Wilson, 1995).

References

Aoshima S., Phillips C., Weinberg A. (2004) Processing Filler-Gap Dependencies in a Head-Final Language. *Journal of Memory and Language*, 51, pp. 23–54.

Blackburn P., Meyer-Viol W. (1994) Linguistics, Logic and Finite Trees. *Bulletin of the IGPL*, 2, pp. 3–31.

Branigan H., Pickering, M., Cleland A. (2000) Syntactic Co-ordination in Dialogue. *Cognition*, 75, pp. 13–25.

Breheny R. (2005) Communication and Folk Psychology. *Mind and Language*. to appear.

Cann R. (2005) Semantic Underspecification and the Interpretation of Copular Clauses in English. In von Heusinger K., Turner K. (eds.), *Where Semantics Meets Pragmatics*. Elsevier, Oxford. to appear.

Cann R., Kempson R., Marten L. (2005) *The Dynamics of Language*, No. 35 in Syntax and Semantics. Elsevier, Oxford.

Carston R. (2002) *Thoughts and Utterances*. Blackwell, Oxford.

Chomsky N. (1981) *Lectures on Government and Binding*. Foris, Dordrecht.

Clark H. H. (1996) *Using Language*. Cambridge University Press.

Clark H. H., Wilkes-Gibbs D. (1986) Referring as a Collaborative Process. *Cognition*, 22, pp. 1–39.

Cleland A., Pickering M. (2003) The Use of Lexical and Syntactic Information in Language Production: Evidence from the Priming of Noun Phrases. *Journal of Memory and Language*, 49, pp. 214–230.

Crocker M., Pickering M., Clifton C. (eds.), (2000) *Architectures and Mechanisms in Sentence Comprehension*. Cambridge University Press.

Crouch R., van Genabith J. (1999) Context Change, Underspecification and the Structure of Glue Language Derivations. In Dalrymple M. (ed.), *Semantics and Syntax in Lexical Functional Grammar. The Resource Logic Approach*. MIT Press.

Cutler A. (2002) Phonological Processing. In Gussenhoven C., Warner N. (eds.), *Papers in Laboratory Phonology VII*. Mouton De Gruyter, Berlin. pp. 275–296.

Dale R. (1992) *Generating Referring Expressions: Constructing Descriptions in a Domain of Objects and Processes*. MIT Press.

Dalrymple M., Shieber S.M., Pereira F.C.N. (1991) Ellipsis and Higher-Order Unification. *Linguistics and Philosophy*, 14/4, pp. 399–452.

Fernando T. (2002) Three processes in natural language interpretation. In Sieg W., Sommer R., Talcott C. (eds.), *Reflections on the Foundations of Mathematics: Essays in Honor of Solomon Feferman*. Mass: Association for Symbolic Logic, Natick, pp. 208–227.

Ferreira V. (1996) Is it Better to Give than to Donate? Syntactic Flexibility in Language Production. *Journal of Memory and Language*, 35, pp. 724–755.

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

Fletcher C. (1994), Levels of Representation in Memory for Discourse. In Gernsbacher M. (ed.), *Handbook of Psycholinguistics*. Academic Press, pp. 589–607.

Garrod S., Anderson A. (1987) Saying What You Mean in Dialogue: A Study in Conceptual and Semantic Co-ordination. *Cognition*, 27, pp. 181–218.

Garrod S., Doherty G. (1994) Conversation, Co-ordination and Convention: an Empirical Investigation of How Groups Establish Linguistic Conventions. *Cognition*, 53, pp. 181–215.

Ginzburg J.: forthcoming, *A Semantics for Interaction in Dialogue*. CSLI Publications. Draft chapters available from: `http://www.dcs.kcl.ac.uk/staff/ginzburg`.

Ginzburg J., Cooper R. (2004) Clarification, Ellipsis, and the Nature of Contextual Updates in Dialogue. *Linguistics and Philosophy*, 27/3, pp. 297–365.

Ginzburg J., Sag I. (2000) *Interrogative Investigations: the Form, Meaning and Use of English Interrogatives*, No. 123 in CSLI Lecture Notes. CSLI Publications.

Ginzburg J., Sag I., Purver M. (2003) Integrating Conversational Move Types in the Grammar of Conversation. In Kühnlein P., Rieser H., Zeevat H. (eds.), *Perspectives on Dialogue in the New Millennium*, Vol. 114 of *Pragmatics and Beyond New Series*. John Benjamins, pp. 25–42.

Groenendijk J., Stokhof M. (1984), On the Semantics of Questions and the Pragmatics of Answers. In Landman F., Veltman F. (eds.), *Varieties of Formal Semantics*, Vol. 3 of *Groningen-Amsterdam Studies in Semantics (GRASS)*. Foris, pp. 143–170.

Heim I. (1982) *The Semantics of Definite and Indefinite Noun Phrases*. Ph.D. thesis, University of Massachusetts at Amherst.

Higginbotham J., May R. (1981) Questions, Quantifiers and Crossing. *Linguistic Review*, 1, pp. 41–79.

Hirschbühler P. (1982) VP Deletion and Across-the-Board Quantifier Scope. In Pustejovsky J., Sells P. (eds.), *Proceedings of NELS 12*.

Jackendoff R. (2002) *Foundations of Language: Brain, Meaning, Grammar, Evolution*. Oxford University Press, Oxford.

Joshi A., Kulick S. (1997) Partial Proof Trees as Building Blocks for a Categorial Grammar. *Linguistics and Philosophy*, 20, pp. 637–667.

Kamp H., Reyle U. (1993) *From Discourse To Logic*. Kluwer Academic Publishers.

Kaplan R., Zaenen A. (1989) Long-distance Dependencies, Constituent Structure, and Functional Uncertainty. In Baltin M., Kroch A. (eds.), *Alternative Conceptions of Phrase Structure*. University of Chicago Press, Chicago. pp. 17–42.

Karttunen L. (1976) Discourse Referents. In McCawley J. (ed.), *Syntax and Semantics*, Vol. 7. Academic Press, New York. pp. 363–385.

Kempson R. (1995) Ellipsis as Labelled Deduction. *Bulletin of the Interest Group in Pure and Applied Logics*, 3 2,3, pp. 489–525.

Kempson R., Meyer-Viol W. (2002) The Dynamics of Syntax: Relative Pronouns, Anaphora and Crossover. In Weise H., Simon H. (eds.), *Pronouns: Grammar and Representation*. John Benjamins, Amsterdam. pp. 137–160.

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

Kempson R., Meyer-Viol W., Otsuka M. (2003) Growth of Logical Form: the Dynamics of Syntax. In Peregrin J. (ed.), *Meaning: The Dynamic Turn*. Elsevier, Oxford.

Levinson S. (2000) *Presumptive Meanings*. MA: MIT Press, Cambridge.

Ludlow P. (2005) A Note on Alleged Cases of Nonsentential Assertion. In Elugardo R., Stainton R. (eds.), *Nonsentential Speech*. Kluwer Academic Publishers.

Marcus M. (1987) Deterministic Parsing and Description Theory. In Whitelock P., Wood M., Somers H., Johnson R., Bennett P. (eds.), *Linguistic Theory and Computer Applications*. Academic Press, New York. pp. 69–112.

Marten L. (2002) *At the Syntax-Pragmatics Interface: Verbal Underspecification and Concept Formation in Dynamic Syntax*. Oxford University Press, Oxford.

Merchant J. (2001) *The Syntax of Silence*. Oxford University Press, Oxford.

Meyer-Viol W. (1995) *Instantial Logic*. Ph.D. thesis, University of Utrecht.

Millikan R. (2004) *The Varieties of Meaning: The Jean-Nicod Lectures*. MIT Press.

Neumann G. (1998) Interleaving Natural Language Parsing and Generation Through Uniform Processing. *Artificial Intelligence*, 99, pp. 121–163.

Otsuka M., Purver M. (2003) Incremental Generation by Incremental Parsing. In *Proceedings of the 6th CLUK Colloquium*. Edinburgh, pp. 93–100, CLUK.

Phillips C. (2003) Linear Order and Constituency. *Linguistic Inquiry*, 34, pp. 37–90.

Pickering M., Garrod S. (2004) Toward a Mechanistic Psychology of Dialogue. *Behavioral and Brain Sciences*, 27, pp. 169–226.

Piwek P. (1998) *Logic, Information and Conversation*. Ph.D. thesis, Tilburg University.

Poesio M., Rieser H. (2003) Coordination in a PTT Approach to Dialogue. In *Proceedings of the 7th Workshop on the Semantics and Pragmatics of Dialogue (DiaBruck)*. Saarbrücken.

Pullum G., Scholz B. (2001) On the Distinction between Model-Theoretic and Generative-Enumerative Syntactic Frameworks. In Morrill G., Le Groote P., Retore C. (eds.), *Logical Aspects of Computational Linguistics: 4th International Conference*. pp. 17–43, Springer.

Purver M., Otsuka M. (2003) Incremental Generation by Incremental Parsing: Tactical Generation in Dynamic Syntax. In *Proceedings of the 9th European Workshop in Natural Language Generation (ENLG-2003)*. Budapest, pp. 79–86, Association for Computational Linguistics.

Ranta A. (1994) *Type-Theoretical Grammar*. Oxford University Press.

Reinhart T. (1991) Elliptic Conjunctions – Non-Quantificational LF. In Kasher A. (ed.), *The Chomskian Turn*. Blackwell, pp. 360–384.

Schlenker P. (2003) Clausal Equations: a Note on the Connectivity Problem. *Natural Language and Linguistic Theory*, 21, pp. 157–214.

Shieber S., Pereira F., Dalrymple M. (1996) Interactions of Scope and Ellipsis. *Linguistics and Philosophy*, 19, pp. 527–552.

Smith N. (ed.), (1982) *Mutual Knowledge*. Blackwell, Oxford.

Sperber D., Wilson D. (1995) *Relevance: Communication and Cognition*. Blackwell, second edition.

Stainton R. (2005) *Words and Thoughts*. Oxford University Press, Oxford. to appear.

Stalnaker R. (1978) Assertion. In Cole P. (ed.), *Syntax and Semantics 9: Pragmatics*. Academic Press, New York. pp. 315–332.

Stanley J. (2000) Context and Logical Form. *Linguistics and Philosophy*, 23, pp. 391–434.

Stone M. (2003) Specifying Generation of Referring Expressions by Example. In *Proceedings of the AAAI Spring Symposium on Natural Language Generation in Spoken and Written Dialogue*. Stanford, pp. 133–140.

Stone M., Doran C. (1997) Sentence Planning as Description Using Tree-Adjoining Grammar. In Cohen P., Wahlster W. (eds.), *Proceedings of the 35th Annual Meeting of the ACL*. pp. 198–205.

van Deemter K., Odijk J. (1997) Context Modeling and the Generation of Spoken Discourse. *Speech Communication*, 21, pp. 101–121.