Incremental Parsing, or Incremental Grammar?[*]

Matthew Purver[†] and **Ruth Kempson**[‡]
Departments of [†]Computer Science and [‡]Philosophy,
King's College London, Strand,
London WC2R 2LS,
UK
{matthew.purver, ruth.kempson}@kcl.ac.uk

Abstract

Standard grammar formalisms are defined without reflection of the incremental and serial nature of language processing, and incrementality must therefore be reflected by independently defined parsing and/or generation techniques. We argue that this leads to a poor setup for modelling dialogue, with its rich speaker-hearer interaction, and instead propose context-based parsing and generation models defined in terms of an inherently incremental grammar formalism (Dynamic Syntax), which allow a straightforward model of otherwise problematic dialogue phenomena such as shared utterances, ellipsis and alignment.

1 Introduction

Despite increasing psycholinguistic evidence of incrementality in language processing, both in parsing (see e.g. (Crocker et al., 2000)) and in production (Ferreira, 1996), there is almost universal agreement that this should not be reflected in grammar formalisms which constitute the underlying model of language (for rare exceptions, see (Hausser, 1989; Kempson et al., 2001)). Constraint-based grammar formalisms are accordingly defined neutrally between either of these applications, with parsing/generation systems (whether incremental or not) defined as independent architectures manipulating the same underlying system.[1] Such assumptions however lead to formal architectures that are relatively poorly set up for modelling dialogue, for they provide no basis for the very rich degree of interaction between participants in dialogue. A common phenomenon in dialogue is that of *shared utterances* (Clark and Wilkes-Gibbs, 1986), with exchange of roles of parser and producer midway through an utterance:[2]

(1)
Daniel:	Why don't you stop mumbling and
Marc:	Speak proper like?
Daniel:	speak proper?

(2)
Ruth:	What did Alex ...
Hugh:	Design? A kaleidoscope.

Such utterances clearly show the need for a strictly incremental model: however, they are particularly problematic for any overall architecture in which parsing and generation are independently defined as applications of a use-neutral grammar formalism which yields as output the set of well-formed strings, for in these types of exchange, the string uttered by one and parsed by the other need not be a wellformed string in its own right, so will not fall within the set of data which the underlying formalism is set up to capture. Yet, with the transition between interlocutors seen as a shift from one system to another, each such substring will have to be characterised independently. Many other dialogue phenomena also show the need for interaction between the parsing and generation processes, among them cross-speaker ellipsis (e.g. simple bare fragment answers to *wh*-questions), and *alignment* (Pickering and Garrod, 2004), in which conversational participants mirror each other's patterns at many levels (including lexical choice and syntactic structure).

The challenge of being able to model these phenomena, problematic for theorists but extremely easy and natural for dialogue participants themselves, has recently been put out by Pickering and Garrod (2004) as a means of evaluating both putative grammar formalisms and

---

[1] Authors vary as to the extent to which these architectures might be defined to be reversible. See (Neumann, 1994).

[2] Example (1) from the BNC, file KNY (sentences 315–317).

models of language use. In response to this challenge, we suggest that an alternative means of evaluating parsing implementations is by evaluation of paired parsing-generation models and the dialogue model that results. As an illustration of this, we show how if we drop the assumption that grammar formalisms are defined neutrally between parsing and production in favour of frameworks in which the serial nature of language processing is a central design feature (as in Dynamic Syntax: (Kempson et al., 2001)), then we can define a model in which incremental sub-systems of parsing and generation are necessarily tightly coordinated, and can thus provide a computational model of dialogue which directly corresponds with currently emerging psycholinguistic results (Branigan et al., 2000). In particular, by adding a shared model of context to previously defined word-by-word incremental parsing and generation models, we show how the switch in speaker/hearer roles during a shared utterance can be seen as a switch between processes which are directed by different goals, but which share the same incrementally built data structures. We then show how this inherent incrementality and structure/context sharing also allows a straightforward model of cross-speaker ellipsis and alignment.

2 Background

Dynamic Syntax (DS) (Kempson et al., 2001) 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. In DS, grammaticality is defined as parsability (the successful incremental construction of a tree-structure logical form, 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 (see (Meyer-Viol, 1995) for a recent development), 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. With all quantification expressed as type e terms, the standard grounds for mismatch between syntactic and semantic

analysis for all NPs is removed; and, indeed, all syntactic distributions are explained in terms of this incremental and monotonic growth of partial representations of content. Hence the claim that the model itself constitutes a NL grammar formalism.

Parsing (Kempson et al., 2001) defines parsing as a process of building labelled semantic trees in a strictly left-to-right, word-by-word incremental fashion by using computational actions and lexical actions defined (for some natural language) using the modal tree logic LOFT (Blackburn and Meyer-Viol, 1994). These actions are defined as transition functions between intermediate states, which monotonically extend tree structures and node decorations. Words are specified in the lexicon to have associated lexical actions: the (possibly *partial*) semantic trees are monotonically extended by applying these actions as each word is consumed from the input string. Partial trees may be underspecified: tree node relations may be only partially specified; node decorations may be defined in terms of unfulfilled requirements and metavariables; and trees may lack a full set of scope constraints. Anaphora resolution is a familiar case of update: pronouns are defined to project metavariables that are substituted from context as part of the construction process. Relative to the same tree-growth dynamics, long-distance dependency effects are characterised through restricted licensing of partial trees with relation between nodes introduced with merely a constraint on some fixed extension (following D-Tree grammar formalisms (Marcus, 1987)), an underspecification that gets resolved within the left-to-right construction process.[3] Quantifying terms are also built up using determiner and noun to yield a partially specified term e.g. $(\epsilon, y, Man'(y))$ with a requirement for a scope statement. These scope statements, of the form $x < y$ ('the term binding x is to be evaluated as taking scope over the term binding y'), are added to a locally dominating type-t-requiring node. Generally, they are added to an accumulating set following the serial order of processing in determining the scope dependency, but indefinites (freer in scope potential) are assigned a metavariable as first argument, allow-

---

[3]In this, the system is also like LFG, modelling long-distance dependency in the same terms as *functional uncertainty* (Kaplan and Zaenen, 1989), differing from that concept in the dynamics of update internal to the construction of a single tree.

Figure 1: **Parsing *"john likes mary"* ...**　　　**... and generating *"john likes mary"***
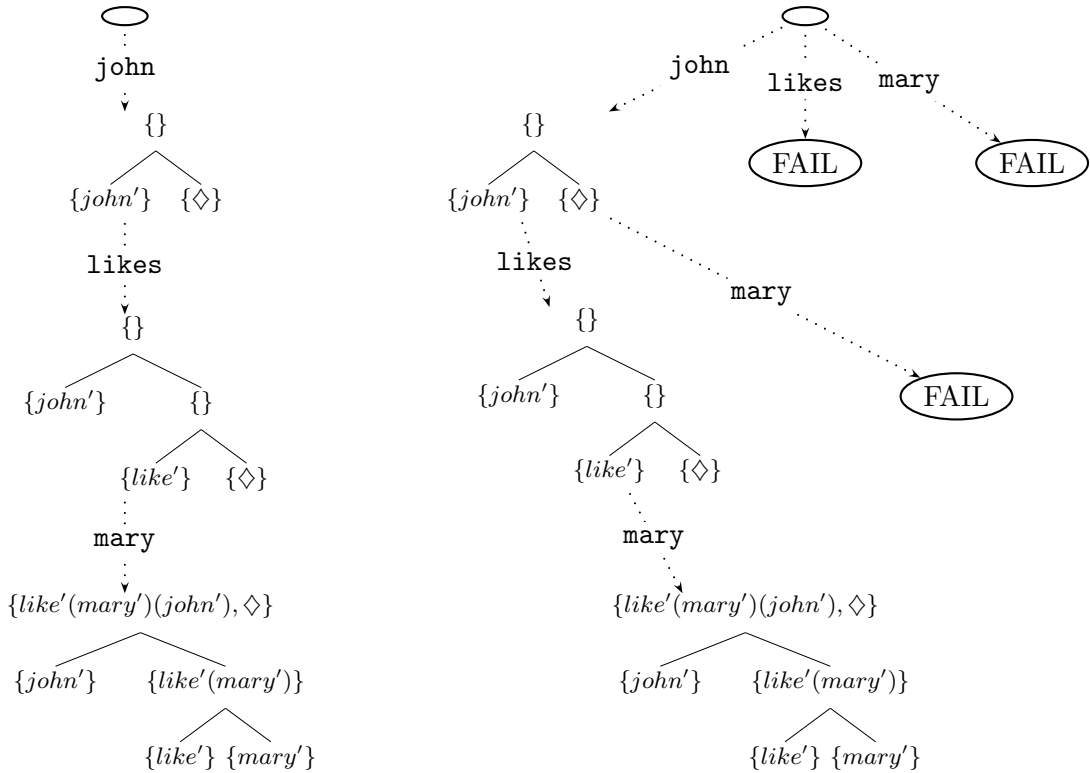
ing selection from any term already added, including temporally-sorted variables associated with tense/modality specifications. The general mechanism is the incremental analogue of quantifier storage; and once a propositional formula of type t has been derived at a node with some collection of scope statements, these are jointly evaluated to yield fully expanded terms that reflect all relative dependencies within the restrictor of the terms themselves. For example, parsing *A man coughed* yields the pair $S_i < x$, $Cough'(\epsilon, x, Man'(x))$ (S_i the index of evaluation), then evaluated as $Man'(a) \wedge Cough'(a)$ where $a = (\epsilon, x, Man'(x) \wedge Cough'(x))$.[4]

Once all requirements are satisfied and all partiality and underspecification is resolved, trees are *complete*, parsing is successful and the input string is said to be grammatical. Central to the formalism is the incremental and monotonic growth of labelled partial trees: 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.[5]

---

[4]For formal details of this approach to quantification see (Kempson et al., 2001) chapter 7; for an early implementation see (Kibble et al., 2001).

[5]Figure 1 assumes, simplistically, that linguistic names correspond directly to scopeless names in the log-

Generation (Otsuka and Purver, 2003; Purver and Otsuka, 2003) (hereafter O&P) give an initial method of context-independent tactical generation based on the same incremental parsing process, 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 produce trees which subsume the goal tree are kept, and the process succeeds when a complete tree identical to the goal tree is produced. Generation and parsing thus use the same tree representations and tree-building actions throughout.

3　Contextual Model

The current proposed model (and its implementation) is based on these earlier definitions but modifies them in several ways, most significantly by the addition of a model of context:

---

ical language that decorate the tree.

while they assume some notion of context they give no formal model or implementation.[6] The contextual model we now assume is made up not only of the semantic trees built by the DS parsing process, but also the sequences of words and associated lexical actions that have been used to build them. It is the presence of (and associations between) all three, together with the fact that this context is equally available to both parsing and generation processes, that allow our straightforward model of dialogue phenomena.[7] For the purposes of the current implementation, we make a simplifying assumption that the length of context is finite and limited to the result of some immediately previous parse (although information that is independently available can be represented in the DS tree format, so that, in reality, larger and only partially ordered contexts are no doubt possible): context at any point is therefore made up of the trees and word/action sequences obtained in parsing the previous sentence and the current (incomplete) sentence.

Parsing in Context   A parser state is therefore defined to be a set of triples $\langle T, W, A \rangle$, where T is a (possibly partial) semantic tree,[8] W the sequence of words and A the sequence of lexical and computational actions that have been used in building it. This set will initially contain only a single triple $\langle T_a, \emptyset, \emptyset \rangle$ (where T_a is the basic axiom taken as the starting point of the parser, and the word and action sequences are empty), but will expand as words are consumed from the input string and the corresponding actions produce multiple possible partial trees. At any point in the parsing process, the context for a particular partial tree T in

this set can then be taken to consist of: (a) a similar triple $\langle T_0, W_0, A_0 \rangle$ given by the previous sentence, where T_0 is its semantic tree representation, W_0 and A_0 the sequences of words and actions that were used in building it; and (b) the triple $\langle T, W, A \rangle$ itself. Once parsing is complete, the final parser state, a set of triples, will form the new starting context for the next sentence. In the simple case where the sentence is unambiguous (or all ambiguity has been removed) this set will again have been reduced to a single triple $\langle T_1, W_1, A_1 \rangle$, corresponding to the final interpretation of the string T_1 with its sequence of words W_1 and actions A_1, and this replaces $\langle T_0, W_0, A_0 \rangle$ as the new context; in the presence of persistent ambiguity there will simply be more than one triple in the new context.[9]

Generation in Context   A generator state is now defined as 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). Initially, the set X will usually contain only one pair, of an empty candidate string and the standard initial parser state, $(\emptyset, \{\langle T_a, \emptyset, \emptyset \rangle\})$. However, as both parsing and generation processes are strictly incremental, they can in theory start from *any* state. The context for any partial tree T is defined exactly as for parsing: the previous sentence triple $\langle T_0, W_0, A_0 \rangle$; and the current triple $\langle T, W, A \rangle$. Generation and parsing are thus very closely coupled, with the central part of both processes being a parser state: a set of tree/word-sequence/action-sequence triples. Essential to this correspondence is the lack of construction of higher-level hypotheses about the state of the interlocutor. All transitions are defined over the context for the individual (parser or generator). In principle, contexts could be extended to include high-level hypotheses, but these are not essential and are not implemented in our model (see (Millikan, 2004) for justification of this stance).

4   Shared Utterances

One primary evidence for this close coupling and sharing of structures and context is the ease with which shared utterances can be expressed. O&P suggest an analysis of shared utterances,

---

[6]There are other departures in the treatment of linked structures (for relatives and other modifiers) and quantification, and more relevantly to improve the incrementality of the generation process: we do not adopt the proposal of O&P to speed up generation by use of a restricted multiset of lexical entries selected on the basis of goal tree features, which prevents strictly incremental generation and excludes modification of the goal tree.

[7]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 (see e.g. (Stone and Doran, 1997)), both parsing and generation are not head-driven, but fully (word-by-word) incremental. This has the advantage of allowing fully incremental models for all languages, matching psycholinguistic observations (Ferreira, 1996).

[8]Strictly speaking, scope statements should be included in these n-tuples – for now we consider them as part of the tree.

[9]The current implementation of the formalism does not include any disambiguation mechanism. We simply assume that selection of some (minimal) context and attendant removal of any remaining ambiguity is possible by inference.

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

$$P_t = \left\langle \begin{array}{c} \{+Q\} \\ \{WH\} \quad \{alex'\}\{?Ty(e \to t), \Diamond\} \end{array}, \{\texttt{what}, \texttt{did}, \texttt{alex}\}, \{a_1, a_2, a_3\} \right\rangle$$

$$G_t = \left(\begin{array}{c} \{+Q, design'(WH)(alex')\} \\ \{alex'\} \quad \{design(WH)\} \\ \{WH\}\{design'\} \end{array}, \left(\emptyset, \left\langle \begin{array}{c} \{+Q\} \\ \{WH\} \quad \{alex'\}\{?Ty(e \to t), \Diamond\} \end{array}, \{\texttt{what}, \texttt{did}, \texttt{alex}\}, \{a_1, a_2, a_3\} \right\rangle \right) \right)$$

$$G_1 = \left(\begin{array}{c} \{+Q, design'(WH)(alex')\} \\ \{alex'\} \quad \{design'(WH)\} \\ \{WH\}\{design'\} \end{array}, \left(\{\texttt{design}\}, \left\langle \begin{array}{c} \{+Q\} \\ \{WH\}\{alex'\} \quad \{?Ty(e \to t)\} \\ \{\Diamond\}\{design'\} \end{array}, \{\ldots, \texttt{design}\}, \{\ldots, a_4\} \right\rangle \right) \right)$$

and this can now be formalised given the current model. As the parsing and generation processes are both fully incremental, they can start from any state (not just the basic axiom state $\langle T_a, \emptyset, \emptyset \rangle$). As they share the same lexical entries, the same context and the same semantic tree representations, a model of the switch of roles now becomes relatively straightforward.

Transition from Hearer to Speaker Normally, the generation process begins with the initial generator state as defined above: $(T_g, \{(\emptyset, P_0)\})$, where P_0 is the standard initial "empty" parser state $\{\langle T_a, \emptyset, \emptyset \rangle\}$. 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,[10] 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.[11] Note that the context does not change between processes.

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. the proposition to be expressed must be subsumed by the incomplete proposition that has been 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 propositional structure is available.

Transition from Speaker to Hearer At 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 as far as B is concerned.[12] 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 O&P.

It is the incrementality of the underlying grammar formalism that allows this simple switch: the parsing process can begin directly

---

[10]We have little to say about exactly *when* transitions occur. Presumably speaker pauses and the availability to the hearer of a possible goal tree both play a part.

[11]Figure 2 contains several simplifications to aid readability, both to tree structure details and by showing parser/generator states as single triples/pairs rather than sets thereof.

[12]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.

from a state produced by an incomplete generation process, and vice versa, as their intermediate representations are *necessarily* the same.

5   Cross-Speaker Ellipsis

This inherent close coupling of the two incremental processes, together with the inclusion of tree-building actions in the model of context, also allows a simple analysis of many cross-speaker elliptical phenomena.
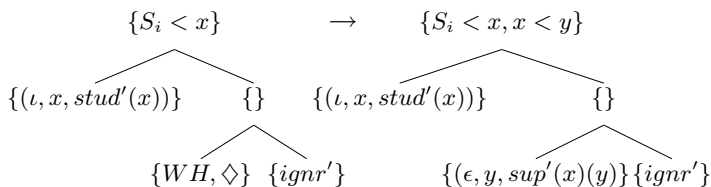
Fragments   Bare fragments (3) may be analysed as taking a previous structure from context as a starting point for parsing (or generation). *WH*-expressions are analysed as particular forms of metavariables, so parsing a *wh*-question yields a type-complete but open formula, which the term presented by a subsequent fragment can update:

(3) | A: What did you eat for breakfast?
    | B: Porridge.

Parsing the fragment involves constructing an unfixed node, and merging it with the contextually available structure, so characterising the wellformedness/interpretation of fragment answers to questions without any additional mechanisms: the term $(\epsilon, x, porridge'(x))$ stands in a licensed growth relation from the metavariable WH provided by the lexical actions of *what*.

Functional questions (Ginzburg and Sag, 2000) with their fragment answers (4) pose no problem. As the *wh*-question contains a metavariable, the scope evaluation cannot be completed; completion of structure and evaluation of scope can then be effected by merging in the term the answer provides, identifying any introduced metavariable in this context (the genitive imposes narrow scope of the introduced epsilon term):

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

$$\{S_i < x\} \qquad \rightarrow \qquad \{S_i < x, x < y\}$$

$$\{(\iota, x, stud'(x))\} \quad \{\} \quad \{(\iota, x, stud'(x))\} \qquad \{\}$$

$$\{WH, \Diamond\} \ \{ignr'\} \qquad \{(\epsilon, y, sup'(x)(y)\} \{ignr'\}$$

VP Ellipsis   Anaphoric devices such as pronouns and VP ellipsis are analysed as decorating tree nodes with metavariables licensing update from context using either established terms, or,

for ellipsis, (lexical) tree-update actions. Strict readings of VP ellipsis result from taking a suitable semantic formula directly from a tree node in context: any node $n \in (T_0 \cup T)$ of suitable type $(e \rightarrow t)$ with no outstanding requirements. Sloppy readings involve re-use of actions: any sequence of actions $(a_1; a_2; \ldots ; a_n) \in (A_0 \cup A)$ can be used (given the appropriate elliptical trigger) to extend the current tree T if this provides a formula of type $e \rightarrow t$.[13] This latter approach, combined with the representation of quantified elements as terms, allows a range of phenomena, including those which are problematic for other (abstraction-based) approaches (for discussion see (Dalrymple et al., 1991)):

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

The actions associated with A's use of *read him his rights* in (5) include the projection of a metavariable associated with *him*, and its resolution to the term in context associated with *Bill*. B's ellipsis allows this action sequence to be re-used, again projecting a metavariable and resolving it, this time (given the new context) to the term provided by parsing *Tom*. This leads to a copy of *Tom* within the constructed predicate, and a sloppy reading.

This analysis also applies to yield parallellism effects in scoping (Hirschbühler, 1982; Shieber et al., 1996), allowing narrow scope construal for indefinites in subject position:

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

Resolution of the underspecification in the scope statement associated with an indefinite can be performed at two points: either at the immediate point of processing the lexical actions, or at the later point of compiling the resulting node's interpretation within the emergent tree.[14] In (6), narrow scope can be assigned to the subject in A's utterance via this late assignment of scope; at this late point in the

---

[13]In its re-use of actions provided by context, this approach to ellipsis is essentially similar to the glue language approach (see (Asudeh and Crouch, 2002) and papers in (Dalrymple, 1999) but, given the lack of independent syntax /semantics vocabularies, the need for an intermediate mapping language is removed.

[14]This pattern parallels expletive pronouns which equally allow a delayed update (Cann, 2003).

parse process, the term constructed from the object node will have been entered into the set of scope statements, allowing the subject node to be dependent on the following quantified expression. The elliptical word *did* in *B*'s utterance will then license re-use of these late actions, repeating the procedures used in interpreting *A*'s antecedent and so determining scope of the new subject relative to the object.

Again, these analyses are possible because parsing and generation processes share incrementally built structures and contextual parsing actions, with this being ensured by the incrementality of the grammar formalism itself.

6   Alignment & Routinization

The parsing and generation processes must both search the lexicon for suitable entries at every step (i.e. when parsing or generating each word). For generation in particular, this is a computationally expensive process in principle: every possible word/action pair must be tested – the current partial tree extended and the result checked for goal tree subsumption. As proposed by O&P (though without formal definitions or implementation) our model of context now allows a strategy for minimising this effort: as it includes previously used words and actions, a subset of such actions can be re-used in extending the current tree, avoiding full lexical search altogether. High frequency of elliptical constructions is therefore expected, as ellipsis licenses such re-use; the same can be said for pronouns, as long as they (and their corresponding actions) are assumed to be pre-activated or otherwise readily available from the lexicon.

As suggested by O&P, this can now lead directly to a model of alignment phenomena, characterisable as follows. For the generator, if there is some action $a \in (A_0 \cup A)$ suitable for extending the current tree, a can be re-used, generating the word w which occupies the corresponding position in the sequence W_0 or W. This results in *lexical* alignment – repeating w rather than choosing an alternative word from the lexicon. Alignment of *syntactic* structure (e.g. preserving double-object or full PP forms in the use of a verb such as *give* rather than shifting to the semantically equivalent form (Branigan et al., 2000)) also follows in virtue of the procedural action-based specification of lexical content. A word such as *give* has two possible lexical actions a' and a'' despite semantic equivalence of output, corresponding to the two alternative

forms. A previous use will cause either a' or a'' to be present in $(A_0 \cup A)$; re-use of this action will cause the same form to be repeated.[15]

A similar definition holds for the parser: for a word w presented as input, if $w \in (W_0 \cup W)$ then the corresponding action a in the sequence A_0 or A can be used without consulting the lexicon. Words will therefore 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 $(a_1; a_2; \ldots; a_n) \in (A_0 \cup A)$ can be re-used by a generator, producing the corresponding word sequence $(w_1; w_2; \ldots; w_n) \in (W_0 \cup W)$; and similarly the sub-sequence of words $(w_1; w_2; \ldots; w_n) \in (W_0 \cup W)$ will cause the parser to use the corresponding action sequence $(a_1; a_2; \ldots; a_n) \in (A_0 \cup A)$. 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 be alignment across different levels, are expressible without invoking distinct levels of syntactic or lexical structure, since context, content and lexical actions are all defined in terms of the same tree configurations.

7   Summary

The inherent left-to-right incrementality and monotonicity of DS as a grammar formalism allows both parsing and generation processes to be not only incremental but closely coupled, sharing structures and context. This enables shared utterances, cross-speaker elliptical phenomena and alignment to be modelled straightforwardly. A prototype system has been implemented in Prolog which reflects the model given here, demonstrating shared utterances and alignment phenomena in simple dialogue sequences. The significance of this direct reflection of psycholinguistic data is to buttress the DS claim that the strictly serial incrementality of processing is not merely essential to the modelling of natural-language parsing, but

---

[15]Most frameworks would have to reflect this via preferences defined over syntactic rules or parallelisms with syntactic trees in context, both problematic.

to the design of the underlying grammar formalism itself.

Acknowledgements

References

A. Asudeh and R. Crouch. 2002. Derivational parallelism and ellipsis parallelism. In *Proceedings of WCCFL 21*.

P. Blackburn and W. Meyer-Viol. 1994. Linguistics, logic and finite trees. *Bulletin of the IGPL*, 2:3–31.

H. Branigan, M. Pickering, and A. Cleland. 2000. Syntactic co-ordination in dialogue. *Cognition*, 75:13–25.

R. Cann. 2003. Semantic underspecification and the interpretation of copular clauses in English. In *Where Semantics Meets Pragmatics*. University of Michigan.

H. H. Clark and D. Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22:1–39.

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

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

M. Dalrymple, editor. 1999. *Syntax and Semantics in Lexical Functional Grammar: The Resource-Logic Approach*. MIT Press.

V. Ferreira. 1996. Is it better to give than to donate? Syntactic flexibility in language production. *Journal of Memory and Language*, 35:724–755.

S. Garrod and A. Anderson. 1987. Saying what you mean in dialogue. *Cognition*, 27:181–218.

J. Ginzburg and I. A. Sag. 2000. *Interrogative Investigations*. CSLI Publications.

R. Hausser. 1989. *Computation of Language*. Springer-Verlag.

P. Hirschbühler. 1982. VP deletion and across-the-board quantifier scope. In *Proceedings of NELS 12*.

A. Joshi and S. Kulick. 1997. Partial proof trees as building blocks for a categorial grammar. *Linguistics and Philosophy*, 20:637–667.

R. Kaplan and A. Zaenen. 1989. Long-distance dependencies, constituent structure, and functional uncertainty. In M. Baltin and A. Kroch, editors, *Alternative Conceptions of Phrase Structure*, pages 17–42. University of Chicago Press.

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

R. Kibble, W. Meyer-Viol, D. Gabbay, and R. Kempson. 2001. Epsilon terms: a labelled deduction account. In H. Bunt and R. Muskens, editors, *Computing Meaning*. Kluwer Academic Publishers.

M. Marcus. 1987. Deterministic parsing and description theory. In P. Whitelock et al., editor, *Linguistic Theory and Computer Applications*, pages 69–112. Academic Press.

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

R. Millikan. 2004. *The Varieties of Meaning*. MIT Press.

G. Neumann. 1994. *A Uniform Computational Model for Natural Language Parsing and Generation*. Ph.D. thesis, Universität des Saarlandes.

M. Otsuka and M. Purver. 2003. Incremental generation by incremental parsing. In *Proceedings of the 6th CLUK Colloquium*.

M. Pickering and S. Garrod. 2004. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, forthcoming.

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

M. Purver and M. Otsuka. 2003. Incremental generation by incremental parsing: Tactical generation in Dynamic Syntax. In *Proceedings of the 9th European Workshop in Natural Language Generation (ENLG-2003)*.

S. Shieber, F. Pereira, and M. Dalrymple. 1996. Interactions of scope and ellipsis. *Linguistics and Philosophy*, 19:527–552.

M. Stone and C. Doran. 1997. Sentence planning as description using tree-adjoining grammar. In *Proceedings of the 35th Annual Meeting of the ACL*, pages 198–205.