Incrementality, Alignment and Shared Utterances

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

This paper describes an implemented prototype dialogue model within the Dynamic Syntax (DS) framework (Kempson et al., 2001) which directly reflects dialogue phenomena such as alignment, routinization and shared utterances. In DS, word-by-word incremental parsing and generation are defined in terms of *actions* on semantic tree structures. This paper proposes a model of dialogue context which includes these trees and their associated actions, and shows how alignment and routinization result directly from minimisation of lexicon search (and hence speaker's effort), and how switch of speaker/hearer roles in shared utterances can be seen as a switch between incremental processes directed by different goals, but sharing the same (partial) data structures.

1 Introduction

Study of dialogue has been proposed by (Pickering and Garrod, 2004) as the major new challenge facing both linguistic and psycholinguistic theory. Two of the phenomena which they highlight as common in dialogue, but posing a significant challenge to and having received little attention in theoretical linguistics, are *alignment* (including *routinization*) and *shared utterances*. Alignment describes the way that dialogue participants apparently mirror each other's patterns at many levels (including lexical word choice and syntactic structure), while routinization describes their convergence on set descriptions (words or sequences of words) for a particular reference or sense. Shared utterances are those in which participants shift between the roles of parser and producer midway through an utterance:[1]

(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.

These are especially problematic for theoretical or computational approaches in which parsing and generation are seen as separate disconnected processes, even more so when as applications of a grammar formalism whose output is the set of wellformed strings:[2] the initial hearer must parse an input which is not a standard constituent, and assign a (partial) interpretation, then presumably complete that representation and generate an output from it which takes the previous words and their syntactic form into account but does not produce them. The initial speaker must also be able to integrate these two fragments.

In this paper we describe a new approach and implementation within the Dynamic Syntax (DS) framework (Kempson et al., 2001) which al-

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

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

lows these phenomena to be straightforwardly explained. By defining a suitably structured concept of context, and adding this to the basic word-by-word incremental parsing and generation models of (Kempson et al., 2001; Otsuka and Purver, 2003; Purver and Otsuka, 2003), we show how alignment phenomena result directly from minimisation of effort on the part of both hearer and speaker independently (implemented as minimisation of lexical search in parsing and generation), and how the switch in roles at any stage of a sentence can be seen as a switch between processes which are directed by different goals, but which share the same incrementally built data structures.

2 Background

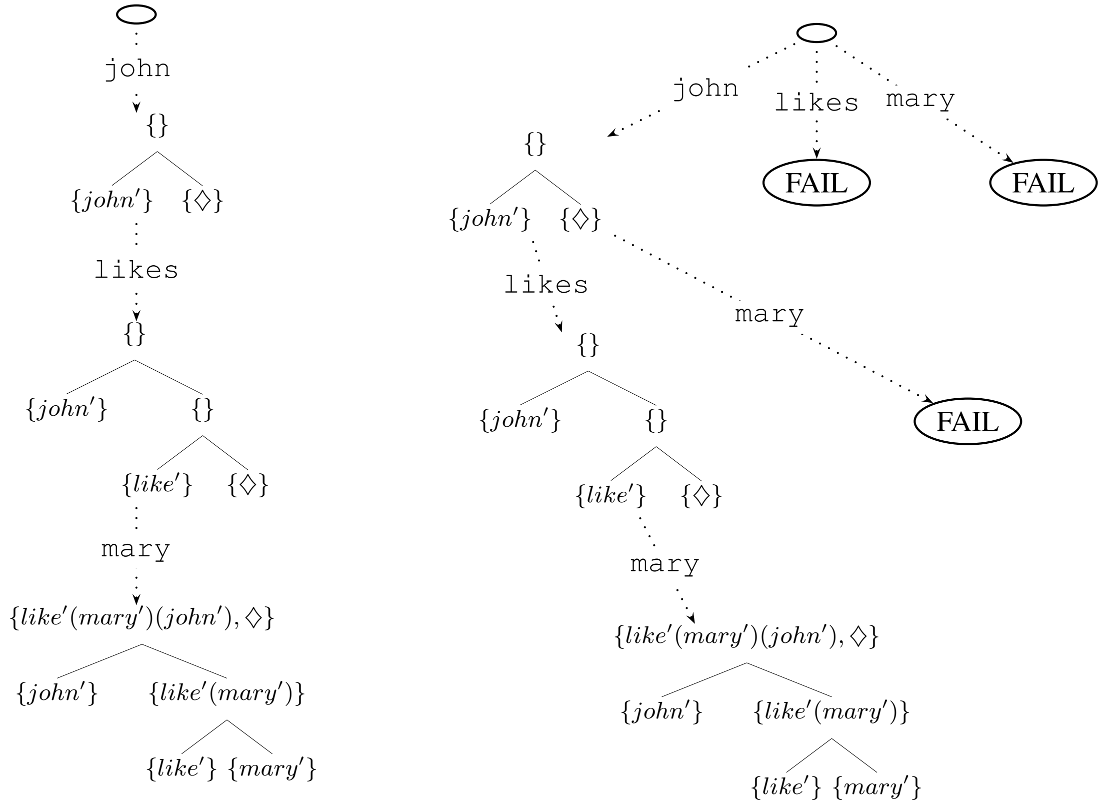
DS is a parsing-directed grammar formalism in which a decorated tree structure representing a semantic interpretation for a string is incrementally projected following the left-right sequence of the words. Importantly, this tree is not a model of syntactic structure, but is strictly semantic, being a record of how some formula representing interpretation assigned to the sentence in context is compiled, with the topnode of the tree being decorated with some (type t) formula, and dominated nodes with subterms of that formula. In this process, sequences of linked trees may be constructed, sharing decorations through anaphoric processes, e.g. for relative clause construal. 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), and there is no central use-neutral grammar of the kind assumed by most approaches to parsing/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. 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.[3] With all quantification expressed as type e terms, the standard grounds for mismatch between syntactic and semantic analysis for all NPs are 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 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 will be underspecified in one or more ways, each being associated with a requirement for subsequent update: the tree may lack a full set of nodes; some relation between nodes may be only partially specified (as in the parsing of long-distance dependency effects); some node may lack a full formula specification (as in the parsing of anaphoric/expletive expressions); and the sequence of scope constraints may be incomplete. Once all requirements are satisfied and all partiality and underspecification resolved, trees are *complete*, parsing is successful and the input string is said to be grammatical. 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 produced by the portion of the string so far consumed which remain candidates for completion.

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

[3] For formal details of this approach to quantification see (Kempson et al., 2001) chapter 7.

Figure 1: **Parsing *"john likes mary"* . . .** **. . . and generating *"john likes mary"***



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 A Model of Context

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: while some notion of context was assumed no formal model or implementation was given.[4] 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.[5] 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

[4] 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.

[5] 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.

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,[6] 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.[7]

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).

Anaphora & Ellipsis Anaphoric devices such as pronouns and VP ellipsis are analysed as decorating tree nodes with metavariables to be updated from context using terms established, or, for ellipsis, the (lexical) tree-update actions. Strict readings of VP ellipsis result from taking a suitable semantic formula directly from a tree node in context; sloppy readings involve reuse of actions. This action re-use approach, combined with the representation of quantified elements as terms, allows even ellipsis phenomena which are problematic for other e.g. abstraction-based approaches (see (Dalrymple et al., 1991) for discussion):

(3)
A: A policeman who arrested Bill read him his rights.
B: The policeman who arrested Harry did too.

Here re-use of the actions associated with *read him his rights* allows *Harry* to be selected as antecedent for the metavariable projected by these re-used actions, given the new context, leading to a new term and a sloppy reading. Other forms of ellipsis such as bare fragments involve taking a previous structure from context as a starting point for parsing (here *wh*-expressions are analysed as particular forms of metavariables, so parsing the question yields an open formula which the term

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

[7] 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.

presented by the fragment updates):

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

4 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. If a first search through context finds a subset of such actions which can be re-used in extending the current tree, full lexical search can be avoided altogether. Even given a more complex model of the lexicon which might avoid searching all possible words during generation (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.

High frequency of elliptical constructions is therefore expected, as ellipsis licenses the use of context, either in providing some term directly or in licensing re-use of actions which context makes available; 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.

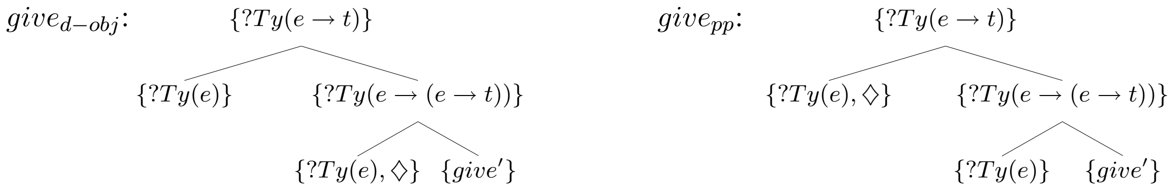
Lexical Alignment As suggested by O&P, this can now lead directly to a model of alignment phenomena, characterisable as follows. For the generator, if there is some action $a \in (A_0 \cup A)$ suitable for extending the current tree, a can be re-used, generating the word w which occupies the corresponding position in the sequence W_0 or W. This results in *lexical* alignment – repeating w rather than choosing an alternative but as yet unused word from the lexicon.

In this connection, re-use of the actions associated with the construction of semantic trees is importantly distinct from re-use of these trees and the terms that decorate them. For example, the actions associated with the parse of a pronoun decorate a node with a metavariable which must then be provided with a fully specified value from a term in context (see section 3); subsequent re-use of this action will introduce a new metavariable, rather than merely copying the previous value, and so the resulting value in this case may differ from the value given previously. Such re-use of actions is essential to construal of indexical pronouns, such as *I* and *you*, as their actions will require values to be assigned from the current context (which must contain information about the identity of the current speaker and addressee) rather than copying values from previous uses.

This re-use of actions applies also to quantifying expressions, e.g. indefinites. By definition, on the DS approach, the construal of a quantified noun phrase introduces a new variable as formula decoration, and re-use of these actions will not therefore license introduction of the same term. This is in contrast to the approach of (Lemon et al., 2003) in which strings are re-used in a way that licenses the same construal, necessitating a special rule to prevent a generator from re-using indefinite NPs with the same interpretation as the antecedent occurrence.

Syntactic Alignment 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 a' and a'' corresponding to the two

Figure 2: Output of alternative lexical actions for *give*


$give_{d-obj}$:
$\{?Ty(e \rightarrow t)\}$
$\{?Ty(e)\}$
$\{?Ty(e \rightarrow (e \rightarrow t))\}$
$\{?Ty(e), \Diamond\}$
$\{give'\}$
$give_{pp}$:
$\{?Ty(e \rightarrow t)\}$
$\{?Ty(e), \Diamond\}$
$\{?Ty(e \rightarrow (e \rightarrow t))\}$
$\{?Ty(e)\}$
$\{give'\}$

alternative forms (figure 2). 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.

Repetition of adjective structures as attributive or in a predicative relative-clause (*a green book* vs. *a book which is green* (Cleland and Pickering, 2003)) can be explained in the same way. Adjective construal in DS is distinguished by whether a linked tree structure is constructed before the head noun (by the lexical actions associated with attributive adjectives) or after the head (by the actions associated with a relative pronoun); and re-use of these actions will cause repetition of form. So again the 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.

Semantic Alignment & Routines 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 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).

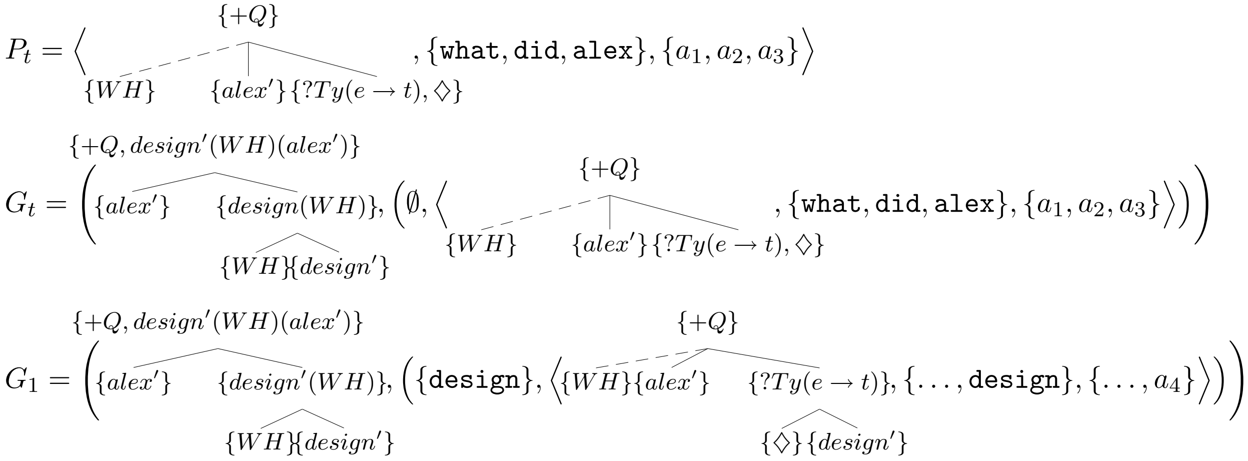
It is notable that these various patterns of alignment, said by Pickering and Garrod (2004) to be alignment across different levels, are expressible without invoking distinct levels of syntactic or lexical structure, since context, content and lexical actions are all defined in terms of the same tree configurations. Note also that this context-based approach models both speaker and hearer actions without any need for meta-level calculations about their interlocutor.

5 Shared Utterances

O&P suggest an analysis of shared utterances, 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

Figure 3: Transition from hearer to speaker: *"What did Alex . . . / . . . design?"*


$P_t = \langle$
$\{+Q\}$
$\{WH\}$
$\{alex'\}$
$\{?Ty(e \rightarrow t), \diamondsuit\}$
$, \{\texttt{what}, \texttt{did}, \texttt{alex}\}, \{a_1, a_2, a_3\} \rangle$
$G_t = ($
$\{+Q, design'(WH)(alex')\}$
$\{alex'\}$
$\{design(WH)\},$
$\{WH\}\{design'\}$
$(\emptyset, \langle$
$\{+Q\}$
$\{WH\}$
$\{alex'\}$
$\{?Ty(e \rightarrow t), \diamondsuit\}$
$, \{\texttt{what}, \texttt{did}, \texttt{alex}\}, \{a_1, a_2, a_3\} \rangle))$
$G_1 = ($
$\{+Q, design'(WH)(alex')\}$
$\{alex'\}$
$\{design'(WH)\},$
$\{WH\}\{design'\}$
$(\{\texttt{design}\}, \langle$
$\{+Q\}$
$\{WH\}\{alex'\}$
$\{?Ty(e \rightarrow t)\},$
$\{\diamondsuit\}\{design'\}$
$\{\ldots, \texttt{design}\}, \{\ldots, a_4\} \rangle))$

usual until transition,[8] 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 3 as a display of the interpretation process for example (2).[9] 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. the proposition to be expressed must be subsumed by the incomplete proposition 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 for B).[10] 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.

Transition Effects Just as with alignment, the change in reference of the indexicals *I* and *you* across the speaker/hearer transition (example (5)) 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:

(5) | *A:* Have you read . . .
| *B:* Your latest chapter?

Note that there is no constraint on when in

[8] 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.

[9] Figure 3 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.

[10] 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.

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. 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:

(6) *A:* Did a nurse . . .
B: See every patient?

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, even, parallelling expletive pronouns, including some term subsequently constructed.

6 Summary

The left-to-right incrementality and monotonicity of DS, together with the close coupling of parsing and generation processes, allow shared utterances to be modelled in a straightforward fashion. Alignment phenomena can be predicted given a suitable model of context already motivated by the DS treatment of anaphora and ellipsis. A prototype system has been implemented in Prolog which reflects the model given here, demonstrating shared utterances and alignment phenomena in simple dialogue sequences.

Acknowledgements

This paper is an extension of joint work on the DS framework with Wilfried Meyer-Viol, on defining a context-dependent formalism with Ronnie Cann, and on DS generation with Masayuki Otsuka. Each has provided ideas and input without which the current results would have differed, although any mistakes here are ours. Thanks are also due to the anonymous reviewers. This work was supported by the ESRC (RES-000-22-0355) and (for the second author) by the Leverhulme Trust.

References

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.

A. Cleland and M. Pickering. 2003. The use of lexical and syntactic information in language production. *Journal of Memory and Language*, 49:214–230.

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

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

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

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

O. Lemon, A. Gruenstein, R. Gullett, A. Battle, L. Hiatt, and S. Peters. 2003. Generation of collaborative spoken dialogue contributions in dynamic task environments. In *Proceedings of the AAAI Spring Symposium on Natural Language Generation in Spoken and Written Dialogue*.

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

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

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*.

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