Incrementality, Speaker-Hearer Switching and the Disambiguation Challenge

Ruth Kempson, Eleni Gregoromichelaki
King's College London
{ruth.kempson, eleni.gregor}@kcl.ac.uk

Yo Sato
University of Hertfordshire
y.sato@herts.ac.uk

Abstract

Taking so-called *split utterances* as our point of departure, we argue that a new perspective on the major challenge of disambiguation becomes available, given a framework in which both parsing and generation incrementally involve the same mechanisms for constructing trees reflecting interpretation (*Dynamic Syntax*: (Cann et al., 2005; Kempson et al., 2001)). With all dependencies, syntactic, semantic and pragmatic, defined in terms of incremental progressive tree growth, the phenomenon of speaker/hearer role-switch emerges as an immediate consequence, with the potential for clarification, acknowledgement, correction, all available incrementally at any sub-sentential point in the interpretation process. Accordingly, at all intermediate points where interpretation of an utterance subpart is not fully determined for the hearer in context, uncertainty can be resolved immediately by suitable clarification/correction/repair/extension as an exchange between interlocutors. The result is a major check on the combinatorial explosion of alternative structures and interpretations at each choice point, and the basis for a model of how interpretation in context can be established without either party having to make assumptions about what information they and their interlocutor share in resolving ambiguities.

1 Introduction

A major characteristic of dialogue is effortless switching between the roles of hearer and speaker. Dialogue participants seamlessly shift between parsing and generation bi-directionally across any syntactic dependency, without any indication of there being any problem associated with such shifts (examples from Howes et al. (in prep)):

(1) Conversation from A and B, to C:
    A: We're going
    B: to Bristol, where Jo lives.

(2) A smelling smoke comes into the kitchen:
    A: Have you burnt
    B the buns. Very thoroughly.
    A: But did you burn
    B: Myself? No. Luckily.

(3) A: Are you left or
    B: Right-handed.

Furthermore, in no case is there any guarantee that the way the shared utterance evolves is what either party had in mind to say at the outset, indeed obviously not, as otherwise the exchange risks being otiose. This flexibility provides a vehicle for ongoing clarification, acknowledgement, corrections, repairs etc. ((6)-(7) from (Mills, 2007)):

(4) A: I'm seeing Bill.
    B: The builder?
    A: Yeah, who lives with Monica.

(5) A: I saw Don
    B: John?
    A: Don, the guy from Bristol.

(6) A: I'm on the second switch
    B: Switch?
    A: Yeah, the grey thing

(7) A: I'm on the second row third on the left.
    B: What?
    A: on the left

The fragmental utterances that constitute such incremental, joint contributions have been analysed as falling into discrete structural types according to their function, in all cases resolved to propositional types by combining with appropriate abstractions from context (Fernández, 2006; Purver, 2004). However, any such fragment and their resolution may occur as mid-turn interruptions, well before any emergent propositional structure is completed:

(8) A: They X-rayed me, and took a urine sample, took a blood sample. Er, the doctor ...

B: Chorlton?

A: Chorlton, mhm, he examined me, erm, he, he said now they were on about a slight [shadow] on my heart. [BNC: KPY 1005-1008]

The advantage of such ongoing, incremental, joint conversational contributions is the effective narrowing down of the search space out of which hearers select (a) interpretations to yield some commonly shared understanding, e.g. choice of referents for NPs, and, (b) restricted structural frames which allow (grammatical) context-dependent fragment resolution, i.e. exact specifications of what contextually available structures resolve elliptical elements. This seems to provide an answer as to why such fragments are so frequent and undemanding elements of dialogue, forming the basis for the observed *coordination* between participants: successive resolution at sub-sentential stages yields a progressively jointly established common ground, that can thereafter be taken as a secure, albeit individual, basis for filtering out interpretations inconsistent with such confirmed knowledge-base (see (Poesio and Rieser, 2008; Ginzburg, forthcmg) etc). All such dialogue phenomena, illustrated in (1)-(8), jointly and incrementally achieved, we address with the general term *split utterances*.

However, such exchanges are hard to model within orthodox grammatical frameworks, given that usually it is the sentence/proposition that is taken as the unit of syntactic/semantic analysis; and they have not been addressed in detail within such frameworks, being set aside as deviant, given that such grammars in principle do not specify a concept of *grammaticality* that relies on a description of the context of occurrence of a certain structure (however, see Poesio and Rieser (2008) for German *completions*). In so far as fragment utterances are now being addressed, the pressure of compatibility with sentence-based grammars is at least partly responsible for analyses of e.g. clarificatory-request fragments as sentential in nature (Ginzburg and Cooper, 2004). But such analyses fail to provide a basis for incrementally resolved clarification requests such as the interruption in (8) where no sentential basis is yet available over which to define the required abstraction of contextually provided content.

In the psycholinguistic literature, on the other hand, there is broad agreement that *incrementality* is a crucial feature of parsing with semantic interpretation taking place as early as possible at the sub-sentential level (see e.g. (Sturt and Crocker, 1996)). Nonetheless, this does not, in and of itself, provide a basis for explaining the ease and frequency of split utterances in dialogue: the interactive coordination between the parsing and production activities, one feeding the other, remains as a challenge.

In NLP modelling, parsing and generation algorithms are generally dissociated from the description of linguistic entities and rules, i.e. the grammar formalisms, which are considered either to be independent of processing ('process-neutral') or to require some additional generation- or parsing-specific mechanisms to be incorporated. However, this point of view creates obstacles for a successful account of data as in (1)-(8). Modelling those would require that, for the current speaker, the initiated generation mechanism has to be displaced mid-production without the propositional generation task having been completed. Then the parsing mechanism, despite being independent of, indeed in some sense the reverse of, the generation component, has to take over mid-sentence as though, in some sense there had been parsing involved up to the point of switchover. Conversely, for the hearer-turned-speaker, it would be necessary to somehow connect their parse with what they are now about to produce in order to compose the meaning of the combined sentence. Moreover, in both directions of switch, as (2) shows, this is not a phenomenon of both interlocutors intending to say the same sentence: as (3) shows, even the function of the utterance (e.g. question/answer) can alter in the switch of roles and such fragments can play two roles (e.g. question/completion) at the same time (e.g. (2)). Hence the grammatical integration of such joint contributions must be flexible enough to allow such switches which means that such fragment resolutions must occur before the computation of intentions at the pragmatic level. So the ability of language users to successfully process such utterances, even at sub-sentential levels, means that modelling their grammar requires fine-grained grammaticality definitions able to characterise and integrate sub-sentential fragments in turns jointly constructed by speaker and hearer.

This can be achieved straightforwardly if features like incrementality and context-dependent processing are built into the grammar architecture itself. The modelling of split utterances then becomes straightforward as each successive processing step exploits solely the grammatical apparatus to succeed or fail. Such a view notably does not invoke high-level decisions about speaker/hearer intentions as part of the mechanism itself. That this is the right view to take is enhanced by the fact that as all of (1)-(8) show, neither party in such role-exchanges can definitively know in advance what will emerge as the eventual joint proposition. If, to the contrary, generation decisions are modelled as involving intentions for whole utterances, there will be no the basis for modelling how such incomplete strings can be integrated in suitable contexts, with joint propositional structures emerging before such *joint intentions* have been established.

An additional puzzle, equally related to both the challenges of disambiguation and the status of modelling speaker's intentions as part of the mechanism whereby utterance interpretation takes place, is the common occurrence of hearers NOT being constrained by any check on consistency with speaker intentions in determining a putative interpretation, failing to make use of well established shared knowledge:

(9) A: I'm going to cook salmon, as John's coming.
     B: What? John's a vegetarian.
     A: Not my brother. John Smith.

(10) A: Why don't you have cheese and noodles?
     B: Beef? You KNOW I'm a vegetarian

Such examples are problematic for any account that proposes that interpretation mechanisms for utterance understanding solely depend on selection of interpretations which either the speaker could have intended (Sperber and Wilson, 1986; Carston, 2002), or ones which are compatible with checking consistency with the common ground/plans established between speaker and hearer (Poesio and Rieser, 2008; Ginzburg, forthcmg), mutual knowledge, etc. (Clark, 1996; Brennan and Clark, 1996). To the contrary, the data in (9)-(10) tend to show that the full range of interpretations computable by the grammar has in principle to be available at all choice points for construal, without any filter based on plausibility measures, thus leaving the disambiguation challenge still unresolved.

In this paper we show how with speaker and hearer in principle using the same mechanisms for construal, equally incrementally applied, such disambiguation issues can be resolved in a timely manner which in turn reduces the multiplication of structural/interpretive options. As we shall see, what connects our diverse examples, and indeed underpins the smooth shift in the joint endeavour of conversation, lies in *incremental*, context-dependent processing and *bidirectionality*, essential ingredients of the *Dynamic Syntax* (Cann et al., 2005) dialogue model.

2   Incrementality in Dynamic Syntax

Dynamic Syntax (*DS*) is a procedure-oriented framework, involving incremental processing, i.e. strictly sequential, word-by-word interpretation of linguistic strings. The notion of incrementality in DS is closely related to another of its features, the *goal-directedness* of BOTH parsing and generation. At each stage of processing, *structural predictions* are triggered that could fulfill the goals compatible with the input, in an underspecified manner. For example, when a proper name like *Bob* is encountered sentence-initially in English, a semantic predicate node is predicted to follow $(?Ty(e \rightarrow t))$, amongst other possibilities.

By way of introducing the reader to the DS devices, let us look at some formal details with an example, *Bob saw Mary*. The 'complete' semantic representation tree resulting after the complete processing of this sentence is shown in Figure 2 below. A DS tree is formally encoded with the tree logic *LOFT* (Blackburn and Meyer-Viol (1994)), we omit these details here) and is generally binary configurational, with annotations at every node. Important annotations here, see the (simplified) tree below, are those which represent semantic formulae along with their type information (e.g. '$Ty(x)$') based on a combination of the epsilon and lambda calculi[1].

Such complete trees are constructed, starting from a radically underspecified annotation, the *axiom*, the leftmost minimal tree in Figure 2, and going through *monotonic updates* of partial, or *structurally underspecified*, trees. The outline of this process is illustrated schematically in Figure 2. Crucial for expressing the goal-directedness are *requirements*, i.e. unrealised but expected

---

[1]These are the adopted semantic representation languages in DS but the computational formalism is compatible with other semantic-representation formats
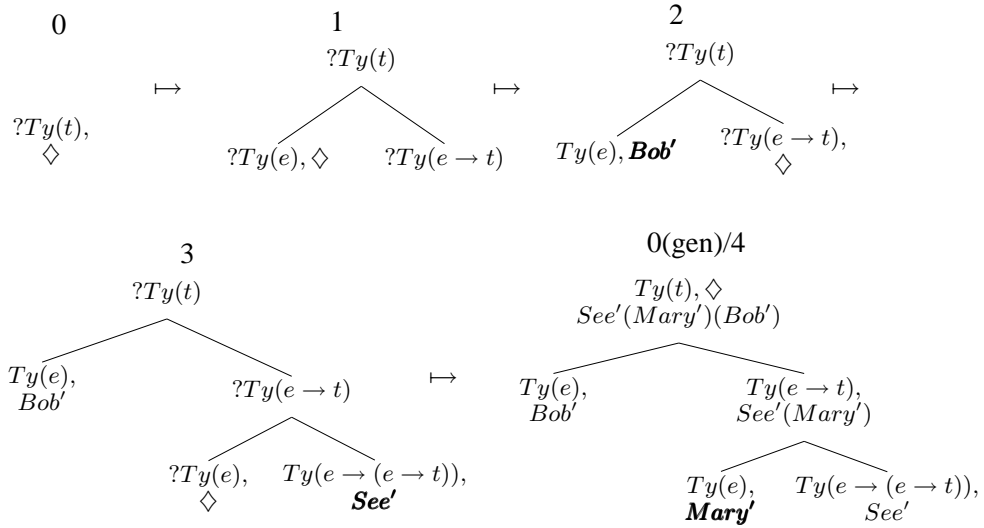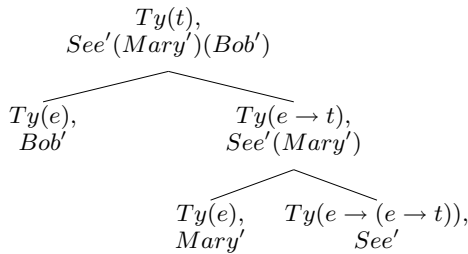
Figure 2: Monotonic tree growth in DS

Figure 1: A DS complete tree

node/tree specifications, indicated by '?' in front of annotations. The axiom says that a proposition (of type t, $Ty(t)$) is expected to be constructed. Furthermore, the *pointer*, notated with '$\diamondsuit$' indicates the 'current' node in processing, namely the one to be processed next, and governs word order.

Updates are carried out by means of applying *actions*, which are divided into two types. *Computational actions* govern general tree-constructional processes, such as moving the pointer, introducing and updating nodes, as well as compiling interpretation for all non-terminal nodes in the tree. In our example, the update of (1) to (2) is executed via computational actions specific to English, expanding the axiom to the subject and predicate nodes, requiring the former to be processed next by the position of the $\diamondsuit$. Construction of only weakly specified tree relations (*unfixed nodes*) can also be induced, characterised only as dominance by some current node, with subsequent update required. Individual lexical items also provide procedures for

building structure in the form of *lexical actions*, inducing both nodes and annotations. For example, in the update from (2) to (3), the set of lexical actions for the word *see* is applied, yielding the predicate subtree and its annotations. Thus *partial trees* grow incrementally, driven by procedures associated with particular words as they are encountered.

Requirements embody structural predictions as mentioned earlier. Thus unlike the conventional bottom-up parsing,[2] the DS model takes the parser/generator to entertain some predicted *goal(s)* to be reached eventually at any stage of processing, and this is precisely what makes the formalism incremental. This is the characterisation of incrementality adopted by some psycholinguists under the appellation of *connectedness* (Sturt and Crocker, 1996; Costa et al., 2002): an encountered word always gets 'connected' to a larger, predicted, tree.

Individual DS trees consist of predicates and their arguments. Complex structures are obtained via a general tree-adjunction operation licensing the construction of so-called *LINKed* trees, pairs of trees where sharing of information occurs. In its simplest form this mechanism is the same one which provides the potential for compiling in-

---

[2]The examples in (1)-(8) also suggest the implausibility of purely bottom-up or head-driven parsing being adopted directly, because such strategies involve waiting until all the daughters are gathered before moving on to their projection. In fact, the parsing strategy adopted by DS is somewhat similar to mixed parsing strategies like the left-corner or Earley algorithm to a degree. These parsing strategic issues are more fully discussed in Sato (forthcmg).
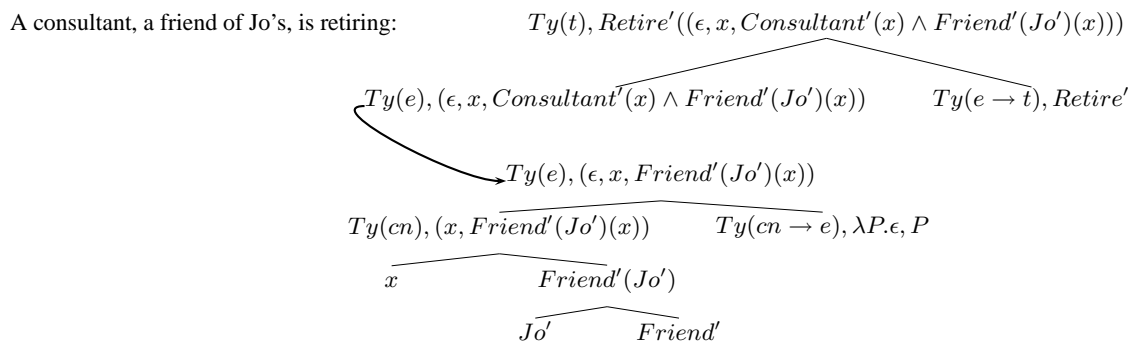
A consultant, a friend of Jo's, is retiring:
$$Ty(t), Retire'((\epsilon, x, Consultant'(x) \wedge Friend'(Jo')(x)))$$

$$Ty(e), (\epsilon, x, Consultant'(x) \wedge Friend'(Jo')(x)) \qquad Ty(e \to t), Retire'$$

$$Ty(e), (\epsilon, x, Friend'(Jo')(x))$$

$$Ty(cn), (x, Friend'(Jo')(x)) \qquad Ty(cn \to e), \lambda P.\epsilon, P$$

$$x \qquad Friend'(Jo')$$

$$Jo' \qquad Friend'$$

Figure 3: Apposition in DS

terpretation for *apposition* constructions as can be seen in Figure (3)[3]. The assumption in the construction of such LINKed structures is that at any arbitrary stage of development, some type-complete subtree may constitute the context for the subsequent parsing of the following string as an adjunct structure candidate for incorporation into the primary tree, hence the obligatory sharing of information in the resulting semantic representation.

More generally, *context* in DS is defined as the storage of *parse states*, i.e., the storing of partial tree, word sequence parsed to date, plus the actions used in building up the partial tree. Formally, a parse state P is defined as a set of triples $\langle T, W, A \rangle$, where: T is a (possibly partial) tree; W is the associated sequence of words; A is the associated sequence of lexical and computational actions. 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 set of triples $P' = \{\ldots, \langle T_i, W_i, A_i \rangle, \ldots\}$ resulting from the previous sentence(s); and the triple $\langle T, W, A \rangle$ itself, the subtree currently being processed. Anaphora and ellipsis construal generally involve re-use of formulae, structures, and actions from the set $\mathcal{C}$. *Grammaticality* of a string of words is then defined relative to its context $\mathcal{C}$, a string being well-formed iff there is a mapping from string onto completed tree with no outstanding requirements given the monotonic processing of that string relative to context. All fragments illustrated above are processed by means of either extending the current

tree, or constructing LINKed structures and transfer of information among them so that one tree provides the context for another, and are licensed as wellformed relative to that context. In particular, fragments like *the doctor* in (8) are licensed by the grammar because they occur at a stage in processing at which the context contains an appropriate structure within which they can be integrated. The definite NP is taken as an anaphoric device, relying on a substitution process from the context of the partial tree to which the node it decorates is LINKed to achieve the appropriate construal and tree-update:

(11) The "parse" tree licensing production of *the doctor*: LINK adjunction

$$?Ty(t)$$

$$Chorlton' \qquad ?Ty(e \to t)$$

$$(Doctor'(Chorlton')), \diamond$$
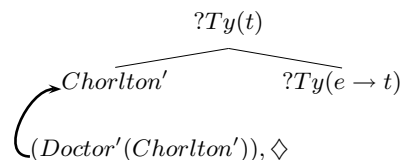
3 Bidirectionality in DS

Crucially, for our current concern, this architecture allows a dialogue model in which generation and parsing function in parallel, following exactly the same procedure in the same order. See Fig (2) for a (simplified) display of the transitions manipulated by a parse of *Bob saw Mary*, as each word is processed and integrated to reach the complete tree. Generation of this utterance from a complete tree follows precisely the same actions and trees from left to right, although the complete tree is available from the start (this is why the complete tree is marked '0' for generation): in this case the eventual message is known by the speaker, though of course not by the hearer. What generation involves in addition to the parse steps is reference

---

[3]*Epsilon terms*, like $\epsilon, x, Consultant'(x)$, stand for witnesses of existentially quantified formulae in the epsilon calculus and represent the semantic content of indefinites in DS. Defined relative to the equivalence $\psi(\epsilon, x, \psi(x)) = \exists x \psi(x)$, their defining property is their reflection of their containing environment, and accordingly they are particularly well-suited to expressing the growth of terms secured by such appositional devices.

to this complete tree to check whether each putative step is consistent with it in order not to be deviated from the legitimate course of action, that is, a *subsumption* check. The trees (1-3) are licensed because each of these subsumes (4). Each time then the generator applies a lexical action, it is licensed to produce the word that carries that action under successful subsumption check: at Step (3), for example, the generator processes the lexical action which results in the annotation 'See'', and upon success and subsumption of (4) license to generate the word *see* at that point ensues.

For split utterances, two more assumptions are pertinent. On the one hand, speakers may have initially only a partial structure to convey: this is unproblematic, as all that is required by the formalism is monotonicity of tree growth, the check being one of *subsumption* which can be carried out on partial trees as well. On the other hand, the utterance plan may change, even within a single speaker. Extensions and clarifications in DS can be straightforwardly generated by appending a LINKed structure projecting the added material to be conveyed (preserving the monotonicity constraint)[4].

(12) I'm going home, with my brother, maybe with his wife.

Such a model under which the speaker and hearer essentially follow the same sets of actions, updating incrementally their semantic representations, allows the hearer to 'mirror' the same series of partial trees, albeit not knowing in advance what the content of the unspecified nodes will be.

4 Parser/generator implementation

The process-integral nature of DS emphasised thus far lends itself to the straightforward implementation of a parsing/generating system, since the 'actions' defined in the grammar directly provide a major part of its implementation. By now it should also be clear that the DS formalism is fully bi-directional, not only in the sense that the same grammar can be used for generation and parsing, but also because the two sets of activities, conventionally treated as 'reverse' processes, are modelled to run in parallel. Therefore, not only can the same sets of actions be used for both processes,

but also a large part of the parsing and generation algorithms can be shared.

This design architecture and a prototype implementation are outlined in (Purver and Otsuka, 2003), and the effort is under way to scale up the DS parsing/generating system incorporating the results in (Gargett et al., 2008; Gregoromichelaki et al., to appear).[5] The parser starts from the axiom (step 0 in Fig.2), which 'predicts' a proposition to be built, and follows the applicable actions, lexical or general, to develop a complete tree. Now, as has been described in this paper, the generator follows exactly the same steps: the axiom is developed through successive updates into a complete tree. The only material difference from – or rather in addition to– parsing is the complete tree (Step 0(gen)/4), given from the very start of the generation task, which is then referred to at each tree update for *subsumption* check. The main point is that despite the obvious difference in their purposes –outputting a string from a meaning versus outputting a meaning from a string– parsing and generation indeed share the *direction* of processing in DS. Moreover, as no intervening level of syntactic structure over the string is ever computed, the parsing/generation tasks are more efficiently incremental in that semantic interpretation is directly imposed at each stage of lexical integration, irrespective of whether some given partially developed constituent is complete.

To clarify, see the pseudocode in the Prolog format below, which is a close analogue of the implemented function that both does parsing and generation of a word (context manipulation is ignored here for reasons of space). The plus and minus signs attached to a variable indicate it must/needn't be instantiated, respectively. In effect, the former corresponds to the input, the latter to the output.

```
(13) parse_gen_word(
     +OldMeaning,±Word,±NewMeaning):-
         apply_lexical_actions(+OldMeaning, ±Word,
         +LexActions, −IntermediateMeaning ),

         apply_computational_actions(
         +IntermediateMeaning, +CompActions,
         ±NewMeaning )
```

OldMeaning is an obligatory input item, which corresponds to the semantic structure constructed so far (which might be just structural tree information initially before any lexical

---

input has been processed thus advocating a strong predictive element even compared to (Sturt and Crocker, 1996). Now notice that the other two variables —corresponding to the word and the new (post-word) meaning— may function either as the input or output. More precisely, this is intended to be a shorthand for either (+OldMeaning,+Word,−NewMeaning) i.e. Word as input and NewMeaning as output, or (+OldMeaning,−Word,+NewMeaning), i.e. NewMeaning as input and Word as output, to repeat, the former corresponding to parsing and the latter to generation.

In either case, the same set of two sub-procedures, the two kinds of actions described in (13), are applied sequentially to process the input to produce the output. These procedures correspond to an incremental 'update' from one partial tree to another, through a word. The whole function is then recursively applied to exhaust the words in the string, from left to right, either in parsing or generation. Thus there is no difference between the two in the order of procedures to be applied, or words to be processed. Thus it is a mere switch of input/output that shifts between parsing and generation.[6]

4.1 Split utterances in Dynamic Syntax

Split utterances follow as an immediate consequence of these assumptions. For the dialogues in (1)-(8), therefore, while A reaches a partial tree of what she has uttered through successive updates as described above, B as the hearer, will follow the same updates to reach the same representation of what he has heard. This provides him with the ability at any stage to become the speaker, interrupting to continue A's utterance, repair, ask for clarification, reformulate, or provide a correction, as and when necessary[7]. According to our model of dialogue, repeating or extending a constituent of A's utterance by B is licensed only if B, the hearer turned now speaker, entertains a message

to be conveyed that matches or extends the parse tree of what he has heard in a monotonic fashion. In DS, this message is a semantic representation in tree format and its presence allows B to only utter the relevant subpart of A's intended utterance. Indeed, this update is what B is seeking to clarify, extend or acknowledge. In DS, B can reuse the already constructed (partial) parse tree in his context, rather than having to rebuild an entire propositional tree or subtree.

The fact that the parsing formalism integrates a strong element of *predictivity*, i.e. the parser is always one step ahead from the lexical input, allows a straightforward switch from parsing to generation thus resulting in an explanation of the facility with which split utterances occur (even without explicit reasoning processes). Moreover, on the one hand, because of *incrementality*, the issue of interpretation-selection can be faced at any point in the process, with corrections/acknowledgements etc. able to be provided at any point; this results in the potential exponential explosion of interpretations being kept firmly in check. And, structurally, such fragments can be integrated in the current partial tree representation only (given the position of the pointer) so there is no structural ambiguity multiplication. On the other hand, for any one of these intermediate check points, *bidirectionality* entails that consistency checking remains internal to the individual interlocutors' system, the fact of their mirroring each other resulting at their being at the same point of tree growth. This is sufficient to ensure that any inconsistency with their own parse recognised by one party as grounds for correction/repair can be processed AS a correction/repair by the other party without requiring any additional metarepresentation of their interlocutors' information state (at least for these purposes). This allows the possibility of building up apparently complex assumptions of shared content, without any necessity of constructing hypotheses of what is entertained by the other, since all context-based selections are based on the context of the interlocutor themselves. This, in its turn, opens up the possibility of hearers constructing interpretations based on selections made that transparently violate what is knowledge shared by both parties, for no presumption of common ground is essential as input to the interpretation process (see, e.g. (9)-(10)).

---

[6]Thus the parsing procedure is dictated by the grammar to a large extent, but importantly, not completely. More specifically, the grammar formalism specifies the state paths themselves, but not *how* the paths should be searched. The DS actions are defined in conditional terms, i.e. what to do as and when a certain condition holds. If a number of actions can be applied at some point during a parse, i.e. locally ambiguity is encountered, then it is up to a particular implementation of the parser to decide which should be traversed first. The current implementation includes suggestions of search strategies.

[7]The account extends the implementation reported in (Purver et al., 2006)

5 Conclusion

It is notable that, from this perspective, no presumption of common ground or hypothesis as to what the speaker could have intended is necessary to determine how the hearer selects interpretation. All that is required is a concept of system-internal consistency checking, the potential for clarification in cases of uncertainty, and reliance at such points on disambiguation/correction/repair by the other party. The advantage of such a proposal, we suggest, is the provision of a fully mechanistic account for disambiguation (cf. (Pickering and Garrod, 2004)). The consequence of such an analysis is that language use is essentially interactive (see also (Ginzburg, forthcmg; Clark, 1996)): the only constraint as to whether some hypothesised interpretation assigned by either party is confirmed turns on whether it is acknowledged or corrected (see also (Healey, 2008)).

Acknowledgements

References

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

Susan E. Brennan and Herbert H. Clark. 1996. Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 22:482–1493.

Ronnie Cann, Ruth Kempson, and Lutz Marten. 2005. *The Dynamics of Language*. Elsevier, Oxford.

Robyn Carston. 2002. *Thoughts and Utterances: The Pragmatics of Explicit Communication*. Blackwell.

Herbert H. Clark. 1996. *Using Language*. Cambridge University Press.

Fabrizio Costa, Paolo Frasconi, Vincenzo Lombardo, Patrick Sturt, and Giovanni Soda. 2002. Enhancing first-pass attachment prediction. In *ECAI 2002: 508-512*.

Raquel Fernández. 2006. *Non-Sentential Utterances in Dialogue: Classification, Resolution and Use*. Ph.D. thesis, King's College London, University of London.

Andrew Gargett, Eleni Gregoromichelaki, Chris Howes, and Yo Sato. 2008. Dialogue-grammar correspondence in dynamic syntax. In *Proceedings of the 12th SEMDIAL (LONDIAL)*.

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

Jonathan Ginzburg. forthcmg. *Semantics for Conversation*. CSLI.

Eleni Gregoromichelaki, Yo Sato, Ruth Kempson, Andrew Gargett, and Christine Howes. to appear. Dialogue modelling and the remit of core grammar. In *Proceedings of IWCS 2009*.

Patrick Healey. 2008. Interactive misalignment: The role of repair in the development of group sub-languages. In R. Cooper and R. Kempson, editors, *Language in Flux*. College Publications.

Christine Howes, Patrick G. T. Healey, and Gregory Mills. in prep. a: An experimental investigation into. . . b: . . . split utterances.

Ruth Kempson, Wilfried Meyer-Viol, and Dov Gabbay. 2001. *Dynamic Syntax: The Flow of Language Understanding*. Blackwell.

Gregory J. Mills. 2007. *Semantic co-ordination in dialogue: the role of direct interaction*. Ph.D. thesis, Queen Mary University of London.

Martin Pickering and Simon Garrod. 2004. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*.

Massimo Poesio and Hannes Rieser. 2008. Completions, coordination, and alignment in dialogue. Ms.

Matthew Purver and Masayuki Otsuka. 2003. Incremental generation by incremental parsing: Tactical generation in Dynamic Syntax. In *Proceedings of the 9th European Workshop in Natural Language Generation (ENLG)*, pages 79–86.

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

Matthew Purver. 2004. *The Theory and Use of Clarification Requests in Dialogue*. Ph.D. thesis, University of London, forthcoming.

Yo Sato. forthcmg. Local ambiguity, search strategies and parsing in dynamic syntax. In Eleni Gregoromichelaki and Ruth Kempson, editors, *Dynamic Syntax: Collected Papers*. CSLI.

Dan Sperber and Deirdre Wilson. 1986. *Relevance: Communication and Cognition*. Blackwell.

Patrick Sturt and Matthew Crocker. 1996. Monotonic syntactic processing: a cross-linguistic study of attachment and reanalysis. *Language and Cognitive Processes*, 11:448–494.