Context-Based Incremental Generation for Dialogue

Matthew Purver[1] and Ruth Kempson[2]

Departments of [1]Computer Science and [2]Philosophy,
King's College London, Strand, London WC2R 2LS, UK
{matthew.purver, ruth.kempson}@kcl.ac.uk

Abstract. This paper describes an implemented model of context-based incremental tactical generation within the Dynamic Syntax framework [1] which directly reflects dialogue phenomena such as alignment, routinization and shared utterances, problematic for many theoretical and computational approaches [2]. In Dynamic Syntax, both parsing and generation are defined in terms of *actions* on semantic tree structures, allowing these structures to be built in a word-by-word incremental fashion. 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 [2] as the major new challenge facing both linguistic and psycholinguistic theory. Several phenomena common in dialogue pose a significant challenge to (and have received little attention in) theoretical and computational linguistics; amongst them *alignment*, *routinization*, *shared utterances* and various *elliptical* constructions. Alignment describes the way that dialogue participants mirror each other's patterns at many levels (including lexical 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:[1]

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

(2)
Sandy: if, if you try and do enchiladas or
Katriane: Mhm.
Sandy: erm
Katriane: Tacos?
Sandy: tacos.

These are especially problematic for 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]

[1] Examples from the BNC, file KNY (sentences 315–317) and KPJ (555–559).
[2] Although see [3] for an initial DRT-based approach.

The initial hearer B 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 A 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 [1] which allows 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 [1, 4, 5], we show how otherwise problematic elliptical phenomena can be modelled. We then show how alignment and routinization result directly from minimisation of effort on the part of the speaker (implemented as minimisation of lexical search in 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 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), and 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 [6] 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 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.

Projected trees are, in general, simpler than in other frameworks, because adjunct structures (e.g. for relative clause construal) are constructed as paired "linked" structures. Such structures may be constructed in tandem, with evaluation rules then determining that these independent structures, once completed,

[3] For formal details of this approach to quantification see [1] chapter 7.

are compiled together via conjunction.[4] So the overall construction process involves constructing predicate-argument structures, in tree format.

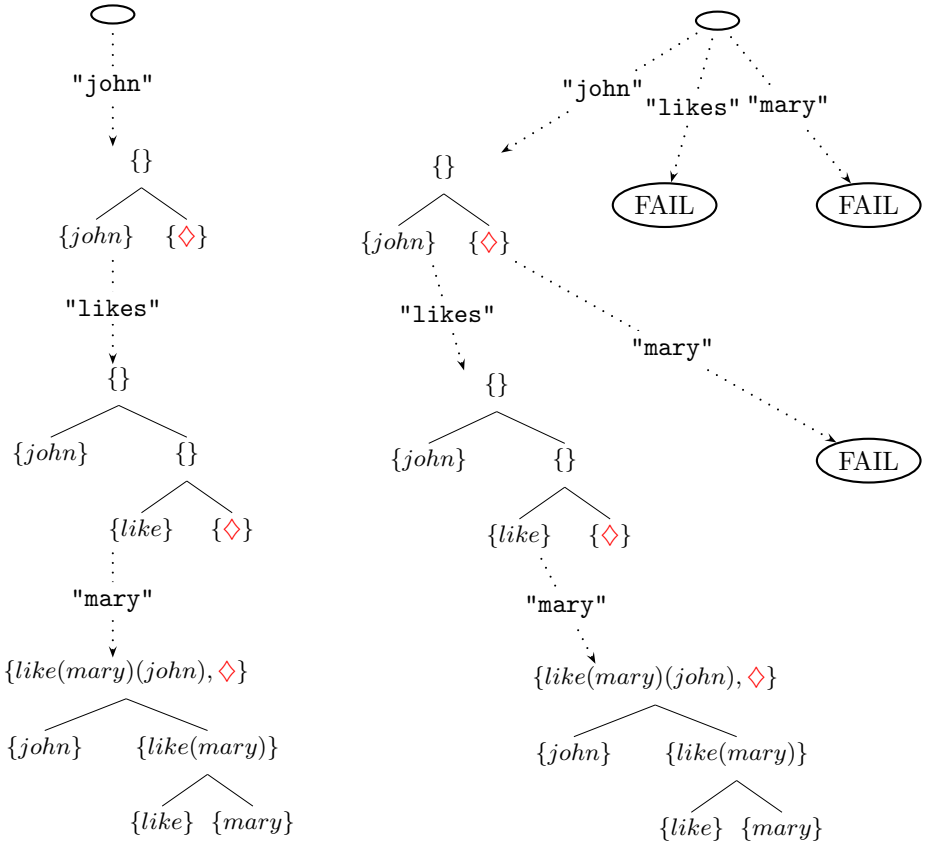
Parsing [1] defines parsing as a process of building labelled semantic trees in a strictly left-to-right, word-by-word incremental fashion by using computational and lexical actions defined (for some natural language) using the modal tree logic LOFT [7]. These actions are defined as transition functions 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 [8]), an underspecification that gets resolved within the left-to-right construction process.[5] 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. For the purposes of the current paper, the important point is that the process is monotonic: the parser state at any point contains all the partial trees which have been produced by the portion of the string so far consumed and which remain candidates for completion.

Generation [4, 5] 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.

[4] The difference between restrictive and nonrestrictive relative construal turns on whether the LINK transition is defined from an epsilon term variable (as in the restrictive *"The man who I like smokes"*) leading to conjunction of the restrictor for the term under construction, or from the constructed term itself (as in *"John, who I like, smokes"*) in which case the result is conjunction of formulae.

[5] In this, the system is also like LFG, modelling long-distance dependency in the same terms as *functional uncertainty* [9], differing from that concept in the dynamics of update internal to the construction of a single tree.

Parsing *"john likes mary"* **Generating *"john likes mary"***



The current model (and implementation) is based on these earlier definitions, but modifies them in several ways, most significantly by the addition of a model of context as described in full in sections 3 and 4; here we briefly describe two other departures. Firstly, we do not adopt the proposal of [4, 5] to speed up generation by use of a restricted multiset of lexical entries (word/action pairs), selected from the lexicon on the basis of goal tree features. Such a strategy assumes a global search of the goal tree prior to generation, preventing subsequent modification or extension, and so is not strictly incremental.

Secondly, the implementation has been extended to allow linked structures as input: the generation of a relative-clause containing sequence *"John, who Sue likes, smokes"*, following the subsumption constraint that the partial tree(s) subsume the goal tree, may involve at any step a partial tree that subsumes a pair of trees, associated with a compound propositional formula $Smoke(John) \wedge Like(John)(Sue)$. Generation of a sentence involving quantification can now also take a goal tree with evaluated formula, so a sentence such as *"A man smokes"* would be generated from a tree whose top node is assigned a formula

$Smoke(\epsilon, x, Man(x))$, evaluated as $Man(a) \wedge Smoke(a)$, where $a = (\epsilon, x, Man(x) \wedge Smoke(x))$. As with relative clauses, given the entailment relation between a conjunctive formula and each of its conjuncts, the subsumption constraint on generation may be met by a partial tree in the sequence of developed trees, despite the compound formula assigned to the goal tree, given that the concept of growth is defined over the parse process leading to such a result.[6]

3 Modelling Context

The basic definitions of parsing and generation [1, 4, 5] assume some notion of context but give no formal model or implementation. For a treatment of dialogue, of course, such a model is essential, and its definition and resulting effects are the subject of this paper. This section defines the model and redefines the parsing and generation processes to include it. Section 4 then describes how the resulting framework allows a treatment of dialogue phenomena.

Context Model NLG systems often assume models of context which include information about both semantic representation and surface strings. [13, 14] both describe models of context which include not only entities and propositions, but also the sentences and phrases associated with them, for purposes of e.g. information structure and subsequent clarificational dialogue. The model of context we require here adds one further element: not only semantic trees (propositional structures) and word strings, but the sequences of lexical actions that have been used to build them. It is the presence of, and associations between, all three that allow our straightforward model of dialogue phenomena, together with the fact that this context is equally available to the parsing and generation processes, as both use the same lexical actions to build the same tree representations.

For the purposes of the current simple implementation, we make a simplifying assumption that the length of context is finite and limited to the immediately previous sentence (although as information that is independently available can be represented in the DS tree format, larger and only partially ordered contexts are no doubt possible in reality): context at any point in either parsing or generation is therefore made up of the trees and word/action sequences obtained in parsing or producing 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, 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 $\langle T_a, \emptyset, \emptyset \rangle$

[6] In building n-tuples of trees corresponding to predicate-argument structures, the system is similar to LTAG formalisms [10]. However, unlike LTAG systems (see e.g. [11]), 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 [12] irrespective of the position in the clausal sequence of the verb.

(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 parser state will again be 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.[7] This triple will now form the new starting context for the next sentence, replacing $\langle T_0, W_0, A_0 \rangle$.

Generation in Context A generator state is now defined as a pair (G, X) of a goal tree 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 – this will be required for our analysis of shared utterances.

In generation, 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 $\langle T, W, A \rangle$. As generation and parsing use the same parsing actions, they make parallel use of context. Thus the generation of *He smiled* in *"John came in. He smiled"* is licensed not simply because the metavariable lexically provided by the pronoun allows the structure induced by the string to (trivially) subsume the goal tree, but because, following the parsing dynamics, a value for this metavariable must be identified from context, and the parse of the previously uttered string provides such a value *john* which (less trivially) allows subsumption. 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 close 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 [15] for justification of this stance).

4 Modelling Dialogue

Anaphora & Ellipsis This model, with its inclusion of action sequences, now allows a full analysis of anaphora and ellipsis. Pronouns and strict readings of VP ellipsis are formally defined as decorating tree nodes with metavariables, to

[7] This formalisation assumes all ambiguity is removed by inference etc. If not, the final parser state, and thus the initial context for the next sentence, will contain more than one triple.

be updated using terms established in context, i.e. by copying a suitably typed semantic formula which decorates some tree node $n \in (T_0 \cup T)$. The analysis of sloppy readings of VP ellipsis instead defines this update as achieved by action re-use: any contextual sequence of actions $(a_1; a_2; \ldots; a_n) \in (A_0 \cup A)$ which causes a suitably typed formula to be derived can be re-used. This allows generation of a range of phenomena, including those which are problematic for other (e.g. abstraction-based) approaches [16], such as cases in which the interpretation of a pronoun to be reconstructed in the elliptical fragment must involve binding not by the subject, but by some term contained within it:

(3) *A:* A policeman who arrested Bill said he was speeding.
B: The policeman who arrested Harry did too.

The actions associated with *said he was speeding* in (3) include the projection of a metavariable associated with the pronoun which is updated from context (in A's utterance, taking *Bill* as antecedent). Re-using the actions in B's utterance allows *Harry* to be selected from the new context as antecedent, leading to the desired sloppy reading.

The incremental nature of the generation process and its ability to start from any state also licenses other forms of ellipsis such as bare fragments as taking a previous structure from context as a starting point (4). Here *wh*-expressions are analysed as particular forms of metavariables, so parsing A's question yields an open formula; B can then begin generation from a resulting partial tree which the fragment updates and completes (rather than having to postulate a separate grammar rule to license fragments as complete utterances):

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

Minimizing Lexicon Search The context model and notion of action re-use now also allows the minimisation of lexical search, as proposed by [4, 5] (though without formal definitions or implementation). At each stage, the generation process must extend the current partial tree using a lexical action, then check for goal tree subsumption. In principle, this is a computationally expensive process: the lexicon must be searched for all possible word/action pairs, the tree extended and the result checked – and this performed at every step i.e. for each word. Any strategy for minimising this task (reflecting the psychological concept of minimizing speaker's effort) will therefore be highly preferred.[8] The apparently high frequency of elliptical constructions is expected as ellipsis minimises lexical lookup by re-use of structure or actions from context; the same can be said for pronouns, as long as they (and their corresponding actions) are assumed to be pre-activated by default; and as suggested by [4, 5], this makes possible a model of alignment.

[8] Even given a more complex model of the lexicon which might avoid searching all possible words (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 task.

Alignment & Routinization Alignment is now characterisable as follows. If there exists some action $a \in (A_0 \cup A)$ which is suitable for extending the current tree, full lexical search can be avoided altogether by re-using a and generating the word w which occupies the corresponding position in the sequence W_0 or W. This results in lexical alignment – w will be repeated rather than choosing an alternative but as yet unused word from the lexicon. Alignment of syntactic structure in which participants mirror syntactic choices (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 [17]) also follows in virtue of the procedural action-based specification of lexical content. A word such as *give* has two possible sequences of lexical actions a' and a'' despite semantic equivalence of output, corresponding to the two alternative forms. A previous use of a particular form will cause either a' or a'' to be present in $(A_0 \cup A)$, and re-use of this action will cause the form to be repeated.[9] This can 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, generating the corresponding word sequence $(w_1; w_2; \ldots; w_n) \in (W_0 \cup W)$. This will result in sequences or phrases being re-used whenever the same sense or reference is to be conveyed, modelling the semantic alignment described by [19], and resulting in what [2] call *routinization* (construction and re-use of word sequences with consistent meanings).

Shared Utterances [4, 5] also suggest that shared utterances as in examples (1) and (2) should be easy to analyse. The definitions of section 3 now provide a formal basis for allowing the switch of speaker/hearer roles as follows, given that the generation and parsing processes can start from any state, and share the same lexical entries, context and semantic tree representations. We take in order transition from hearer to speaker, transition from speaker to hearer.

Transition from Hearer to Speaker: Normally, the generation process begins with the initial generator state as defined above: $(G, \{(\emptyset, P_0)\})$, where P_0 is the standard initial "empty" parser state $\{\langle T_a, \emptyset, \emptyset\rangle\}$. As long as a suitable goal tree 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 G, forms an initial generator state $G, \{(\emptyset, P_t)\}$, from which generation can begin directly. Note that the context does not change between processes.

For generation to begin from this transition state, the new goal tree G must be subsumed by at least one of the partial trees in P_t (i.e. the proposition to

[9] Most frameworks would have to reflect this via activation of syntactic rules, or preferences defined over parallelisms with syntactic trees in context, both problematic. Though lexical alignment effects might be modelled via a context which includes only semantic referents and associated strings (as used by [18] to echo NPs), independent characterisation will be essential to model syntactic effects and routinization.

[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.

be expressed must be subsumed by the incomplete proposition that has been built so far by the parser). Constructing G prior to the generation task will often be a complex process involving inference and/or abduction over context and world/domain knowledge – [3] 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.[11]

Transition from Speaker to Hearer: At transition, the initial speaker B's generator state 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.[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, it is possible for A to continue in a way that differs from B's initial intentions – i.e. that does not match B's initial goal tree. For B to be able to understand such continuations, it is important that the generation process preserves 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 [5].

5 Summary

The close coupling of parsing and generation processes, and in particular their sharing of a suitable model of context, allow shared utterances and various ellipsis and alignment phenomena to be modelled in a straightforward fashion. A prototype system has been implemented in Prolog which reflects the model given here, demonstrating all the above phenomena in simple dialogue sequences.

Acknowledgements

This paper builds on, and is indebted to, earlier work on the DS framework with Wilfried Meyer-Viol and on generation with Masayuki Otsuka. Thanks are also due to the anonymous INLG reviewers. This work was supported by the ESRC (RES-000-22-0355) and (in the case of the second author) the Leverhulme Trust.

[11] Assuming the goal tree as input raises logical-form equivalence problems [20]. Investigation of the degree to which this affects the current approach must be left for further work, but some points are worth noting: the goal tree is semantic, reflecting predicate-argument structure not that of a NL string, so its construction will not require detailed consultation of a grammar; as the language of inference (and tree decoration) is presumed to be the epsilon calculus (not FOL), structural representations will reflect NL structures relatively closely; and we assume that use of context will help determine not only mode of expression but also the goal tree itself.

[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.

References

1. Kempson, R., Meyer-Viol, W., Gabbay, D.: Dynamic Syntax: The Flow of Language Understanding. Blackwell (2001)
2. Pickering, M., Garrod, S.: Toward a mechanistic psychology of dialogue. Behavioral and Brain Sciences **forthcoming** (2004)
3. Poesio, M., Rieser, H.: Coordination in a PTT approach to dialogue. In: Proceedings of the 7th Workshop on the Semantics and Pragmatics of Dialogue (DiaBruck). (2003)
4. Otsuka, M., Purver, M.: Incremental generation by incremental parsing. In: Proceedings of the 6th CLUK Colloquium. (2003)
5. Purver, M., Otsuka, M.: Incremental generation by incremental parsing: Tactical generation in Dynamic Syntax. In: Proceedings of the 9th European Workshop in Natural Language Generation (ENLG-2003). (2003)
6. Meyer-Viol, W.: Instantial Logic. PhD thesis, University of Utrecht (1995)
7. Blackburn, P., Meyer-Viol, W.: Linguistics, logic and finite trees. Bulletin of the IGPL **2** (1994) 3–31
8. Marcus, M.: Deterministic parsing and description theory. In Whitelock, P., Wood, M., Somers, H., Johnson, R., Bennett, P., eds.: Linguistic Theory and Computer Applications. Academic Press (1987) 69–112
9. Kaplan, R., Zaenen, A.: Long-distance dependencies, constituent structure, and functional uncertainty. In Baltin, M., Kroch, A., eds.: Alternative Conceptions of Phrase Structure. University of Chicago Press (1989) 17–42
10. Joshi, A., Kulick, S.: Partial proof trees as building blocks for a categorial grammar. Linguistics and Philosophy **20** (1997) 637–667
11. Stone, M., Doran, C.: Sentence planning as description using tree-adjoining grammar. In: Proceedings of the 35th Annual Meeting of the ACL. (1997) 198–205
12. Ferreira, V.: Is it better to give than to donate? syntactic flexibility in language production. Journal of Memory and Language **35** (1996) 724–755
13. van Deemter, K., Odijk, J.: Context modeling and the generation of spoken discourse. Speech Communication **21** (1997) 101–121
14. Stone, M.: Specifying generation of referring expressions by example. In: Proceedings of the AAAI Spring Symposium on Natural Language Generation in Spoken and Written Dialogue. (2003)
15. Millikan, R.: The Varieties of Meaning: The Jean-Nicod Lectures. MIT Press (2004)
16. Dalrymple, M., Shieber, S., Pereira, F.: Ellipsis and higher-order unification. Linguistics and Philosophy **14** (1991) 399–452
17. Branigan, H., Pickering, M., Cleland, A.: Syntactic co-ordination in dialogue. Cognition **75** (2000) 13–25
18. Lemon, O., Gruenstein, A., Gullett, R., Battle, A., Hiatt, L., Peters, S.: 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. (2003)
19. Garrod, S., Anderson, A.: Saying what you mean in dialogue: A study in conceptual and semantic co-ordination. Cognition **27** (1987) 181–218
20. Shieber, S.: The problem of logical-form equivalence. Computational Linguistics **19** (1993) 179–190