Dialogue as Collaborative Tree Growth

Ruth Kempson and **Masayuki Otsuka**
ruth.kempson@kcl.ac.uk, masayuki.otsuka@kcl.ac.uk
King's College London

Abstract

This paper applies Dynamic Syntax (Kempson et al., 2001) to dialogue modelling, and provides a characterisation of production that is relative to a converse process of tree growth which constitutes a parse check. As evidence for this approach, we place it in a psycho-linguistic perspective, using it to model (i) the parsing/production of elliptical expressions, (ii) dialogue properties such as a speaker-feedback mechanism, speaker/hearer alignments of structure, and speaker/hearer role reversal in shared utterances (Pickering and Garrod, forthcoming).

Very generally in the current research perspective, production and parsing are taken to be independent applications of a neutral centrally available grammar formalism. In abandoning this assumption and articulating a parsing-directed grammar formalism, the Dynamic Syntax framework (Kempson et al., 2001) faces the challenge of articulating the relationship between them. In this paper, we explore the formal mechanisms needed to model generation (as an idealised form of production) given the parsing-directed assumptions of Dynamic Syntax. We then show how the closeness of correlation posited between parsing and production mechanisms allows a natural extension to dialogue modelling that meets the challenge set by (Pickering and Garrod, forthcoming) that grammar formalisms be evaluated relative to their success in capturing key properties of dialogue.

1 Background Assumptions

Dynamic Syntax is a model of NL understanding in which parsing is defined as the progressive projection of a decorated tree structure following the left-right sequence of words in the string. Logical forms are represented as decorated trees, whose topnode is decorated with a formula $Fo(\alpha)$ of type t, and whose dominated nodes are decorated with subterms of the formula α. The central concept is that of goal-directed tree growth, defined using the modal tree logic LOFT (Blackburn and Meyer-Viol, 1994) with basic operators $\langle\downarrow\rangle$, $\langle\uparrow\rangle$ for mother and daughter relations respectively. Tree growth is defined over partial trees, each sequence of such trees starting from the requirement at a rootnode $?Ty(t)$ constituting the overall requirement to establish a logical form of type t at $Tn(0)$ (Tn for treenode). To this node, additional requirements such as $?\langle\downarrow\rangle Ty(e)$, $?\langle\downarrow\rangle Ty(e \rightarrow t)$, are added as subgoals, and lead to the introduction of daughter nodes with requirements $?Ty(e)$, $?Ty(e \rightarrow t)$. Tree nodes are thus created from the root downwards with imposed *requirements* which are subsequently met by tree growth actions dictated by incoming lexical items as they are parsed in a left-right sequence. Processes such as function application are defined in tandem with type-deduction to decorate nonterminal nodes as pairs of terminal nodes are successfully decorated. (At each stage, there is one itemised node under development, indicated by a pointer, $\diamondsuit$.) A complete tree is one which projects a logical form (with a formula of type t decorating its top node). All update processes are monotonic, progressively developing a tree structure meeting the requirements which are imposed on nodes as they are introduced.

In addition to the concept of requirement are other concepts of structural underspecification. Formula values may be underspecified,

Bos, Foster & Matheson (eds): Proceedings of the sixth workshop on the semantics and pragmatics of dialogue (EDILOG 2002), 4-6 September 2002, Edinburgh, UK, Pages 69-76.

eg for anaphoric expressions, which project formulae of the form $Fo(\mathbf{U})$, $\mathbf{U}$ a metavariable. Underspecified tree relations also may be introduced (for left-dislocated expressions), with a treenode identified as dominated by some treenode a without at that point in the treegrowth process any fixed extension (its treenode described as $\langle\uparrow_*\rangle Tn(a)$ with a requirement for a fixed extension, $?\exists x.Tn(x)$). These various forms of underspecification interact in the process of progressive satisfaction of all imposed requirements through computational, lexical and pragmatic actions,[1] each of which constitutes a monotonic step of tree growth ($Ty(e)$ is a development of $?Ty(e)$, $Fo(Mary)$ is a development of $Fo(\mathbf{U})$, $\langle\uparrow\rangle Tn(a)$ is an update of $\langle\uparrow_*\rangle Tn(a)$, and so on). Wellformedness is defined in terms of the result of such actions: a sentence is wellformed if and only if at least one completed tree structure can be derived from a sequence of words, with no requirements outstanding. In using a tree description language, the system is like other parsing formalisms using descriptions of partial trees (Duchier and Gardent, 2001), (Joshi and Kallmeyer, forthcoming), (Koller et al., 2000), or descriptions of trees (Sturt and Crocker, 1996). But unlike them, partial trees are the basis for the grammar formalism, and not merely for semantic characterisations or parsing algorithms relative to an independently defined syntax. The entire concept of syntax is founded in this concept of growth of semantic representations along a left to right dimension without any intermediate and independent syntactic level of representation (Kempson et al., 2001).

Quantification in this system is expressed using the epsilon calculus (Meyer-Viol, 1995) with variable-binding term operators of type e in the style of arbitrary names of natural deduction systems for predicate logic. A sentence such as *A man smokes* is taken to project a logical form $Smoke(\epsilon, x, Man(x))$, derived like other aspects of the interpretation process through incremental construction from lexical specifications which only partially determine the resulting formula. Like [Copestake *et al*, 1999], scope is defined through an incrementally collected set of scope constraints allowing lexical variation; and these scope constraints jointly determine the evaluation of the constructed logical form. The form $Fo(Smoke(\epsilon, x, Man(x)))$, for example, is evaluated relative to a scope statement $S < x$ (S a variable representing the index of evaluation) as $Fo(S : Man(a) \wedge Smoke(a))$, where $a = (\epsilon x, Man(x) \wedge Smoke(x))$. Underspecification of scope determination is thus not expressed through underspecified tree-relations (as in (Erk et al., forthcoming),(Joshi and Kallmeyer, forthcoming)) but through metavariables in scope statements projected, for example, by indefinites. Like all other aspects of underspecification, these must be resolved during the construction process.

[1] A pragmatic process of substitution is presumed to provide the update for the meta-variables lexically projected by anaphoric expressions. Defined as an architecture within which steps of parsing can be articulated, the framework has nothing to say about the pragmatic constraints that determine actual choices.

2 Dynamic Syntax and Production

Assuming this model of parsing, we wish to articulate the relation between parsing and an idealised production model.[2] What we aim to provide is a "tactical" generation system which takes a source tree as input and incrementally "checks" off nodes in this tree as a progressively enriched partial tree can be successfully induced by some selected word. The pointer in this partial parse tree picks out the node whose analogue in the source tree is being "checked", an action which is matched by the selection of some word from the lexicon and "writing" it at the right-hand edge of a sequence of already established words. Such checking action is licensed if and only if the word selected projects a compound parse action which lead to an update of the parse tree from that defined over the words already decided upon, mapping that partial tree onto an update reflecting the annotations on the node being checked. To see the general dynamics, consider a simple tree representing the content of *John saw a woman* with a formula $Fo(S_{PAST} : See(\epsilon, x, Woman(x))(John))$ decorating its rootnode, with accompanying scope statement

[2] This paper does not address the problem of phonological/phonetic levels of realisation and their relationship to parsability of the output.

$S_{PAST} < x$.[3] We assume initially that the starting point for any production task is a full tree as source representing interpretation of the string, and a parse tree made up of a single rootnode with pointer and requirement $?Ty(t)$ (though see section 5 where both inputs to production and parsing are generalised to arbitrary partial trees). Generation steps are then licensed relative to some associated parsing step. Within the source tree, the generating system can, for example, "check off" the subject node, and "generate" the first word in the string, because there is a parsing routine whereby a combination of node-introducing rules operating on the rootnode together with the lexical actions of the name *John* can lead to the successful annotation of the subject node as $Fo(John), Ty(e)$. The search, then, through the lexicon is to find the word which provides these actions. In this paper, this aspect of the search is trivialised, by making the Fodorian assumption that words and concepts correspond one-to-one. However the selection of a pronoun is nontrivial, and is licensed as long as the underspecified input provided is sufficient for the hearer to identify the term intended, given the context provided.[4] Having checked off the node associated with the subject in the source tree in virtue of a successful step in the parse tree annotating the subject node in that tree, the next production step again follows the move of the parsing pointer which is back to the rootnode of the parse tree. From there a following computational step allows the introduction of a predicate-requiring node. Accordingly, the generator attempts to check the content of its predicate node in the source tree. With verbs defined as being of the form $IF\ ?Ty(e \rightarrow t), THEN...$, the lexicon is scanned for a word which will lead to the introduction and decoration of a node with $Fo(see), Ty(e \rightarrow (e \rightarrow t)$. The word *saw* is duly selected, which in addition enables the information about the tense construal also to be checked; and it is written at the right edge of the sequence, yielding *John saw*. Given that the actions of the word *saw* leave the pointer in the parse tree at the object node, that node alone remains to be realised by some sequence of words. The term $Fo(\epsilon, x, Man(x))$ can then be checked off because there is a sub-sequence of actions provided by the determiner and noun that introduce a subtree decorated with $Fo(\lambda P.\epsilon, P)$ and $Fo(x, Man(x))$ for some fresh variable x combining together to yield $Fo(\epsilon, x, Man(x))$. Once the hearer can be presumed to have a tree with all terminal nodes annotated, the nonterminal nodes then have to be taken as checked, given the automatic parse steps of functional-application/type-deduction that would apply to decorate nonterminal nodes in the parsing routine. The pattern is quite general. Generation involves three trees: one a fully annotated tree which forms the input to the process; second a tree with a subset of its nodes checked; and third, a tree which reflects the corresponding parse tree commensurate with establishing the node currently being checked. The provided sequence of actions is by no means unique. Any sequence of actions conforming to the pattern of pairing source tree incrementally with emergent parse tree following the sequence of words is licensed: eg left-peripheral placement of any "dislocated" NP presuming on a parse-sequence projecting an initially unfixed node (see section 4 below, and (Kempson et al., 2001)).

[3]We take S_{PAST} as a shorthand representation of tense, S_{PAST} the representation of the index relative to which the logical form $Fo(See(\epsilon, x, Woman(x))(John))$ is evaluated.

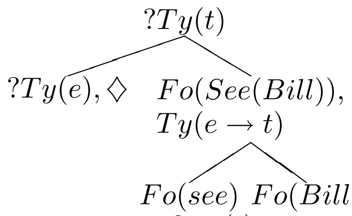
[4]In a fuller account of proper names, these too would require nontrivial identification of the individual being talked about with a lexical characterisation involving a meta-variable requiring substitution. However, we leave all such issues aside.

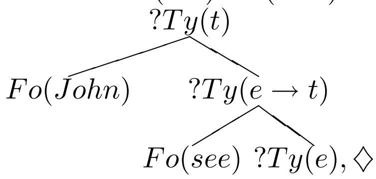
3 Ellipsis as Tree Abstraction

Without independent motivation, such a checking process would be nothing more than stipulation, and indication of the need for a separate production "grammar". The challenge is to reduce this checking system to some operation independently needed for natural language processing. In the interpretation of ellipsis, we find the mechanism we need, this being a tree-abstraction process enabling partial structures to be re-used. For example, consider the parsing of the elliptical fragment in (1), which can be interpreted either as "Harry saw Bill" or as "John saw Harry", the choice being free and determined pragmatically:

(1) John saw Bill. Harry too.

For this fragment, a subtree needs to be constructed from the tree established from the first sentence as the basis for interpreting the fragment *Harry*. This subtree is (1a) or (1b):

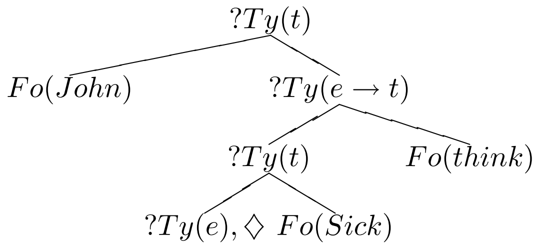
(1a)


(1b)


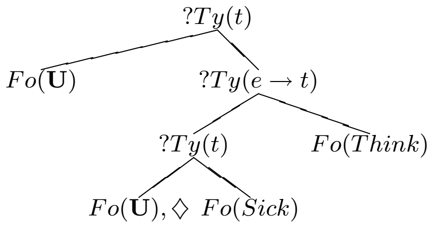
This process takes some decorated tree as input, and replaces some annotated node(s) of that tree with requirements, "abstracting" out established values and in so doing providing a "newly incomplete" tree with which to establish some different completion (notice how such reversal along the process of tree growth at a terminal node automatically applies also to nodes dominating that terminal node).

More generally, we allow a process of tree abstraction back along ANY dimension of tree growth defined over pointed partial trees, returning annotations to requirements ($Ty(e)$ to $?Ty(e)$), Formula values to meta-variables ($Fo(Mary)$ to $Fo(\mathbf{U})$), and so on, to create a partial tree needing completion.[5] We can then define ellipsis construal as the parsing of the expression relative to an arbitrary partial tree such that the result yields a propositional formula as value. This process of abstraction for the processing of a fragment yields (like (Dalrymple et al., 1991)) the parallelism of construal between antecedent structure and interpretation of the fragment characteristic of ellipsis. Sloppy/strict interpretations for ellipsis are expressed as different forms of abstraction or anaphoric construal.[6] For example, the interpretation of the elliptical element as 'John thinks Bill is sick' in (2) requires abstraction of decorations at one node:

(2) John thinks he is sick. Bill too.

(2a)


Replacing one annotated term of type e with $?Ty(e), \diamondsuit$, will allow the parsing of the fragment to provide a new object value and so lead to $Fo(Think(Sick)(Bill))(John)$ decorating a newly annotated root node. The sloppy interpretation of (2) can be characterised as an abstraction that returns $Formula$ values to metavariable values, yielding a tree with $Fo(\mathbf{U})$ decorating both matrix and subordinate subject nodes, an abstraction which upon parsing the fragment will enable $Fo(Think(Sick(Bill))(Bill))$ to be derived:

(2b)


The abstraction process, being free, will include abstraction over only non-terminal nodes. This provides a basis for modelling fragments as in

(3) John read a book. Mary's. Which he enjoyed.

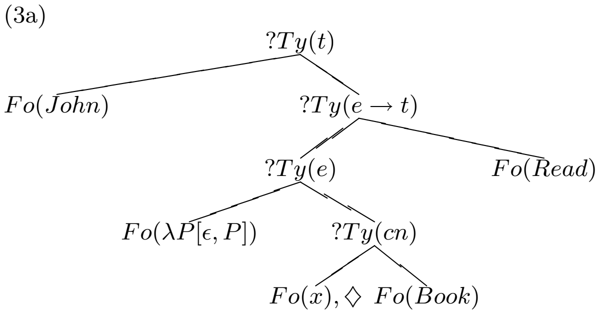
Abstraction of the nonterminal nodes dominating the variable in (3a) allows further extension of the information to be compiled at the dominating nominal node by the building of adjunct structures:[7]

[5]This recovery of a partial tree by abstraction is the structural analogue of the recovery of a suitable relation in the HOU account of ellipsis (Dalrymple et al., 1991). See (Kempson et al., 1998) for arguments that ellipsis construal involves a structural representation of content.

[6]Strict interpretations of the VP pronominal form *do so* involve substitution of the pronominal predicate by the provided antecedent predicate.

[7]In (Kempson et al., 2001), relative clauses and genitives are analysed as projecting quasi-independent linked structures with propositional formulae con-

(3a)



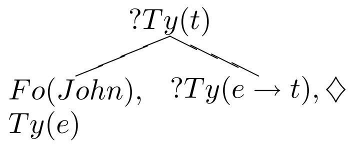
The availability of such abstraction allows elliptical fragments whose construal extends the contextually provided tree, a type of ellipsis that is more problematic for a purely semantic account.

4 The Ellipsis Generation Parallelism

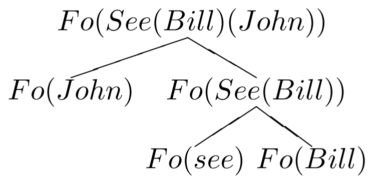
The significance of such abstraction for generation, and the primary focus of this paper, is that it provides a basis for the checking process. Notice that the partial trees defined by this abstraction process need not be those constructed in the course of parsing. (1a), (2a), or (2b) are not intermediate structures in parsing an English string. However, such trees notably correspond to intermediate steps of generation if we model the generation process of "checking" a node as a process of successively abstracting out information from the source tree. On this view, checking of nodes is an emptying process, reversing the growth process by replacing annotations on nodes with weaker decorations, such as the replacement of a formula by $?Ty(X)$, for suitable type. For example (1a) is the tree established when the subject term is the first constituent to be "checked" in uttering *John saw Bill.*

Exploring this further, let us define the checking process as an abstraction from some partial tree replacing it with some partial tree further along a process of tree growth away from a complete tree, eg replacing annotations with requirements. The steps of abstraction licensed in generation are constrained as before to be those for which there is a corresponding parse step from the partial tree associated with a previous generation step to that resulting from the sequence of actions associated with the word selected for linearisation. But now the effect of the constraint is that at each intermediate step, the pair of progressively abstracted partial tree and progressively enriched parse tree unify to yield back the source tree. For example, the linearisation of the word *John* in uttering (1) and the abstraction from source tree to (1a) is licensed, because there is a well-defined parse step from a tree with single node $?Ty(t)$ to the tree (1c), (1a) and (1c) unifying to provide all that is needed to establish the source tree (1d):

(1c)



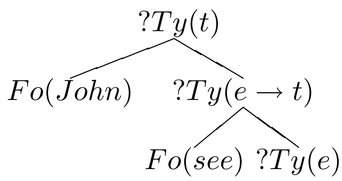
(1d)



Alternative word orders follow the same pattern, the generation of left dislocation sequences also being incremental. Thus the non-subject term in the source tree for (4) is licensed to be linearised first and abstracted from the source tree to yield (4a), in virtue of a corresponding parse step from the start state (4b) to an update (4c) introducing an unfixed node decorated with a non-subject, (4a) and (4c) merging to yield the source tree:

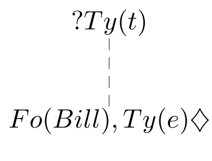
(4) Bill, John saw

(4a)



(4b) $?Ty(t), \diamond$

(4c)



Pronouns, equally, are licensed by this generation process. A node in a tree can be abstracted over and replaced by a requirement, with presentation in the linear sequence of words of a pronoun, as long as the hearer can

---

structed from them that are used to extend the restrictor decorating the $Ty(cn)$ node.

be presumed to have access not only to the lexical actions given by the pronoun (which project a meta-variable as their formula decoration), but also a contextually provided occurrence of some appropriate term with which to substitute that meta-variable.[8]

One advantage of this assumption of the correspondence of the checking process intrinsic to generation and tree abstraction is that checking of nonterminal nodes is now expected. Abstraction of a terminal node that removes a Fo value automatically induces removal of any annotations on dominating nodes which contain that formula as a sub-term.

A second major advantage of this approach is the characterisation it provides of the parsing-production relation while nevertheless maintaining their distinctiveness.[9] Parsing is the development of tree growth from start state ($?Ty(t)$) across intermediate pointed partial trees (T_i) to some completed goal (using Computational, Lexical and Pragmatic actions (C, L, P) involving a string $w_1, ...w_n$), with all subgoals emptied. Generation is the reverse shift from some completed tree as the source tree, with progressive emptying (tree abstraction) as each word $w_1, ...w_n$ is selected until a tree skeleton is reached essentially equivalent to the start state, with all nodes decorated only with requirements. However, generation is not simply a sequence of parsing actions in reverse. Rather there is an interdependence between them – the success of abstraction relies on the parsability of the linearised output. As Figure 1 indicates, the difference between parsing and generation lies in the metalevel aspect to generation. Each step in generating a string with interpretation $Fo(\phi)$ represented as a tree structure T involves correlating some proposed abstracted tree (T_i) and some corresponding parse tree ($T_i{}'$) which must unify to yield back the source tree T. The tightness of this parsing-generation correspondence is due to there being no intermediate system of constraints: the parsing architecture is the grammar formalism, syntactic structure is no more than progressive construction of semantic representation, and production is defined relative to this tree-growth process.

5 Dynamic Syntax as a basis for dialogue

While this articulation of the correspondence between production and parsing remains no more than a blueprint for future research, it notably satisfies desiderata for successful dialogue modelling set out in (Pickering and Garrod, forthcoming), who list as properties of dialogue: widespread alignment between speakers of syntactic, lexical and semantic properties, a large proportion of ellipsis and repetition in dialogue, common occurrence of utterances in which speakers finish each others sentences, and the need in any model for a parsing feedback mechanism. First, Dynamic Syntax assumptions lead one to anticipate maximal alignment of syntactic/semantic/situational levels by eliminating levels other than that of semantic representation. Despite no separate syntactic representational level, even alignment of double object constructions is expected:[10]

(5) R: "I bought Eliot a scooter."
H: "I bought him a skateboard."

Lexical specifications involve projection of concept plus a sequence of tree-update actions; and repetition of a word ensures identity of both concept and sequence of actions employed in uttering/parsing the first occurrence. Verbs such as *give*, which have two discrete sequences of actions, require different lexical specifications to reflect these strategies. In consequence, repetition will ensure recovery of the concept and actions used in the previous

[8] There are many similarities between this account and the system of SPUD (Stone and Doran, 1997), both providing a syntactic and semantic characterisation in which pragmatic steps are integrated. However, SPUD, in using LTAG, is head-driven and assumes a family of elementary trees for each head, a tree for a left-dislocation string being an input variant of a tree reflecting the canonically ordered string. In the DS account, generation of left-dislocation strings is fully incremental.

[9] The lack of simple reversability of parsing and generation actions is wellknown. See (Levelt, 1989).

[10] This distinguishing of discrete sub-specifications of such lexical entries according to the discrete actions they induce will, equally, provide a basis for explaining the oddity of switching from one to another in ellipsis, as in (i):
(i) I gave Eliot a scooter. ??And a skateboard to Bill.

occurrence, whichever that is. Hence 'syntactic' as well as 'semantic' alignment. Nonalignment would involve shifting from one lexical entry to another.

The account automatically yields an incremental self-monotoring device, as the checking of each generation step against the grammar formalism is a self-monitoring parsing step. We expect immediate production breakdown with parsing feedback delay (contra (Levelt, 1989): (Mackay, 1987)).

Dynamic Syntax assumptions also lead one to expect ellipsis to play a large role in dialogue, with generation and construal of ellipsis both defined in terms of a partial tree as context.[11] By the proposed account of generation, utterance of elliptical fragments is licensed because the fragment may be all that is required to enable a hearer to construct a second propositional formula, given the availability of tree abstraction on some preceding tree to provide a partial tree as context which unifies with the partial tree which the fragment provides:

(6) A: I bought a scooter for Bill. A skateboard too.

For example, in (6), relative to the context provided by the tree resulting from processing *I bought a scooter for Bill*, all that is required as input to interpreting the fragment *a skateboard* is to abstract the decoration $Fo(\epsilon x, Scooter(x))$ from the tree provided by the first sentence, replacing it by the requirement $?Ty(e)$ and pointer at the object node, this inducing a shift back at the predicate and root nodes to the requirements $?Ty(e \rightarrow t)$ and $?Ty(t)$. In consequence all that is required to communicate such a structure is to utter the fragment *a skateboard*, for with this utterance, a second propositional formula can be compiled by the hearer. The processing of ellipsis, that is, requires a partial tree as input to the processing task. The generation of ellipsis can accordingly presume on the recovery of such a partial tree by the hearer. Hence its economy to both speaker and hearer.

It is notable that in all cases of parsing, whether complete or incomplete, each step is defined relative to some partial structure as context. Ellipsis differs only in that the input to the interpretation process is generalised from the initial start state to an arbitrary partial tree. Given the ellipsis-generation parallelism, this allows a corresponding generalisation of the generation task. The source tree too can be from an arbitrary partial tree. This provides an appropriate basis for characterising what it means to convey an incomplete thought, thereby incorporating the phenomenon of speaking without a complete structure in mind within the general account of production without special stipulation.

As a final testcase, we show how the account can model role-reversals in collaborative utterances, presented by Pickering and Garrod as a major challenge to current orthodox grammar formalisms: Consider (7) in the context of a discussion of what to buy R's new grandson (H being the father):

(7) R: What shall I give to ...
H: Eliot?

In linearising the partial string, R's generation task starts from a source tree with a metavariable WH as the object argument whose value is to be requested and some constant denoting the person to be bought a present. R initiates the string with the *wh* expression on the grounds that H has a parse strategy for introducing an unfixed node, decorating it with the lexically projected WH metavariable, and merging it later with some appropriate fixed position. H, parsing, constructs such a partial tree, updating it by merging the unfixed node with the object node introduced in parsing the verb *give*. This leaves one further node in the parse tree needing completion to meet the three-argument requirement of *give*. At the juncture of uttering/parsing the indirect object expression, this partial tree is shared by R and H. With R signalling name loss, H provides it from independent knowledge, taking over the generation task, completing the pair of emptied tree (associated with the linearised string) and the completed tree (as constructed interpretation). R, now parsing, already has the partial tree constructed as a check on her own incomplete utterance; so all she has to do is to process the name *Eliot* to complete the partial parse tree. The construction of a tree in parsing some string, and the construction

[11] The account requires anaphora/ellipsis resolution at the level of tactical generation (contra (Dale, 1992)).

of the same tree in producing that string as a constraint on the generation steps licensed ensures the naturalness of the shift from one activity to the other.

Overall, the characterisation of NL syntax in terms of left-to-right growth of a structural representation of content and the characterisation of NL production as a metalevel activity making essential reference to such processing steps, jointly provide the basis for a natural account of dialogue with no stipulation of actions other than those independently required in NL processing.

References

P. Blackburn and W. Meyer-Viol. 1994. Linguistics, logic, and finite trees. *Bulletin of Interest Group of Pure and Applied Logics*, 2:2–39.

R. Dale. 1992. *Generating Referring Expressions.* MIT Pres, Cambridge MA.

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

D. Duchier and C. Gardent. 2001. Tree descriptions, constraints and incrementality. In H. Bunt et al, editor, *Computing Meaning.* Reidel, Dordrecht.

K. Erk, A. Koller, and J. Niehren. forthcoming. Processing underspecified semantic representations in the constraint language for lambda structures. *Research on Language and Computation.*

A. Joshi and L. Kallmeyer. forthcoming. Factoring predicate argument and scope semantics: Underspecified semantics with ltag. *Research on Language and Computation.*

R. Kempson, W. Meyer-Viol, and D. Gabbay. 1998. Vp ellipsis: towards a dynamic, structural account. In S. Lappin and S. Benmamoun, editors, *Fragments: Studies in Ellipsis and Gapping*, pages 227–290. Oxford University Press, Oxford.

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

A. Koller, J. Niehren, and K. Striegnitz. 2000. Relaxing underspecified semantic representations for re-interpretation. volume 3.

W. Levelt. 1989. *Speaking: From Intention to Articulation.* MIT Press, Cambridge, MA.

D. Mackay. 1987. *The Organization of Perception and actions: A theory for Language and Other cognitive Skills.* Springer, New York.

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

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

M. Stone and C. Doran. 1997. Sentence planning as description using *TAG. ACL 1997*, pages 198–207.

P. Sturt and M. Crocker. 1996. Monotonic syntactic processing: a cross-linguistic study of attachment and reanalysis. *Language and Cognitive Processes*, 11:449–94.

Parsing

$START(?Ty(t))$	$\longmapsto^{\{C,L,P\}}$	T_i	$\longmapsto^{\{C,L,P\}}$	$GOAL(Fo(\psi), Ty(t))$
$String(w_1)$		$\langle w_1 \ldots w_i \rangle$		$\langle w_1 \ldots w_n \rangle$

Generation As Tree Abstraction Plus a Parsing Control

$Abstraction:$	$Source(Fo(\phi), Ty(t))$	$\longmapsto^{\{ABSTR\}}$	T_i	$\longmapsto^{\{ABSTR\}}$	$START(?Ty(t))$
$Parsing:$	$START(?Ty(t))$	$\longmapsto^{\{C,L,P\}}$	T_i'	$\longmapsto^{\{C,L,P\}}$	$Source(Fo(\phi), Ty(t))$
$String:$	0		$\langle \ldots w_i \rangle$		$\langle \ldots w_i \ldots w_n \rangle$

Figure 1: **Parsing and Generation**