Conversational Interactions: Capturing Dialogue Dynamics

Arash Eshghi, Julian Hough, Matthew Purver (QMUL, London)
Ruth Kempson, Eleni Gregoromichelaki (KCL London)

May 16, 2012

1. The scope of grammar

In this paper, we set out the case for combining the Type Theory with Records framework (TTR, Cooper (2005)) with Dynamic Syntax (DS, Kempson, Meyer-Viol, and Gabbay (2001); Cann, Kempson, and Marten (2005)) in a single model (DS-TTR). In a nutshell, this fusion captures a phenomenon inexpressible in any direct way by frameworks grounded in orthodox sententialist assumptions – the dynamics of how, in ordinary conversations, we build up information together, incrementally, bit by bit, through half starts, suggested add-ons, possible modifications to the emergent structure which we are apparently collaborating on, all the while allowing that we might be uncertain as to the final outcome, or even in fierce disagreement. To this hybrid, TTR brings representations of content which, through its rich notion of subtyping, allows for highly structured models of both content and context. DS contributes a grammar framework in which syntax is defined as the progressive building of representations of content via update mechanisms following real-time dynamics. Together they provide a framework in which the interactive dynamics of conversational dialogue is an immediate consequence. And the data we present below show that such a model is essential if core syntactic properties of natural language are to be fully captured.

1.1. Incrementality, radical context-dependence and dialogue phenomena

1.1.1. The (non-)autonomy of syntax

Evidence for incrementality in conversation comes from the widespread use of utterances that are fragmentary, subsentential, yet intelligible, all in virtue of ongoing interaction between interlocutors and their physical environment:

(1) Context: Friends of the Earth club meeting
A: So what is that? Is that er... booklet or something?
B: It's a [[book]]
C: [[Book]] (*Answer/Acknowledgement/Completion*)
B: Just ... [[talking about al you know
alternative]] (*Continuation*)
D: [[On erm... renewable yeah]] (*Extension*)
B: energy really I think... (*Completion*)
A: Yeah (*Acknowledgment*) [BNC:D97]

Moreover, the placing of items like inserts, repairs, hesitation markers etc. follows systematic patterns that show subtle interaction with grammatical principles at a sub-sentential level (Levelt 1983; Clark and Fox Tree 2002):

(2) "Sure enough ten minutes later the bell r-the doorbell rang"
(Schegloff, Jefferson, and Sacks 1977)

(3) "I-I mean the-he-they, y'know the guy, the the pathologist, looks at the tissue in the microscope…" *(Schegloff, Jefferson, and Sacks 1977)*

The heart of the incrementality challenge is that people can make perfect sense of and systematically manipulate not only their own sub-sentential utterances as they produce them, but also others'. Even very young children can seamlessly take over from an adult in conversation. Participants may seek to finish what someone else has in mind to say as in (4), but equally, they may interrupt to alter what someone else has proffered, taking the conversation in a different or even contrary direction, as in (5) :

(4) Gardener: I shall need the mattock.
Home-owner: The...
Gardener: mattock. For breaking up clods of earth.[BNC]

(5) (A and B arguing:)
A: In fact what this shows is
B: that you are an idiot

Yet, this phenomenon of *compound contributions* is by no means restricted to one party completing someone else's utterance according to their own sense of the required outcome. Participants may, in some sense, "just keep going" from where their interlocutor had got to, contributing the next little bit. Such exchanges can indeed be indefinitely extended without either contributor knowing in advance the end-point of the exchange:

(6) (a) A: Robin's arriving today
(b) B: from?
(c) A: Sweden
(d) B: with Elisabet?
(e) A: and a dog, a puppy and very bouncy
(f) B: but Robin's allergic
(g) A: to dogs? but it's a Dalmatian.

(h) B: and so?
(j) A: it won't be a problem. No hairs.

The upshot is that it is hard to tell where one sentence stops and the next starts.

This phenomenon is not a dysfluency of dialogue. The forms of such 'fragments' are not random: with only very isolated exceptions, they follow exactly the licensing conditions specified by the NL grammar, with syntactic dependencies of the most fundamental sort holding between the subsentential parts.[1]

(7) A: I'm afraid I burned the buns.
B: Did you burn
A: myself? No, fortunately not.

(8) A: D'you know whether every waitress handed in
B: her taxforms? A: or even any payslips?

People can take over from one another at any arbitrary point in an exchange (Purver et al. 2010), setting up the anticipation of possible dependencies to be fulfilled. We have already seen that it can be between a preposition and its head, (6b-c), between a head and its complement (6f-g), between one conjunct and the next (6d-j), between a reflexive pronoun and its presented antecedent (7), determiner and noun (4), quantifier and expressions it binds (8) etc. So, unless the grammar reflects the possibility of such dependencies to be set and fulfilled across participants, not a single grammatical phenomenon will have successfully been provided with a complete, uniform characterisation. Moreover, any attempt to reflect this type of context-dependence, and the attendant sense of continuity it gives rise to, through grammar-internal specifications will have to involve constraints on fragment construal that go well beyond what is made available in terms of denotational content: such constraints will have to include the full range of syntactic and morphosyntactic dependencies (Ginzburg and Cooper 2004; Ginzburg 2012).

Amongst the proposed solutions to capturing such dependencies is the stipulation of a salient antecedent utterance, whose syntactic characterisation is projected into context and taken to constrain the form of the following fragment (see e.g. Ginzburg's approach (2012)). However, even in the absence of any linguistic antecedent, where the derivation of speech act content is achieved purely pragmatically, such fragments need to respect the morphosyntactic requirements of the relevant NL:

(9) Context: A and B enter a room and see a woman lying on the floor:
A to B: Schnell, den Arzt/*der Arzt [German]
"Quick, the doctor$_{ACC}$ /*the doctor$_{NOM}$" [command]

(10) A is contemplating the space under the mirror while re-arranging the furniture and B brings her a chair:
tin karekla tis mamas?/*i karekla tis mamas? Ise treli?
[Greek] [clarification]
the chair of mum's$_{ACC}$/*the chair$_{NOM}$ of mum's. Are you crazy?

(11) A is handing a brush to B:
A: for painting the wall? [clarification]

(12) A is pointing to Bill:
B: No, his sister [correction]

Thus no account that relies on rules that require reference to some salient linguistic form of antecedent utterance will be general enough (even Ginzburg's invocation of genre-specific scripts does not provide the relevant licensing for such cases). In particular, these data suggest that the grammar needs to be defined as part of a general model of action/perception so that common representations can be retrieved and manipulated both from the linguistic and extra-linguistic context (see e.g. Larsson (2011)). A crucial ingredient in such integration would be licensing mechanisms that operate at a subsentential level with fine-grained sensitivity to the time-linear process of interaction among agents and the evolving context in which such interaction takes place.

1.1.2. Pragmatic/semantic "competence" and radical context-dependence in dialogue

These data are also significant to pragmatists. Almost all pragmatists assume that the supposedly isolatable sentence meaning made available by the grammar should feed into a theory of performance/pragmatics whose burden it is to explain how, relative to context, both full sentences and fragments are uttered on the presumption that the audience will come to understand the propositional content which the speaker has (or could have) in mind. But, contrary to this view, participants understand what each other is saying and switch roles well before any such propositional content could be interpreted

to constitute the object relative to which the speaker or other party could hold a propositional attitude:

(13) Daughter: Oh here dad, a good way to get those corners out
Dad: is to stick yer finger inside.
Daughter: well, that's one way (Lerner 1991)

(14) M: It's generated with a handle and
J: Wound round? [BNC]
M: Yes, wind them round and this should, should generate a charge

There is negotiation here as to the best way to continue a partial structure, with intentions of either party with respect to the resulting content possibly only emerging after the negotiation. Utterances may also be multi-functional, with more than one speech act expressed by a single utterance:

(15) Lawyer: Do you wish your wife to witness your signature, one of your children, or..?
Customer: Joe.

So there is no single proposition/speech act that the individual speaker may have carried out which has to be grasped in order for successful exchanges to have taken place. Participants rely on the setting up of grammatical dependencies which both speaker and hearer are induced to fulfil, so as to perform possibly composite speech acts (Gregoromichelaki et al. forthcoming):

(16) Jim: The Holy Spirit is one who ...gives us?
Unknown: Strength.
Jim: Strength. Yes, indeed. .... The Holy Spirit is one who gives us?
.....
Unknown: Comfort. [BNC HDD: 277-282]

(17) Therapist: What kind of work do you do?
Mother: on food service
Therapist: At ...
Mother: uh post office cafeteria downtown main point office on Redwood
Therapist: Okay (Jones & Beach 1995)

The commitment to recovering any such content as a precondition for successful communication has therefore to be modified; and so too does the presumption of there having to be specific intended propositional plans on the part of the speaker (Grosz and Sidner 1986; Poesio and Rieser 2010; Carberry 1990). Such cases show, in our view, that "fragmentary" interaction in dialogue should be modelled as such, i.e. with grammar defined to provide mechanisms that allow participants to incrementally update the conversational record without at each step requiring reference to some propositional whole. Even though participants can reflect and reify such interactions in explicit propositional terms (Purver et al. 2010), the ongoing metacommunicative interaction observable in dialogue is achievable via the grammatical mechanisms themselves without commitment to deterministic speech-act goals.

The problem current frameworks have in dealing with such data can be traced to the assumption that it is sentential strings that constitute the output of the grammar, over which some propositional content is to be defined, along with the attendant methodological principle debarring any attribute of performance within the grammar-internal characterisation. In this respect, Cooper and colleagues (see e.g. Ginzburg (2012)) have achieved significant advance in defining an explicit semantic model that is not so restricted, exploring ontologies required to define how speech events can cause changes in the mental states of dialogue participants. However, the syntax of that system is defined independently as an HPSG grounded module which precludes a principled modelling of the evolving subsentential (syntactic) context-relativity in these compound contributions with their seamless shifts between parsing and generation. It is within the composite DS-TTR system that their natural modelling emerges, in virtue of both content and context being defined for both parties in the same terms of evolving partial structures.

2. DS-TTR for dialogue modelling

In turning to details of this model, we will need concepts of incrementality applicable to both parsing and generation. Milward's (1991) two key concepts of *strong incremental interpretation* and *incremental representation* apply to semantic incrementality. *Strong incremental interpretation* is the ability to make available the maximal amount of information possible from an unfinished utterance as it is being processed word by word, particularly the se-

mantic dependencies of the informational content (e.g. a representation such as $\lambda x.like'(john', x)$ should be available after processing "John likes"). *Incremental representation*, on the other hand, is defined as a representation being available for each substring of an utterance, but not necessarily including the dependencies between these substrings (e.g. having a representation such as $john'$ attributed to "John" and $\lambda y.\lambda x.like'(y, x)$ attributed to "likes" after processing "John likes"). There are two further concepts of incrementality. In order to model *compound contributions*, the representations produced by parsing and generation should be *interchangeable*, e.g. by defining parsing and generation as employing the same update mechanisms (section 3.1). Finally, the notion of an incrementally constructed and accessible *context* becomes important for modelling self-repair, but also independently motivated for a range of other elliptical phenomena such as stripping and VP-Ellipsis. As we will see in sections 2.2, 3.2 (see also Cann, Kempson, and Purver (2007)), the appropriate concept of context for DS is a *procedural* one since it is by means of conditioned procedures for update that interpretations are incrementally constructed.

2.1. Combining Dynamic Syntax and TTR

DS is in the spirit of Categorial Grammars in directly modelling the building up of interpretations, without presupposing or indeed recognising an independent level of syntactic processing. Thus the output for any given string of words is a purely semantic tree representing its predicate-argument structure; words and grammatical rules correspond to actions which incrementally license the construction of such representations in tree format, employing a modal logic for tree description which provides operators able to introduce constraints on the further development of such trees (LOFT, Blackburn and Meyer-Viol (1994)). However, unlike categorial grammars, it achieves this while also respecting time-linear incrementality, with the left-right progressive build-up of information directly modelled through the incorporation of structural underspecification plus update as a core syntactic device. In particular, analysis of long-distance dependencies and other noncontiguous dependencies are defined in such terms (see Cann, Kempson, and Marten (2005), ch. 2 for details). The DS lexicon consists of *lexical actions* keyed to words. There is also a set of globally applicable *computational actions*. Both constitute packages of monotonic update operations on semantic trees, and take the form of IF-THEN action-like rules which when applied yield semantically

transparent structures. For example, the lexical action corresponding to the word *john* has the preconditions and update operations in (18):

(18) IF ?$Ty(e)$
THEN put($Ty(e)$)
put($\left[\ x : john\ \right]$)
ELSE abort

The trees upon which actions operate represent terms in the typed lambda calculus, with mother-daughter node relations corresponding to semantic predicate-argument structure (see Figure 1 below). The pointer object, $\Diamond$, indicates the node currently under development. In DS-TTR, tree nodes are annotated with a node type (e.g. $Ty(e)$) and semantic formulae in the form of TTR *record types* (Cooper 2005). In this incorporation of TTR into DS (Purver et al. 2010; Purver, Eshghi, and Hough 2011), following Cooper (2005), TTR *record types* consist of fields of the form $\left[\ l : T\ \right]$, containing a unique label l in the record type and its type T; the type of the final field corresponds to the node type of the DS tree at which a record type formula is situated. Functional nodes have node types which correspond to the final field types of argument and functor in the TTR function decorating them. Fields can be *manifest* (i.e. have a singleton type such as $\left[\ l_{=a} : T\ \right]$). Within record types there can be *dependent* fields such as those whose singleton type is a predicate as in $\left[\ p_{=like(x,y)} : t\ \right]$, where x and y are labels in fields preceding it (i.e. are higher up in the graphical representation). Functions from record type to record type in the variant of TTR we use here employ paths, and are of the form $\lambda r : \left[\ l1 : T1\ \right]\left[\ l2_{=r.l1} : T1\ \right]$, an example being the formula at the type $Ty(e_s \rightarrow t)$ node in the trees in Figure 1 below, giving DS-TTR the required functional application capability. Parsing intersperses the testing and application of both lexical actions triggered by input words and the execution of permissible sequences of computational actions, with their updates monotonically constructing and the tree and compiling decorations for its nodes: functor node functions are applied to their sister argument node's formula, with the resulting β-reduced record type added to their mother.[2] Seen in these terms, successful processing sequences are those in which applied actions lead to a tree which is complete (i.e. has no outstanding requirements on any node, and has type $Ty(t)$ at its root node as in Figure (1). Incomplete *partial* structures are maintained in the parse state on a word-by-word basis.

We further adopt an event-based semantics along Davidsonian lines (Davidson 1980). So we include an event node (of type e_s) in the representation: this

allows tense and aspect to be expressed,[3] allowing incremental modification to the the record type on the $Ty(e_s)$ node during parsing and generation after its initial placement in the initial axiom tree.

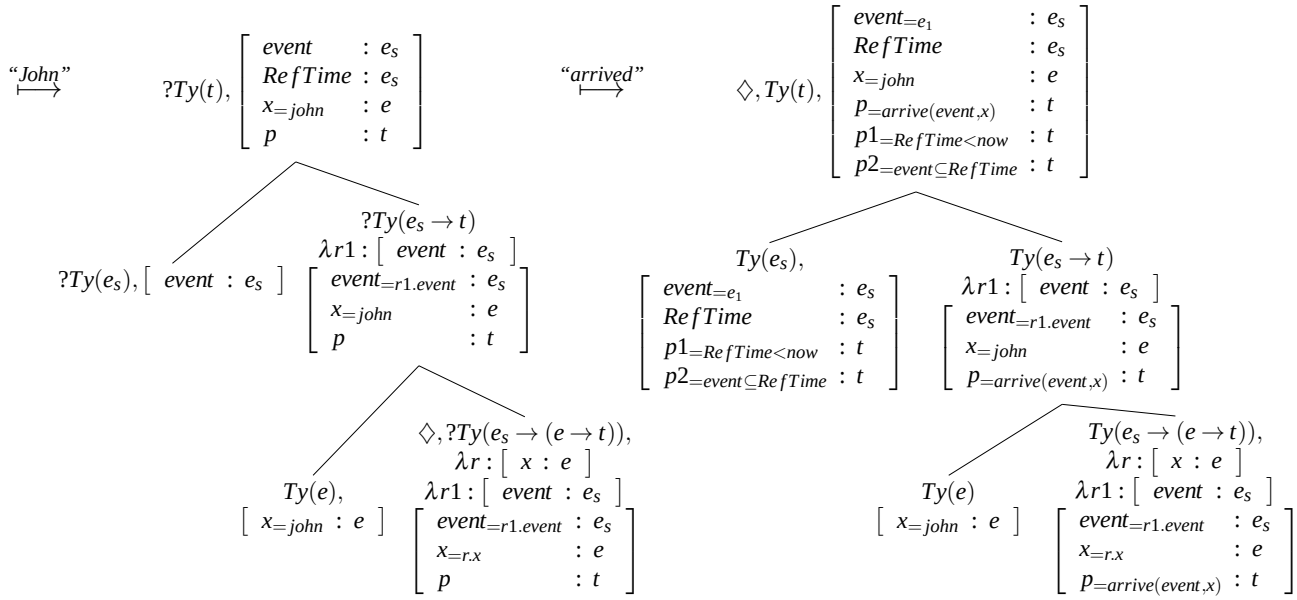

Figure 1. Parsing "John arrived"

This event node specification also permits a straightforward analysis of adjuncts as extensions by the addition of fields from an independently constructed semantic representation (see section 3.1 and Appendix 1 for examples). To achieve this, independent predicate-argument structures are induced via construction of a so-called LINKed tree, an adjunct tree, whose dependency on some host tree despite this structural independence is ensured through a sharing of formula terms at nodes in the two trees in question. A computational action is defined to licensing the appropriate transition from a node of one partial tree to the initiation of this LINKed tree, imposing on its development a dictated co-sharing of terms (see Kempson, Meyer-Viol, and Gabbay (2001)). This device applies to adjunct processing in general (Cann, Kempson, and Marten (2005), ch. 3, also Gregoromichelaki (2006)). In DS-TTR, such LINKs are evaluated as the intersection/concatenation (the *meet* operation, as in Cooper (2005)) of the record-type accumulated at the top of a LINKed tree and the matrix tree's root node record type (see Appendix 1 for

example derivations). So construal of adjuncts boils down to the progressive specification of richer record types.

Through a simple tree compiling algorithm (Hough 2011), the DS-TTR composite now makes available a root record type which gives the maximal amount of semantic information available for partial as well as complete trees (Figure 1). This is achieved by performing all possible functional applications from functor nodes to argument nodes, using underspecified record types as necessary for nodes which have not yet been decorated with semantic content (see e.g. the $Ty(e \rightarrow (e_s \rightarrow t))$ node on the left tree in Figure 1 above, where the functional type corresponding to an upcoming verb does not yet contain an overt predicate to be applied to the subject $john'$, this being simply the *unmanifest/underspecified* field $p : t$).

This root record type compilation via functional application and type intersection meets the requirement of strong incrementality of interpretation, only implicit in DS, as now maximal record types become available as each word is processed. Yet the LOFT underpinning to the mechanisms of tree-growth means that the DS insight that core syntactic restrictions emerge as immediate consequences of the LOFT-defined tree-growth dynamics is preserved without modification (Cann, Kempson, and Marten 2005; Cann, Kempson, and Purver 2007; Kempson and Kiaer 2010; Kempson, Gregoromichelaki, and (eds.) 2011; Chatzikyriakidis and Kempson 2011).

2.2. DS-TTR procedural context as a graph

Aside from the strong incremental interpretation that DS-TTR representations afford, the model provides incremental access to *procedural context* as required not only for modelling the phenomena reviewed above, but independently motivated for phenomena such as VP-Ellipsis and stripping. In DS, this context is taken as including not only the end product of parsing or generating an utterance (the semantic tree and corresponding string), but also information about the dynamics of the parsing process itself – the lexical and computational action sequence used to build the tree (Cann, Kempson, and Purver 2007). This procedural context is modelled as a Directed Acyclic Graph (DAG) (a representation originally used to characterise the parsing process (Sato 2011)), in which edges correspond to DS actions and nodes to (partial) *trees* (Purver, Eshghi, and Hough 2011). This model now satisfies the criterion of *strong incremental representation*: we get a transparent represen-

tation of not only the maximal interpretation for the utterance so far, but also for which sub-utterances contributed which sub-parts of this interpretation. Aside from our model of self-repair set out below, this is required for modelling clarification as well as confirmation behaviour in dialogue. This context DAG can be tightly coupled with a word hypothesis graph (or "word lattice") as obtained from a standard speech recogniser, resulting in ease of integration in modern incremental dialogue systems (Purver, Eshghi, and Hough (2011)).

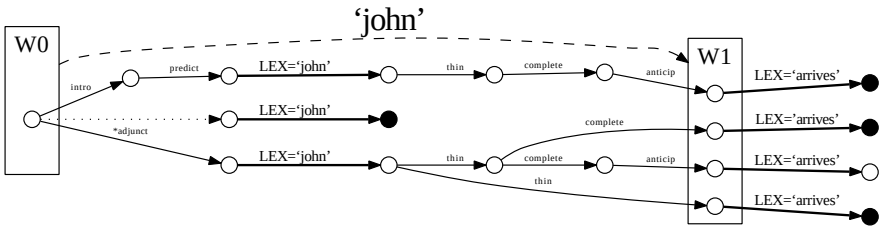

Figure 2. DS context as DAG, consisting of a parse DAG (circular nodes=trees, solid edges=lexical(bold) and computational actions) subsumed by the corresponding word DAG (rectangular nodes=tree sets, dotted edges=word hypotheses) with word hypothesis 'john' spanning tree sets W0 and W1.

The resulting model of context is thus a hierarchical model with DAGs at two levels (figure 2). At the action level, the parse graph DAG (shown in the lower half of figure 2 with solid edges and circular nodes) contains detailed information about the actions (both lexical and computational) used in the parsing or generation process: edges corresponding to these actions are connected to nodes representing the partial trees built by them, and a path through the DAG corresponds to the action sequence for any given tree. At the word level, the word hypothesis DAG (shown at the top of figure 2 with dotted edges and rectangular nodes) connects the words to these action sequences: edges in this DAG correspond to words, and nodes correspond to sets of parse DAG nodes (and therefore sets of hypothesized trees). For any partial tree, the context (the words, actions and preceding partial trees involved in producing it) is now available from the paths back to the root in the word and parse DAGs. Moreover, the sets of trees and actions associated with any word or word subsequence are now directly available as that part of the parse DAG spanned by the required word DAG edges. This, of course, means that the contribution of any word or phrase can be directly obtained,

fulfilling the criterion of incremental representation.

2.3. DS-TTR Generation as Parsing

The goal for the generation module must then, equally, reflect the incremental behaviour that yields confirmations as in (16), (14), continuations as in (4), (16), user interruptions without discarding the semantic content built up so far to provide for realistic clarification and *self-repair* capability such as in (2), (3) and possibly the presumption that the fragment may contribute more than one such attribute as in (15). The same requirements for parsing apply also to generation, viz: *strong incremental interpretation*; *incremental representation* on a word-by-word basis; continual access to *procedural context* to implement all information made available by selected expressions without delay. As noted above, there is the extra requirement in generation of *representational interchangeability* enabling the switch between parsing and production activities. DS-TTR can meet these criteria elegantly in virtue of the DS decision to model generation in terms of the same tree-growth mechanisms as in parsing (Purver and Kempson 2004) with the simple addition of a *subsumption check* against a so-called *goal tree* (but see below for how in DS-TTR this has been replaced with TTR goal concepts).[4] The DS generation process is thus made word-by-word incremental with maximal tree representations continually available, effectively combining lexical selection and linearisation into a single action due to word-by-word iteration through the lexicon.

While no formal model of self-repair was proposed in DS (but see section 3.2), self-monitoring is inherently part of the generation process, as each word generated is parsed. Notwithstanding the degree of incrementality so achieved, the Purver and Kempson (2004) model of generation did not meet the criterion of *strict incremental* interpretation, as maximal information about the dependencies between the semantic formulae in the tree did not need to be computed until the tree is complete. On the other hand, the goal tree needs to be constructed from the grammar's actions, so any dialogue management module must have full knowledge of the DS parsing mechanism and lexicon, and so interchangeability of representation becomes difficult. In moving to the DS-TTR framework, several adjustments were therefore incorporated.

2.3.1. TTR goal concepts and subtype checking

One straightforward modification to the DS generation model enabling representational interchangeability is to replace the previously defined *goal tree* with a *TTR goal concept* which takes the form of a record type e.g.:

(19)
$$\begin{bmatrix} event_{=e1} & : & e_s \\ RefTime & : & e_s \\ p1_{=today(RefTime)} & : & t \\ p2_{=RefTime \bigcirc event} & : & t \\ x1_{=Sweden} & : & e \\ p3_{=from(event,x1)} & : & t \\ x_{=robin} & : & e \\ p_{=arrive(event,x)} & : & t \end{bmatrix}$$

The goal concept may be *partial* as required for such data as (1)-(4), and the dialogue manager may further specify it, but even then it need not correspond to a complete sentence in incremental dialogue management strategies (Guhe 2007; Buß and Schlangen 2011). This move also means a dialogue manager may input goal concepts directly to the generator; and no considerations of the requirements of the DS grammar are needed (contra Purver and Kempson's (2004) approach). The tree subsumption check in the original DS generation model can now be characterised as a TTR subtype relation check (see p.96, Fernández (2006)) between the goal concept record type and the current parse state's root record type.

Figure 3 displays a successful generation path,[5] where the incremental generation of "john arrives" succeeds as the successful lexical action applications at transitions $\boxed{1} \mapsto \boxed{2}$ and $\boxed{3} \mapsto \boxed{4}$ are interspersed with applicable computational action sequences at transitions $\boxed{0} \mapsto \boxed{1}$ and $\boxed{2} \mapsto \boxed{3}$, at each stage passing the subtype relation check with the goal (i.e. the goal is a subtype of the top node's compiled record type), until arriving at a tree that *type matches* the assigned goal concept in $\boxed{4}$ in the rich TTR sense of *type*. In implementational terms, there will in fact be multiple generation paths in the generation state, including incomplete and abandoned paths, which can be incoporated into the DS notion of context as a DAG.

Another advantage of working with TTR record types rather than trees during generation is that selecting relevant lexical actions from the lexicon can take place before generation begins through comparing the semantic formulae of the actions to the goal concept. Subtype checking makes it possible

to reduce the computational complexity of lexical search through a pre-verbal lexical action selection.

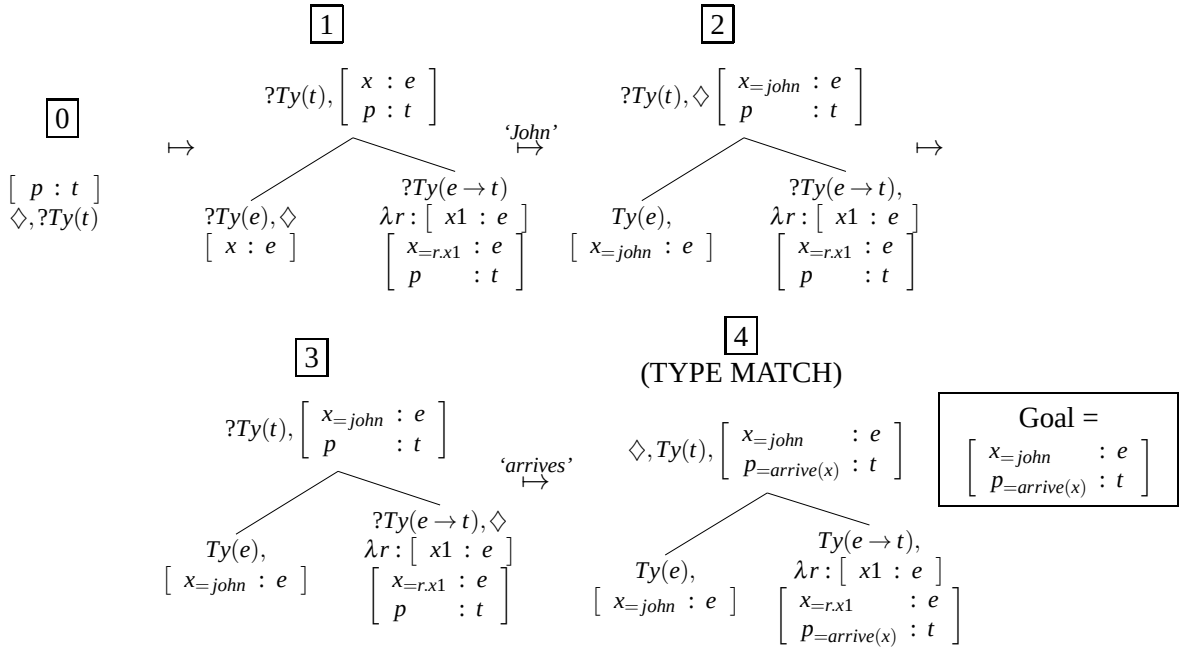

Figure 3. Successful generation path in DS-TTR

3. Incremental processing of dialogue phenomena

We can now see how the resulting DS-TTR model deals with compound contributions; this has been implemented in the publicly available DyLan dialogue system[6] (Eshghi, Purver, and Hough 2011; Purver, Eshghi, and Hough 2011).

3.1. Compound contributions

Previous formal and computational accounts of compound contributions (CCs) have focussed on a sub-category of CCs, so-called *completions*, where a responder succeeds in projecting a string the initial speaker had intended to convey. The foremost implementation is that of Poesio and Rieser (2010), using the PTT model for incremental dialogue interpretation (Poesio and Traum 1997; Poesio and Rieser 2003) in combination with LTAG (Demberg and

Keller 2008). The approach is grammar-based, incorporating syntactic, semantic and pragmatic information via the lexicalised TAG, providing an account of the incremental interpretation process incorporating lexical, syntactic and semantic information.[7] This model meets many of the criteria defined here. Both interpretation and representation are incremental, with semantic and syntactic information being present; the use of PTT suggests that linguistic context can be incorporated suitably. However, while reversibility might be incorporated by choice of suitable parsing and generation frameworks, this is not made explicit; and the extendability of the representations seems limited by TAG's approach to adjunction. The use of TAG also restricts the grammar to licensing grammatical *strings*, problematic for some CCs (e.g. examples (7) in which *semantic* dependencies hold between the two parts of the CC); and the mechanism may not be sustainable for all compound contributions where participants make no attempt to match what the other party might have in mind. So the account is at best incomplete.[8]

The broad range of CCs follows as an immediate consequence of DS-TTR. The use of TTR record types removes the need for grammar-specific parameters; and the interchangeability of representations between parsing and generation means that the construction of a data structure can become a collaborative process between dialogue participants, permitting a range of varied user input behaviour and flexible system responses. This use of the same representations by parsing and generation guarantees the ability to begin parsing from the end-point of any generation process, even mid-utterance; and to begin generation from the end-point of any parsing process. Successive sequential exchanges between participants leading to a collaboratively completed utterance as in (6) are directly predicted. Both parsing and generation models are now characterised entirely by the parse context DAG with the addition for generation of a TTR goal concept. The transition from generation to parsing becomes almost trivial: the parsing process can continue from the final node(s) of the generation DAG, with parsing actions extending the trees available in the final node set as normal. Transition from parsing to generation also requires no change of representation with the DAG produced by parsing acting as the initial structure for generation, though we require the addition of a goal concept to drive the generation process. The same record types are thus used throughout the system: as the concepts for generating system plans, as the goal concepts in NLG, and for matching user input against known concepts in suggesting continuations. Possible system transition points trigger alternation between modules in their co-construction

of the shared parse/generator.A goal concept can be produced by the dialogue manager at a speaker transition by searching its domain concepts for a suitable subtype of the TTR record type built so far, guaranteeing a grammatical continuation given the presence of appropriate lexical actions. This extends the method for CC modelling described in (Purver and Kempson 2004): now the dialogue manager has an elegant decision mechanism for aiding content selection. And, given the presumption of context, content and goal specifications all in terms of record types, the ability to construct goals in a scenario without linguistic antecedents as in (9) and (10).

The data of CCs thus follows in full, even when either the goal record type for the interrupter does not match that of the initiator as in (5), or when the goal record type does not correspond to a complete domain concept, as in the successive fragment exchanges such as (6).This is achieved through progressive extensions of the partial tree so far, either directly, or by adding adjunct LINKed trees. This results in the word-by-word further specification of the record type at the root of the matrix tree representing the maximal interpretation of the string/utterance so far. In Figure 4 we give the progressive record-type specification for the exchange (20), a simplification of (6), showing how incomplete structures may serve as both input and output for either party:

(20) A: Today Robin arrives B: from A: Sweden

Details of the tree derivations are omitted in Figure 4, but we have included these in Appendix 1, which contains a fuller tree derivation for (20). As noted,

$$\begin{bmatrix} event & : e_s \\ RefTime & : e_s \\ p1_{=today(RefTime)} & : t \\ p & : t \end{bmatrix} \mapsto \begin{bmatrix} event_{=e1} & : e_s \\ RefTime & : e_s \\ p1_{=today(RefTime)} & : t \\ p2_{=RefTime \bigcirc event} & : t \\ x_{=robin} & : e \\ p_{=arrive(event,x)} & : t \end{bmatrix} \mapsto \begin{bmatrix} event_{=e1} & : e_s \\ RefTime & : e_s \\ p1_{=today(RefTime)} & : t \\ p2_{=RefTime \bigcirc event} & : t \\ x_{=robin} & : e \\ p_{=arrive(event,x)} & : t \\ x1 & : e \\ p3_{=from(event,x1)} & : t \end{bmatrix} \mapsto \begin{bmatrix} event_{=e1} & : e_s \\ RefTime & : e_s \\ p1_{=today(RefTime)} & : t \\ p2_{=RefTime \bigcirc event} & : t \\ x_{=robin} & : e \\ p_{=arrive(event,x)} & : t \\ x1_{=Sweden} & : e \\ p3_{=from(event,x1)} & : t \end{bmatrix}$$

"A: Today" $\mapsto$ *"..Robin arrives"* $\mapsto$ *"B: from?"* $\mapsto$ *"A: Sweden"*

Figure 4. Incremental interpretation via TTR subtypes

more complex forms can be generated by incorporating LINKed trees, as is presumed in the characterisation of the many extensions by the addition of an adjunct, as in (8) (See Appendix 1), without any of these having to involve any extension of the formal DS vocabulary.

3.2. Self-repair

In this section, we present our initial model of self-repair. Specifically, there are two types of repair that we address here: type 1, where the repair involves a local, and partial restart of the reparandum, as in (2) and type 2 where the repair is simply a local extension, i.e. a further specification of the reparandum as in (3).

In our DS-TTR model of generation set out above, a type 1 repair arises due to an online revision of a record type goal concept, whereby the new goal concept is not a sub-type of the one the speaker had initially set out to realise. We model this via backtracking along the incrementally available context DAG as set out above. More specifically, repair is invoked if there is no possible DAG extension after the semantic filtering stage of generation (resulting in no candidate succeeding word edge). The repair procedure proceeds by restarting generation from the last realised (generated) word edge. It continues backtracking by one DAG vertex at a time until the root record type of the current partial tree is a subtype of the new goal concept. Generation then proceeds as usual by extending the DAG from that vertex. The word edges backtracked over are not removed, but are simply marked as repaired, following the principle that the revision process is on the public conversational record and hence should still be accessible for later anaphoric reference (see Figure 5).

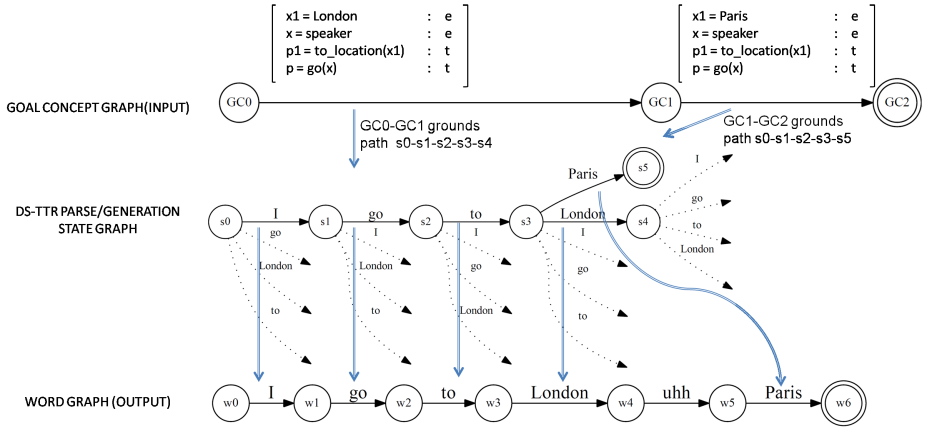

Figure 5. Incremental DS-TTR generation of a self-repair upon change of goal concept. Type-matched record types are double-circled nodes and edges indicating failed paths are dotted.

Our protocol is consistent with Shriberg and Stolcke's (1998) empirical observation that the probability of retracing N words back in an utterance is more likely than retracing from N+1 words back, making the repair as local as possible. Utterances such as "I go, uhh, leave from Paris" are generated incrementally, as the repair is integrated with the semantics of the part of the utterance before the repair point, maximising re-use of existing semantic structure.

Type 2 repairs on the other hand, i.e. *extensions*, where the repair effects an "after-thought", usually in transition relevance places in dialogue after apparently complete turns, are also dealt with straightforwardly by our model. The DS-TTR parser simply treats these as monotonic extension of the matrix tree through LINK Adjunction to it (see Cann, Kempson, and Marten (2005), but also Appendix 1 for an example of such extensions) resulting in subtype extension of the root TTR record type. Thus, a change in goal concept during generation will not always put demands on the system to backtrack, such as in generating the fragment after the pause in "I go to Paris . . . from London". Backtracking only operates at a semantics-syntax mismatch where the revised goal concept is no longer a subtype of the root record type for the (sub-)utterance so far realised, as in Figure 5.

Unlike string-based *speech plan* approaches such as that of Skantze and Hjalmarsson (2010), there is no need here to regenerate a fully-formed string from a revised goal concept and compare it with the string generated thus far to characterise repair. Instead, repair is driven by attempting to extend existing parse paths to construct the new target record type, *retaining* the semantic representation and the procedural context of actions already built up in the generation process to avoid the computational demand of constructing syntactic structures from afresh where possible. Also, importantly, unlike string-based approaches which are bound to be very domain specific, we note that our approach is completely domain-general.

3.3. Speech Acts and speaker/hearer attributions in DS-TTR

A further bonus of combining DS mechanisms with TTR record types as output decorations is the allowance of a much richer vocabulary for such decorations, as empirically warranted. In particular, it provides a basis from which speaker and hearer attributes may be optionally specified. In this connection, (Purver et al. 2010) propose a specification of fields with sub-field specifications, one a *context* sub-field for speaker-hearer attributions and mi-

cro utterance events, and the second, *content*, for familiar lambda-terms, a modification which allows a record of speaker-hearer attributions to be optionally kept alongside function-argument content record type specifications so that the different anaphor-dependency resolutions across switch of participant roles can be modelled as in (7)-(8) without disturbing content compilation of the lambda terms:

$$\left[\begin{array}{ll} \mathit{ctxt}: & \left[\begin{array}{ll} a_{=\mathit{Arash}} & : e \\ r_{=\mathit{Ruth}} & : e \\ u_0 & : \mathit{utt}-\mathit{event} \\ s0_{=\mathit{spkr}(u_0,a)} & : t \\ u_1 & : \mathit{utt}-\mathit{event} \\ s1_{=\mathit{spkr}(u_1,r)} & : t \end{array}\right] \\ \mathit{cont}: & \left[\begin{array}{ll} x_{=\mathit{robin}} & : e \\ p_{=\mathit{arrive}(x)} & : t \end{array}\right] \end{array}\right]$$

$$\mathit{Ty}(e), \qquad\qquad \mathit{Ty}(e \to t)$$

$$\left[\begin{array}{ll} \mathit{ctxt}: & \left[\begin{array}{ll} u_0 & : \mathit{utt}-\mathit{event} \\ a_{=\mathit{Arash}} & : e \\ s0_{=\mathit{spkr}(u_0,a)} & : t \end{array}\right] \\ \mathit{cont}: & \left[\ x_{=\mathit{robin}} : e\ \right] \end{array}\right] \qquad \left[\begin{array}{ll} \mathit{ctxt}: & \left[\begin{array}{ll} u_1 & : \mathit{utt}-\mathit{event} \\ r_{=\mathit{Ruth}} & : e \\ s1_{=\mathit{spkr}(u_1,r)} & : t \end{array}\right] \\ \mathit{cont}: & \lambda r1 : \left[\ \mathit{cont} : \left[\ x : e\ \right]\ \right] \left[\begin{array}{ll} x_{=r1.\mathit{cont}.x} & : e \\ p_{=\mathit{arrive}(x)} & : t \end{array}\right] \end{array}\right]$$

Figure 6. Processing 'Arash: Robin.. Ruth: ..arrived', with micro utterance events and speaker/hearer attributions, adapted from Purver et al. (2010)

With intersection of record types available for record types of arbitrary complexity, such specifications are unproblematic. As Purver et al. (2010) demonstrate, speech act content can also be derived optionally as a later step of inference over such structures by addition of LINKed trees (see *ibid.* for details). We note, nevertheless, that this isn't essential for an explanation of the interactional patterns observable in conversation, even meta-communicative interaction. Instead, we suggest, conversational interaction is buttressed by mechanisms intrinsic to grammar itself, as we have set out. This of course raises issues of what constitutes successful communication, in particular for Gricean and neo-Gricean models in which recognition of the content of the speaker's intentions is essential: Poesio and Rieser (2010) are illustrative. We do not enter into this debate here, but merely note that this stance is commensurate with the data of section 1 in which participants' intentions may only be emergent or be subject to modification during the course of a conversation without jeopardising its success (Gregoromichelaki et al. 2011; Gre-

goromichelaki et al. forthcoming).

4. Conclusion

We have presented a formal framework for modelling conversational dialogue with parsing and generation modules as controlled by a dialogue manager, both of which reflect word by word incrementality, using a hybrid of Dynamic Syntax and Type Theory with Records. The composite framework allows access to record types incrementally during generation, providing strict incremental representation and interpretation for substrings of utterances that can be accessed by existing dialogue managers, parsers and generators equally, allowing the articulation of syntactic and semantic dependencies across parser and generator modules. Several avenues of research now open up. But most important of all, there is a radical shift of perspective, with the defined "competence" model now securely grounded in its articulation of mechanisms for interactive language performance that it makes possible. And with this move, the nesting of the language faculty into a coherent cognitive system at last becomes possible, opening up radical new perspectives on philosophy of language, psychology and cognition.

Blackburn, Patrick, and Wilfried Meyer-Viol
1994 Linguistics, logic and finite trees. *Logic Journal of the Interest Group of Pure and Applied Logics* 2 (1): 3–29.
Buß, Okko, and David Schlangen
2011 Dium : An incremental dialogue manager that can produce self-corrections. *Proceedings of SemDial 2011 (Los Angelogue), Los Angeles, CA*. 47–54.
Cann, Ronnie
2011 Towards an account of the english auxiliary system: building interpretations incrementally. In *Dynamics of Lexical Interfaces*, Ruth Kempson, Eleni Gregoromichelaki, and Christine Howes (eds.). Chicago: CSLI Press.
Cann, Ronnie, Ruth Kempson, and Lutz Marten
2005 *The Dynamics of Language*. Oxford: Elsevier.
Cann, Ronnie, Ruth Kempson, and Matthew Purver
2007 Context and well-formedness: the dynamics of ellipsis. *Research on Language and Computation* 5 (3): 333–358.
Carberry, S.
1990 *Plan recognition in natural language dialogue*. the MIT Press.
Chatzikyriakidis, Stergios, and Ruth Kempson
2011 Standard modern and pontic greek person restrictions: A feature-free dynamic account. *Journal of Greek Lingusitics*, pp. 127–166.
Clark, Herbert H., and Jean E. Fox Tree
2002 Using *uh* and *um* in spontaneous speaking. *Cognition* 84 (1): 73–111.

Cooper, Robin
2005 Records and record types in semantic theory. *Journal of Logic and Computation* 15 (2): 99–112.
Davidson, Donald
1980 *Essays on Actions and Events*. Oxford, UK: Clarendon Press.
Demberg, V., and F. Keller
2008 A psycholinguistically motivated version of tag. *Proceedings of the International Workshop on Tree Adjoining Grammars*.
Eshghi, A., M. Purver, and Julian Hough
2011 Dylan: Parser for dynamic syntax. Technical Report, Queen Mary University of London.
Fernández, Raquel
2006 Non-sentential utterances in dialogue: Classification, resolution and use. Ph.D. diss., King's College London, University of London.
Ginzburg, Jonathan
2012 *The Interactive Stance: Meaning for Conversations*. Oxford University Press.
Ginzburg, Jonathan, and Robin Cooper
2004 Clarification, ellipsis, and the nature of contextual updates in dialogue. *Linguistics and Philosophy* 27 (3): 297–365.
Gregoromichelaki, E.
2006 Conditionals: A dynamic syntax account. Ph.D. diss., King's College London.
Gregoromichelaki, E., R. Cann, and R. Kempson
forthcoming On coordination in dialogue: subsentential talk and its implications. In *On Brevity*, Laurence Goldstein (ed.). OUP.
Gregoromichelaki, Eleni, Ruth Kempson, Matthew Purver, Greg J. Mills, Ronnie Cann, Wilfried Meyer-Viol, and Pat G. T. Healey
2011 Incrementality and intention-recognition in utterance processing. *Dialogue and Discourse* 2 (1): 199–233.
Grosz, Barbara J., and Candace L. Sidner
1986 Attention, intentions, and the structure of discourse. *Computational Linguistics* 12 (3): 175–204.
Guhe, Markus
2007 *Incremental Conceptualization for Language Production*. NJ: Lawrence Erlbaum Associates.
Hough, Julian
2011 Incremental semantics driven natural language generation with self-repairing capability. *Recent Advances in Natural Language Processing (RANLP)*. Hissar, Bulgaria, 79–84.
Kempson, R., and J. Kiaer
2010 Multiple long-distance scrambling: Syntax as reflections of processing. *Journal of Linguistics* 46 (01): 127–192.
Kempson, Ruth, Eleni Gregoromichelaki, and Christine Howes (eds.)
2011 *The Dynamics of Lexical Interfaces*. CSLI - Studies in Constraint Based Lexicalism.
Kempson, Ruth, Wilfried Meyer-Viol, and Dov Gabbay
2001 *Dynamic Syntax: The Flow of Language Understanding*. Blackwell.

Larsson, Staffan
2011 The TTR perceptron: Dynamic perceptual meanings and semantic coordination. *Proceedings of the 15th Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2011 - Los Angelogue)*. 140–148.
Levelt, W.J.M.
1983 Monitoring and self-repair in speech. *Cognition* 14 (1): 41–104.
Milward, David
1991 Axiomatic grammar, non-constituent coordination and incremental interpretation. Ph.D. diss., University of Cambridge.
Poesio, Massimo, and Hannes Rieser
2003 Coordination in a PTT approach to dialogue. *Proceedings of the 7th Workshop on the Semantics and Pragmatics of Dialogue (SEMDIAL)*. Saarbrücken, Germany.
2010 Completions, coordination, and alignment in dialogue. *Dialogue and Discourse* 1: 1–89.
Poesio, Massimo, and David Traum
1997 Conversational actions and discourse situations. *Computational Intelligence* 13, no. 3.
Purver, Matthew, Arash Eshghi, and Julian Hough
2011 Incremental semantic construction in a dialogue system. J. Bos, and S. Pulman (eds.), *Proceedings of the 9th International Conference on Computational Semantics*. Oxford, UK, 365–369.
Purver, Matthew, Eleni Gregoromichelaki, Wilfried Meyer-Viol, and Ronnie Cann
2010 Splitting the 'I's and crossing the 'You's: Context, speech acts and grammar. P. Łupkowski, and M. Purver (eds.), *Aspects of Semantics and Pragmatics of Dialogue. SemDial 2010, 14th Workshop on the Semantics and Pragmatics of Dialogue*. Poznań: Polish Society for Cognitive Science, 43–50.
Purver, Matthew, and Ruth Kempson
2004 Incremental context-based generation for dialogue. A. Belz, R. Evans, and P. Piwek (eds.), *Proceedings of the 3rd International Conference on Natural Language Generation (INLG04)*, Lecture Notes in Artifical Intelligence no. 3123. Brockenhurst, UK: Springer, 151–160.
Sato, Yo
2011 Local ambiguity, search strategies and parsing in Dynamic Syntax. In *The Dynamics of Lexical Interfaces*, E. Gregoromichelaki, R. Kempson, and C. Howes (eds.). CSLI Publications.
Schegloff, Emanuel A., Gail Jefferson, and Harvey Sacks
1977 The preference for self-correction in the organization of repair in conversation. *Language* 53 (2): 361–382.
Shriberg, Elizabeth, and Andreas Stolcke
1998 How far do speakers back up in repairs? A quantitative model. *Proceedings of the International Conference on Spoken Language Processing*. 2183–2186.
Skantze, Gabriel, and Anna Hjalmarsson
2010 Towards incremental speech generation in dialogue systems. *Proceedings of the SIGDIAL 2010 Conference*. Tokyo, Japan: Association for Computational Linguistics, 1–8.

5. Appendix

This appendix provides a derivation for a split dialogue in which both input and output of intermediate generation and parsing steps involve partial structures: "A: Today Robin arrives B: from? A: Sweden". Notice how the event node on the matrix tree is represented as EVENT in the two step derivation for A's first utterance. The matrix tree is then omitted from the rest of the steps of the derivation for reasons of space, and represented just as EVENT (but see Figure 4 for the progressive specification of the matrix tree root record type)[9].

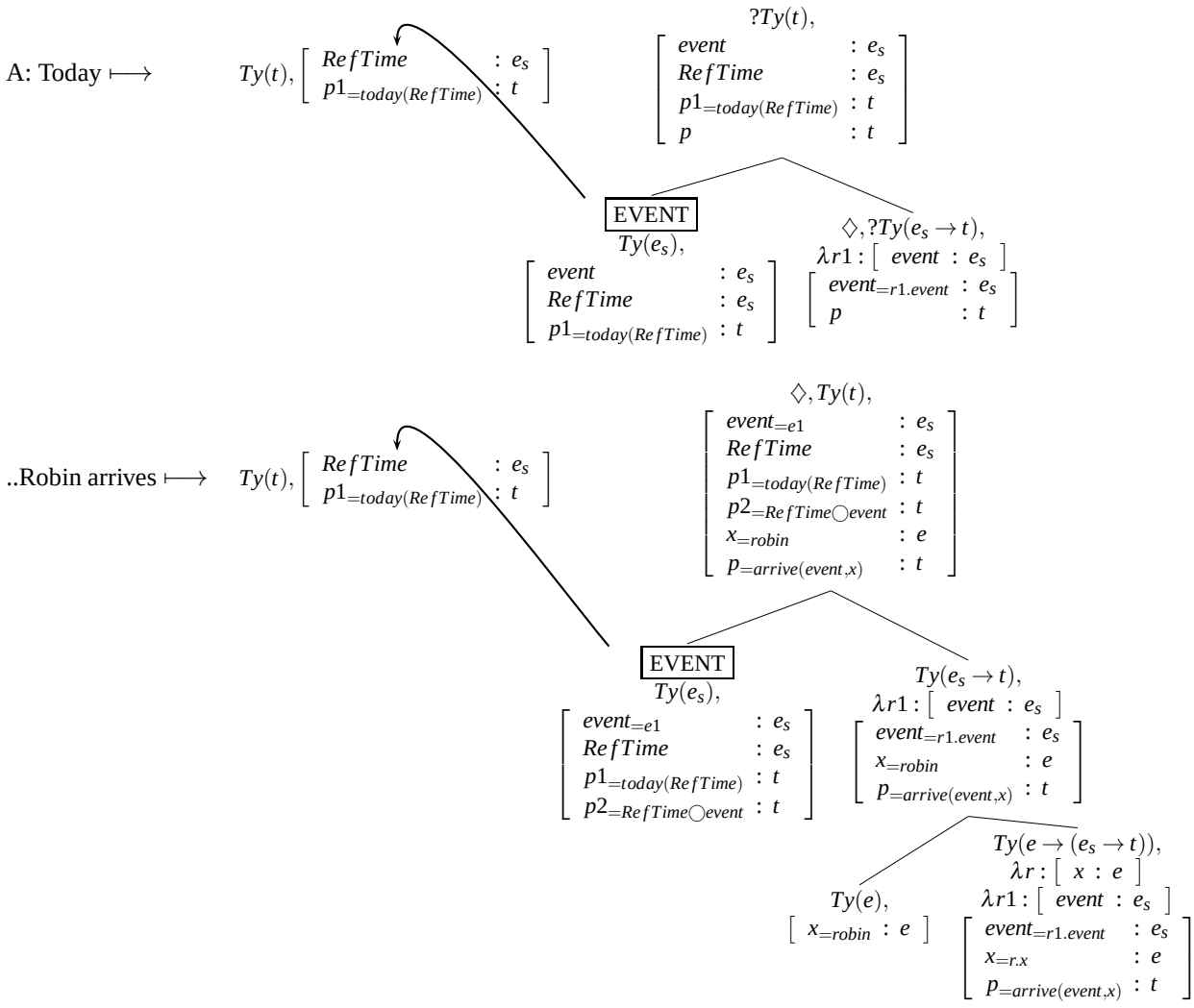

Figure 7. Processing "A: Today, Robin arrives"

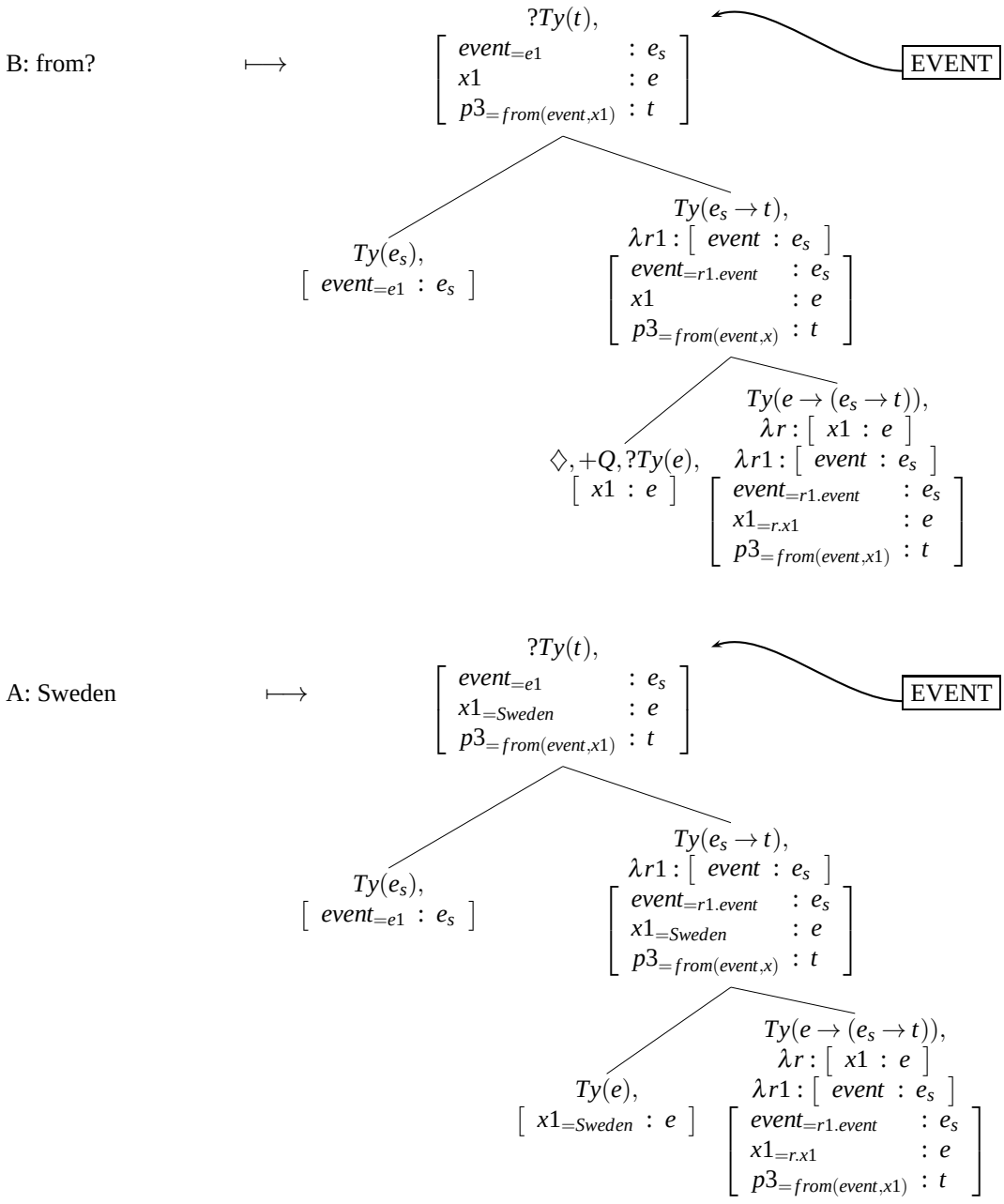

Figure 8. Processing “B: from? A: Sweden”

Notes

1. English has one such exception, in its use of the accusative case in fragments such as *Who is taking this class? Me?*.
2. For functional application and Link-Evaluation (see ch. 3, Cann, Kempson, and Marten (2005), but also Appendix 1 for example DS-TTR derivations involving Link-Evaluation), which require the intersection/concatenation of two record types, *relabelling* is carried out when necessary to avoid leaving incorrect variable names in the record types in the manner of Cooper (2005) and Fernández (2006).
3. See Cann (2011) for the detailed Reichenbachian treatment of tense/aspect used here.
4. This ease of matching incrementality in both generation and parsing is not matched by other models aiming to reflect incrementality in the dialogue model while adopting relative conservative grammar frameworks, some matching syntactic requirements but without incremental semantics (Skantze and Hjalmarsson 2010), others matching incremental growth of semantic input but leaving the incrementality of structural growth unaddressed (Guhe 2007).
5. Since Figure 3 is given to display the generation path dynamics, event term specifications are omitted for simplicity.
6. Available from http://dylan.sourceforge.net/
7. Poesio and Rieser provide a detailed account of how a suggested collaborative completion might be derived using inferential processes and the recognition of plans: by matching the partial representation at speaker transition against a repository of known plans in the relevant domain, an agent can determine the components of these plans which have not yet been made explicit and make a plan to generate them.
8. The calling up of the requisite mechanisms would also lead directly to predictions of processing complexity that we have strong reason to believe will not be met.
9. Each of these steps involves attaching a LINKed tree by way of adjunction to this event node in the (omitted) matrix tree, so that in the final derivation there are in fact two LINKed trees linked to the EVENT node on the matrix tree.