Incrementality and HPSG: Why not?

Jonathan Ginzburg and Robin Cooper and
Julian Hough and David Schlangen

1 Introduction

Incremental processing at least as fine grained as word-by-word has long been accepted as a basic feature of human processing of speech (see e.g., Schlesewsky and Bornkessel (2004)) and as an important feature for design of spoken dialogue systems (see e.g., Schlangen and Skantze (2011), Hough et al. (2015)).[1]

The ability to deal with incrementality has for many years been a selling point of Categorial Grammar in both its versions CCG (Ades and Steedman (1982), Steedman (1996)) and TLG (Morrill (2000)), and in LTAG (Demberg et al. (2013)). It has also served as the motivation for new formalisms, e.g., Dynamic Dependency Grammar (Milward (1994)) and, more recently Dynamic Syntax (Kempson et al. (2001)). Over the years there has been some work on incremental versions of HPSG, e.g., (Güngördü (1997)) and recently (Haugereid and Morey (2012)). Nonetheless, on the whole, incrementality in HPSG has been viewed as a performance issue—see e.g., Sag and Wasow (2015):

> 'The locality of the constraints maximizes the information available in partial structures and supports a variety of processing regimes (top-down, bottom-up, left-corner, probabilistic, etc.). Hence, this property of our sign-based model of grammar is useful in modeling the *incrementality of processing* (our italiscs—JG,RC).' (Sag and Wasow (2015), p. 53)

Over the last few years, several works have appeared detailing the view that grammars should be viewed as systems that classify an utterance as it occurs in conversation see e.g., (Ginzburg (2012), Ginzburg and Poesio (2016), Kempson et al. (2016), Cooper (2016)). Thus, Ginzburg and Poesio (2016) argue that phenomena such as disfluencies, non sentential utterances, quotation, and co–speech gestures are as rule-governed as binding, control, and dislocation—traditional sentence-level phenomena captured in formal grammars. Given the existence of formal accounts for all these conversational phenomena within frameworks such as KoS (Ginzburg (2012)), PTT (Poesio and Rieser (2010)), SDRT (Asher and Lascarides (2003)), Dynamic Syntax (Kempson et al. (2016))

[1] This work is dedicated with friendship to Danièle Godard. Portions of this work were presented at Sinn und Bedeutung 2016, held in September 2016 in Edinburgh and at the workshop to honour Danièle, held at the Hôtel de Lauzon in February 2017.

and other related frameworks, this suggests the need for a new view wherein grammar is a means for directly characterising speech events, abolishing the performance/competence distinction (though recasting this in a way that allows maintaining a distinction between the linguistic phenomena from the specific details of how they get processed.).

Indeed, with respect to incrementality, once one examines ongoing conversational data even in a fairly cursory fashion, as we exemplify below, one discovers the pervasive nature of phenomena whose analysis requires incremental semantic composition.

Consequently, we believe that such data push any grammar formalism that aspires to handle conversation, and this includes without doubt HPSG, to adapt and offer means of handling incremental semantic composition. However, this does not, as we will suggest, *force* one to radically redesign one's formalism, as long as one allows for a sufficiently tight coupling between grammar and conversational context.

We start, therefore in section 2 with a cursory examination of phenomena from conversation that requires incremental semantic composition and draw from this basic specifications for incremental semantics. In section 3 we present the necessary background concerning KOS and TTR, a type theory with records, the frameworks we employ for representing dialogue, grammar, and semantics. In section 4, we sketch an account of dialogical incremental processing, which we apply to some of the data from section 2 in section 5.

2 Incremental Composition: Data and Initial specification

Example (1) exemplifies the fact that at any point in the speech stream of A's utterance B can interject with an acknowledgement whose force amounts to B understanding the initial segment of the utterance (Clark (1996)):

(1) A: Move the train ... B: Aha A: ... from Avon ... B: Right A: ... to Danville. (Trains corpus)

(1) requires us to be able to write a lexical entry for 'aha' and 'yeah' (and their counterparts cross linguistically, e.g., French: 'ouais', 'mmh',...,) whose context is/includes "an incomplete utterance". (2a,b,c) exemplify a contrast between three reactions to an 'abandoned' utterance: in (2a) B asks A to elaborate, whereas in (2b) she asks him to complete her unfinished utterance; in (2c) B indicates that A's content is evident and he need not spell it out:

(2) a. A(i): John ... Oh never mind. B(ii): What about John/What happened to John? A: He's a lovely chap but a bit disconnected.

b. A(i): John ... Oh never mind. B(ii): John what? A: burnt himself while cooking last night.

c. A: Bill is . . . B: Yeah don't say it, we know.

(2a,b,c) requires us to associate a content with A's incomplete utterance which can either trigger an elaboration query (2a), a query about utterance completion (2b), or an acknowledgement of understanding (2c). (3) is an attested example of an abandoned utterance in mid-word:

(3) [Context: J is in the kitchen searching for the always disappearing scissors. As he walks towards the cutlery drawer he begins to make his utterance, before discovering the scissors once the drawer is opened.] J: Who took the sci-. . .

(3) requires us to integrate within-utterance and (in this case, visual) dialogue context processing.

(4) exemplifies two types of expressions—filled pauses and exclamative interjections– that can in principle, be inserted at any point in the speech stream of A's utterance; the interjection 'Oh God' here reacts to the utterance situation conveyed incrementally.

(4) Audrey: Well it's like th- it's like the erm (pause) oh God! I've forgotten what it's bloody called now? (British National Corpus)

(4) requires us to enable the coherence of a question about what word/phrase will follow, essentially at any point in the speech stream; It also requires us to enable the coherence of an utterance expressing negative evaluation of the current incomplete utterance. (5a-e) illustrate that an incomplete clause can serve as an antecedent for a sluice, thereby going against the commonly held assumption that sluicing is an instance of 'S–ellipsis':

(5) a. The translation is by—who else? —Doris Silverstein (The TLS, Feb 2016)

b. He saw—can you guess who?—The Dude;

c. Queen Rhonda is dead. Long live . . . who? (New York Times, Nov 2015);

d. A: A really annoying incident. Some idiot, B: Who? A: Not clear. B: OK A: has taken the kitchen scissors.

e. (From a live blog:) On 2nd & 4, Brady finds, who else?, Damon Amendola who stretches out to make a touchdown catch that gives the Patriots the lead.

(5) requires us to enable either incomplete argument frames or QNPs immediately after their utterance to trigger sluices.

Dialogue Gameboard		
component	*type*	*keeps track of*
Spkr	Individual	*Turn*
Addr	Individual	*ownership*
utt-time	Time	
Facts	Set(propositions)	*Shared assumptions*
VisualSit	Situation	*Visual scene*
Moves	List(Locutionary propositions)	*Grounded utterances*
QUD	Partially ordered	*Live*
	set(⟨question, FEC⟩)	*issues*
Pending	List(Locutionary propositions)	*Ungrounded utterances*

Table 1: Dialogue Gameboard

3 Background

3.1 KoS

For our dialogical framework we use KoS (Ginzburg (1994), Larsson (2002), Purver (2006), Ginzburg (2012))). KoS provides a cognitive architecture in which there is no single common ground, but distinct yet coupled Dialogue GameBoards, one per conversationalist. The structure of the dialogue gameboard (DGB) is given in table 1. The *Spkr* and *Addr* fields allow one to track turn ownership; *Facts* represents conversationally shared assumptions; *VisualSit* represents the dialogue participant's view of the visual situation and attended entities; *Pending*, the nature of which we explicate in more detail below, represents moves that are in the process of being grounded and *Moves* represents moves that have been grounded; *QUD* tracks the questions currently under discussion, though not simply questions *qua* semantic objects, but pairs of entities which we call *InfoStrucs*: a question and an antecedent sub-utterance.[2] This latter entity provides a partial specification of the focal (sub)utterance, and hence it is dubbed the *focus establishing constituent* (FEC). This is similar to the *parallel element* in higher order unification–based approaches to ellipsis resolution e.g. Gardent and Kohlhase (1997); and to Vallduvì (2015), who relates the focus establishing constituent with a notion needed to capture *contrast*.

3.2 TTR

The logical underpinnings of KoS is TTR, a type theory with records (Cooper, 2012, Cooper and Ginzburg, 2015). TTR is a framework that draws its inspirations from two quite distinct sources. One source is Constructive Type Theory for the repertory of type constructors, and in particular records and record types, and the notion of witnessing conditions. The second source is situation

[2]Extensive motivation for this view of QUD can be found in (Fernández, 2006, Ginzburg, 2012), based primarily on semantic and syntactic parallelism in non-sentential utterances such as short answers, sluicing, and various other non-sentential utterances.

semantics (Barwise (1989)) which TTR follows in viewing *semantics as ontology construction*. This is what underlies the emphasis on specifying structures in a model theoretic way, introducing structured objects for explicating properties, propositions, questions etc. It also takes from situation semantics an emphasis on *partiality* as a key feature of information processing. This aspect is exemplified in a key assumption of TTR—the witnessing relation between records and record types: the basic relationship between the two is that a record r is of type T if each value in r assigned to a given label l_i satisfies the typing constraints imposed by T on l_i:

(6) record witnessing

The record:

$$\begin{bmatrix} l_1 & = & a_1 \\ l_2 & = & a_2 \\ \ldots & & \\ l_n & = & a_n \end{bmatrix}$$

is of type:

$$\begin{bmatrix} l_1 & : & T_1 \\ l_2 & : & T_2(l_1) \\ \ldots & & \\ l_n & : & T_n(l_1, l_2, \ldots, l_{n-1}) \end{bmatrix}$$

iff $a_1 : T_1, a_2 : T_2(a_1), \ldots, a_n : T_n(a_1, a_2, \ldots, a_{n-1})$

This allows for cases where there are fields in the record with labels not mentioned in the record type. This is important when e.g., records are used to model contexts and record types model rules about context change—we do not want to have to predict in advance all information that could be in a context when writing such rules.

For what follows, we require use of an analog to priority unification for record types in *asymmetric merge* (Cooper, 2012, Hough, 2015) defined as: given two record types $R1$ and $R2$, $R1 \boxed{\dot{\wedge}} R2$ will yield a record type which is the union of all fields with labels not shared by R1 and R2 and the asymmetric merge of the remaining fields with the same labels, whereby R2's type values take priority over R1's fields, yielding a resulting record type with R2's fields only in those cases.

(7) Asymmetric Merge

$$\begin{bmatrix} \text{a}:T_1 \\ \text{b}:T_2 \\ \text{c}:T_3 \end{bmatrix} \boxed{\dot{\wedge}} \begin{bmatrix} \text{b}:T_2 \\ \text{c}:T_4 \end{bmatrix} = \begin{bmatrix} \text{a}:T_1 \\ \text{b}:T_2 \\ \text{c}:T_4 \end{bmatrix}$$

3.2.1 Conversational Rules

Context change is specified in terms of *conversational rules*, rules that specify the *effects* applicable to a DGB that satisfies certain *preconditions*. This allows both illocutionary effects to be modelled (preconditions for and effects of greeting, querying, assertion, parting etc.), interleaved with *locutionary effects*. We mention here one rule that we use subsequently. QSPEC is KoS' version of Gricean Relevance—it characterizes the contextual background of reactive queries and assertions. QSPEC says that if q is QUD–maximal, then subsequent to this either conversational participant may make a move constrained to be q–specific (i.e. either a partial answer or sub–question of q).[3]

(8) QSPEC

$$\begin{bmatrix} \text{pre} = \begin{bmatrix} \text{qud} = \left\langle \text{i, I} \right\rangle : \text{poset}(\mathit{InfoStruc}) \end{bmatrix} \\ \text{effects} = \mathit{TurnUnderspec} \ \boxed{\dot{\wedge}} \begin{bmatrix} \text{r} : \mathit{AbSemObj} \\ \text{R}: \mathit{IllocRel} \\ \text{LatestMove} = \text{R(spkr,addr,r)} : \mathit{IllocProp} \\ \text{c1} : \text{Qspecific(r,i.q)} \end{bmatrix} \end{bmatrix}$$

Update procedure: Using asymmetric merge, we employ the following update process for a dialogue context C and for some rule R, a record of type (9).

(9)
$$\begin{bmatrix} \text{pre} & : & \mathit{RecType} \\ \text{effects} & : & \mathit{RecType} \end{bmatrix}$$

When updating from one context C_i to the next C_{i+1} with rule R:

(10) If $C_i : T_{C_i}$ and T_{C_i} is a subtype of R.pre,
then R licenses the conclusion that:
$C_{i+1} : T_{C_i} \ \boxed{\dot{\wedge}} \ R.\text{effects}$

The updates operate on various levels of information which can be arbitrarily fine-grained (even phonetic). This gives us the requisite apparatus for the incrementality discussed in section 2.

3.3 Grounding/Clarification interaction Conditions

Much recent work in dialogue has emphasized two essential branches that can ensue in the aftermath of an utterance:

- **Grounding**: the utterance is understood, its content is added to common ground, uptake occurs.
- **Clarification Interaction**: some aspect of the utterance causes a problem; this triggers exchange to repair problem.

[3]We notate the underspecification of the turn holder as *TurnUnderspec*, an abbreviation for the following specification which gets unified together with the rest of the rule:

$$\begin{bmatrix} \text{PrevAud} = \left\{\text{pre.spkr,pre.addr}\right\} & : & \text{Set}(\mathit{Ind}) \\ \text{spkr} & : & \mathit{Ind} \\ \text{c1} & : & \text{member(spkr, PrevAud)} \\ \text{addr} & : & \mathit{Ind} \\ \text{c2} & : & \text{member(addr, PrevAud)} \wedge \text{addr} \neq \text{spkr} \end{bmatrix}$$

KoS's treatment of repair involves two aspects. One is straightforward, drawing on an early insight of Conversation Analysis, namely that repair can involve 'putting aside' an utterance for a while, a while during which the utterance is repaired. That in itself can be effected without further ado by adding further structure to the DGB, specifically the field introduced above called *Pending*. 'Putting the utterance aside' raises the issue of *what is it that we are 'putting aside?'*. In other words, how do we represent the utterance? The requisite information needs to be such that it enables the original speaker to interpret and recognize the coherence of the range of possible clarification queries that the original addressee might make. Ginzburg (2012) offers detailed arguments on this issue, including considerations of the phonological/syntactic parallelism exhibited between CRs and their antecedents and the existence of CRs whose function is to request repetition of (parts of) an utterance. Taken together with the obvious need for *Pending* to include values for the contextual parameters specified by the utterance type, Ginzburg concludes that the type of *Pending* combines tokens of the utterance, its parts, and of the constituents of the content with the utterance type associated with the utterance. An entity that fits this specification is the *locutionary proposition* defined by the utterance. A locutionary proposition is a proposition whose situational component is an utterance situation, typed as in (11a) and will have the form of record (11b):

(11) a. $LocProp =_{def} \begin{bmatrix} \text{sit} : \textit{Sign} \\ \text{sit-type} : \textit{RecType} \end{bmatrix}$ b. $\begin{bmatrix} \text{sit} = u \\ \text{sit-type} = T_u \end{bmatrix}$

Here T_u is a grammatical type for classifying u that emerges during the process of parsing u. It can be identified with a *sign* in the sense of HPSG (Pollard and Sag, 1994). This is operationalized as follows: given a presupposition that u is the most recent speech event and that T_u is a grammatical type that classifies u, a record p_u of the form (11b), gets added to *Pending*. The two branches lead to the following alternative updates:

- Grounding, utterance u understood: update MOVES with p_u and respond appropriately (with the second half of an adjacency pair etc.)
- Clarification Interaction:
  1. p_u remains for future processing in PENDING;
  2. CQ(u), a clarification question calculated from p_u, updates QUD and CQ(u) becomes a discourse topic.

4 An incremental perspective on grounding and clarification

4.1 Incrementalizing dialogue processing

The account in section 3.3 was extended to self-repair in Ginzburg et al. (2014): the basic idea is simply to incrementalize the perspective from the turn level to the word level: as the utterance unfolds incrementally there potentially arise questions about what has happened so far (e.g. *what did the speaker mean with sub-utterance u1?*) or what is still to come (e.g. *what word does the speaker mean to utter after sub-utterance u2?*). These can be accommodated into the context if either uncertainty about the correctness of a sub-utterance arises or the speaker has planning or realizational problems. Overt examples for such accommodation are provided by self-addressed questions (*She saw the ... what's the word?, Je suis comment dire?*), as explained below.

The account of Ginzburg et al. (2014) exemplified some incremental contents and explained a significant conceptual change that would need to be assumed—that *Pending* would have incremental utterance representations. It did not, however, begin to spell out concretely the nature of such representations, which are crucial in a third option a speaker has apart from grounding and (self)clarifying, namely *prediction* (see examples (2) and (3) above).

We can summarize this picture of processing as in (12), the monitoring and update/clarification cycle is modified to happen *at the end of each word utterance event*, and in case of the need for repair, a repair question gets accommodated into QUD.

(12) a. Ground: continue (Levelt (1983)).

b. Predict: stop, since content is predictable.

c. (Self)Clarify: generate CR given lack of expected utterance.

In the rest of this section we sketch an account of incremental utterance representations, including in particular incremental semantic contents.

4.2 Update Rules for specifying syntax

An essential presupposition of our approach (already in its non-incremental version, see above) is a view of syntax as speech event classification by an agent. For a very detailed exposition of such a view see Cooper (2016), a précis of which can be found in Cooper (2013). Starting at the word level—if Lex(T_w, C) is a sign type[4] which is one of the lexical resources available to an agent A and A judges an event e to be of type T_w, then A is licensed to update their

[4]e.g., Lex('Beethoven', NP) representing something like

$$\left[\begin{array}{lll} \text{s-event} & : & \left[\begin{array}{lll} \text{e-loc} & : & \mathit{Loc} \\ \text{sp} & : & \mathit{Ind} \\ \text{au} & : & \mathit{Ind} \\ \text{e} & : & \text{“Beethoven”} \\ \text{c}_{\text{loc}} & : & \text{loc(e,e-loc)} \\ \text{c}_{\text{sp}} & : & \text{speaker(e,sp)} \\ \text{c}_{\text{au}} & : & \text{audience(e,au)} \end{array}\right] \\ \text{syn} & : & \left[\begin{array}{lll} \text{cat=np} & : & \mathit{Cat} \\ \text{daughters=}\varepsilon & : & \mathit{Sign}^* \end{array}\right] \\ \text{cont=Beethoven}' & : & \mathit{Cont} \end{array}\right]$$

where 'Beethoven′' is whatever your theory gives you as the content of the proper noun Beethoven.

DGB with the type Lex(T_w,C). Intuitively, this means that if the agent hears an utterance of the word "composer", then they can conclude that they have heard a sign which has the category noun. This is the beginning of *parsing*, which Cooper shows how to assimilate to a kind of update akin to that involved in non-linguistic event perception. The licensing condition corresponding to lexical resources like (12) is given in (13). We will return below to how this relates to gameboard update. (13) says that an agent with lexical resource Lex(T, C) who judges a speech event, u, to be of type T is licensed to judge that there is a sign of type Lex(T, C) whose 's-event.e'-field contains u.

(13) If Lex(T, C) is a resource available to agent A, then for any u, $u :_A T$ licenses $:_A$ Lex(T, C) $\boxed{\dot{\wedge}}$ $\left[\text{s-event:}\left[\text{e}{=}u{:}T\right]\right]$

Strings of utterances of words can be classified as utterances of phrases. That is, speech events are hierarchically organized into types of speech events in a way akin to the complex event structures needed to model non-linguistic activities. Agents have resources which allow them to reclassify a string of signs of certain types ("the daughters") into a single sign of another type ("the mother"). For instance, a string of type Det⌢N (that is, a concatenation of an event of type Det and an event of type N) can lead us to the conclusion that we have observed a sign of type NP whose daughters are of the types Det and N respectively.

The resource that licences this is a rule which is modelled as the function in (14a) which we represent as (14b)

(14) a. $\lambda u : Det^\frown N \ . \ NP \ \boxed{\dot{\wedge}} \ \left[\text{syn:}\left[\text{daughters}{=}u{:}Det^\frown N\right]\right]$

b. RuleDaughters(NP, $Det^\frown N$)

'RuleDaughters' is to be the function in (15). Thus 'RuleDaughters', if provided with a subtype of $Sign^+$ and a subtype of *Sign* as arguments, will return a function which maps a string of signs of the first type to the second type with the restriction that the daughters field is filled by the string of signs:

(15)
$$\begin{array}{l}\lambda T_1 : \mathit{Type} \ . \\ \quad \lambda T_2 : \mathit{Type} \ . \\ \quad\quad \lambda u : T_1 \ . \ T_2 \ \boxed{\dot{\wedge}} \ \left[\text{syn:}\left[\text{daughters}{=}u{:}T_1\right]\right]\end{array}$$

4.3 Semantic Composition using asymmetric merge

As we mentioned on p. 5, we use asymmetric merge to integrate utterances into the DGB. We postulate as the denotation associated with the root of the tree the type *illocutionary proposition*, which is hence compatible with declarative, interrogative and imperative utterances. This gets refined as each word gets introduced using asymmetric merge, which enables us to effect a combinatory operation that synthesises function application and unification.

We exemplify how this works in explicating the evolution of the speaker's information state in example (3), repeated here as (16).

(16) [Context: J is in the kitchen searching for the always disappearing scissors. As he walks towards the cutlery drawer he begins to make his utterance, before discovering the scissors once the drawer is opened.] J: Who took the sci-. . .

(17) InfState$_0$: T_0 where T_0 is

$$\left[\text{private:}\begin{bmatrix}\text{agenda=}\langle\text{ask(speaker,}q_0)\rangle\text{:}\langle \textit{Type}\rangle\\ \text{vis-sit:}\textit{NoScissors}\end{bmatrix}\right]$$

Here we use *NoScissors* to represent a type of types, $\mathcal{T}$, such that for any type, T, $T : \mathcal{T}$ iff there is no relabelling, T', of the type (18) such that $T \sqsubseteq T'$.

(18)
$$\begin{bmatrix} \text{x} & : & \textit{Ind} \\ \text{e} & : & \text{scissors(x)} \end{bmatrix}$$

That is, the speaker judges the visual situation to be one which does not show any evidence of scissors. For discussion of the relabelling of types see Cooper (2016).

We assume that an utterance, u, of an interrogative NP such as *who* results in the update of the type of the current information state, T_0, in (19).

(19) InfState$_1$: T_0 $\boxed{\wedge}$

$$\left[\text{DGB.Pending} = \begin{bmatrix}\text{sit} = u : \textit{Sit}\\ \text{sit-type=}\begin{bmatrix}\text{phon : who}\\ \text{cont} = w\text{: } (\textit{Ppty}\rightarrow \textit{WhPQ})\end{bmatrix}\text{:}\textit{RecType}\end{bmatrix}\text{: } \textit{RecType}\right]$$

Here *Ppty* is the type given in (20).

(20) $(\begin{bmatrix}\text{x:}\textit{Ind}\end{bmatrix}\rightarrow \textit{RecType})$

The content associated with the utterance involves *projection* in a sense we explicate shortly. Here it is projected to be a question of type *WhPQ*.

(21) $(\begin{bmatrix}\text{x:}\textit{Ind}\\ \text{c:person(x)}\end{bmatrix}\rightarrow \textit{RecType})$

The function, w, in (20) which might be seen as serving as the incremental content (*cf.* Milward and Cooper, 1994) of *who* is given in (22), though we could also regard it as a straightforward static content in a compositional semantics.[5]

(22) $w = \lambda P\text{:}\textit{Ppty}\ .\ \lambda r\text{:}\begin{bmatrix}\text{x:}\textit{Ind}\\ \text{c:person(x)}\end{bmatrix}\ .\ \begin{bmatrix}\text{e:}P(r)\end{bmatrix}$

[5]Milward and Cooper (1994) offer an explicit procedure that converts such lambda terms to existentially quantified propositions. Their fragment considered only declarative utterances. In the current work we could adapt their procedure to yield existentially quantified *illocutionary* propositions by converting functions to record types, in this case, for example:

$$\begin{bmatrix} \text{P} & : & \textit{Ppty} \\ \text{r} & : & \begin{bmatrix} \text{x} & : & \textit{Ind} \\ \text{c} & : & \text{person(x)} \end{bmatrix} \\ \text{e} & : & \text{P(r)} \end{bmatrix}$$

We posit the content of the verb *took* to be (23a) (ignoring tense) of type (23b). We represent this content as 'take′'.

(23) a. $\text{take}' = \lambda r_1{:}\big[\text{x:}\mathit{Ind}\big]\ .\ \lambda r_2{:}\big[\text{x:}\mathit{Ind}\big]\ .\ \big[\text{e:take}(r_2.\text{x},\ r_1.\text{x})\big]$

b. $(\big[\text{x:}\mathit{Ind}\big]{\rightarrow}\mathit{Ppty})$

Thus the incremental content of *who took* can be computed in line with Milward and Cooper (1994) as (24a) which can be expressed with reference to InfState_1 as (24b).

(24) a. $\lambda r{:}\big[\text{x:}\mathit{Ind}\big]\ .\ w(\text{take}'(r))$

b. $\lambda r{:}\big[\text{x:}\mathit{Ind}\big]\ .\ \text{InfoState}_1\text{.DGB.Pending.sit-type.cont}(\text{take}'(r))$

We abbreviate (24b) as *wt*. We can compute a type for InfState_2 as in (25).

(25) $\text{InfState}_2 : T_1\ \boxed{\dot{\wedge}}$

$$\left[\text{DGB.Pending} = \begin{bmatrix}\text{sit} = u_2{:}\mathit{Sit} \\ \text{sit-type=}\begin{bmatrix}\text{phon : who took} \\ \text{cont} = wt : (\big[\text{x:}\mathit{Ind}\big]{\rightarrow}\mathit{WhPQ})\end{bmatrix}{:}\mathit{RecType}\end{bmatrix}{:}\mathit{RecType}\right]$$

We use T_2 to represent the type computed in (25). J opens the drawer and sees the scissors there. This updates the type of the visual situation so that it now requires the presence of scissors. This, in turn, implies that no one took the scissors, and hence, given the existence of a resolving answer to the question, the original motivation for asking it is eliminated. We can now compute a type for the next information state in which the agenda is empty.

What we have sketched here is an approach to incrementality like that in Milward and Cooper (1994) which is similar to that which can be taken in a categorial grammar framework such as CCG (Demberg, 2012). Another approach to incrementality is to use something similar to charts in chart parsing, which we sketch in the next section. We believe that, ultimately, the two approaches need to be combined to provide a complete treatment of incremental semantics.

4.4 Pending and charts

Information included in the 'Pending'-field of the dialogue gameboard includes a type that represents the agent's view of the ongoing parse as the utterance unfolds. We call this type a *chart-type* because we appeal to a notion of chart parsing for this purpose, though as will become clear our approach is compatible with various other approaches for such representations, for instance Hough's graph-based representation (Hough, 2015) which synthesizes a graph-based Dynamic Syntax view of parsing (Sato, 2011) with the Incremental Unit (IU) framework of Schlangen and Skantze (2011) for incremental processing.

The type of Pending remains *LocProp*, as in (26). The issue that remains is how to explicate T_{chart} in order to understand how incremental content arises.

(26) $\begin{bmatrix} \text{sit} = s \\ \text{sit-type} = T_{chart} \end{bmatrix}$

We present here the briefest sketch of chart parsing as it is used in computational linguistics; for a recent textbook introduction to chart parsing see Jurafsky and Martin (2009), Chap. 13, whereas for its implementation in TTR see Cooper (2016). The idea of a chart is that it should store all the hypotheses made during the processing of an utterance which in turn allow us to compute new hypotheses to be added to the chart. Charts can be updated incrementally for each word and they can represent several live possibilities in a single data structure. We will say that a chart is a record and we will use our resources to compute a chart type on the basis of utterance events.

4.5 Charts: a simplified example

Suppose that we have so far heard an utterance of the word *who*. At this point we will say that the type of the chart is (27)

(27) $\begin{bmatrix} \text{e}_1 & : & \text{“who”} \\ \text{e} & : & [\text{e}_1\text{:start(e}_1)]^\frown[\text{e}_1\text{:end(e}_1)] \end{bmatrix}$

The main event of the chart type (represented by the e-field) breaks the phonological event of type “who” down into a string of two events, the start and the end of the “who”-event.[6] Thus (27) records that we have observed an event of the phonological type “who” and an event consisting of the start of that event followed by the end of that event. Given that we have the resource Lex_{WH}(“who”) available which yields the sign type for an utterance of “who”, we can update (27) to (28):

(28) $\begin{bmatrix} \text{e}_1 & : & \text{“who”} \\ \text{e}_2 & : & \text{Lex}_{\text{WH}}(\text{“who”}) \boxed{\wedge} [\text{s-event:}[\text{e=e}_1\text{:}\mathit{Phon}]\ ] \\ \text{e} & : & \begin{bmatrix}\text{e}_1\text{:start(e}_1) \\ \text{e}_2\text{:start(e}_2)\end{bmatrix}^\frown\begin{bmatrix}\text{e}_1\text{:end(e}_1) \\ \text{e}_2\text{:end(e}_2)\end{bmatrix} \end{bmatrix}$

That is, we add the information to the chart that there is an event (labelled ‘e_2’) of the type which is the sign type corresponding to “who” and that the event which is the speech event referred to in that sign type is the utterance event, labelled by ‘e_1’. Furthermore the duration of the event labelled ‘e_2’ is the same as that labelled ‘e_1’.

The type Lex_{WH}(“who”) is a subtype of *NP*. Thus the event labelled ‘e_2’ could be the first item in a string that would be appropriate for the function which we have abbreviated as (29a), which has the type (29b).

(29) a. $S \longrightarrow NP\ \ VP \mid NP'(VP')$

b. $(NP^\frown VP \rightarrow \mathit{Type})$

[6]These starting and ending events correspond to what are standardly called *vertices* in the chart parsing literature.

Cooper (2016) argues for an analogy between non-linguistic event prediction and the prediction that occurs in parsing.[7] So on observing a noun-phrase event one can predict that it might be followed by a verb phrase event thus creating a sentence event. We add a hypothesis event to our chart which takes place at the end of the noun-phrase event as in (30).[8]

(30)
$$
\left[\begin{array}{lll}
\text{e}_1 & : & \text{``who''} \\
\text{e}_2 & : & \text{Lex}_{\text{WH}}(\text{``who''})\ \boxed{\wedge}\ \left[\text{s-event}:\left[\text{e}=\text{e}_1:\textit{Phon}\right]\ \right] \\
\text{e}_3 & : & \left[\begin{array}{l} \text{rule}=S \longrightarrow NP\ VP \mid NP'(VP'):(NP^\frown VP \rightarrow \textit{Type}) \\ \text{fnd}=\text{e}_2:\textit{Sign} \\ \text{req}=VP:\textit{Type} \\ \text{e:required(req,rule)} \end{array}\right] \\
\text{e} & : & \left[\begin{array}{l} \text{e}_1:\text{start}(\text{e}_1) \\ \text{e}_2:\text{start}(\text{e}_2)\end{array}\right]^\frown \left[\begin{array}{l} \text{e}_1:\text{end}(\text{e}_1) \\ \text{e}_2:\text{end}(\text{e}_2) \\ \text{e}_3:\text{start}(\text{e}_3)^\frown\text{end}(\text{e}_3)\end{array}\right]
\end{array}\right]
$$

In the e_3-field the 'rule'-field is for a syntactic rule, that is, a function from a string of signs of a given type to a type. The 'fnd'-field is for a sign or string of signs so far found which match an initial segment of a string of the type required by the rule. The 'req'-field is the type of the remaining string required to satisfy the rule as expressed in the 'e'-field. This hypothesis event both starts and ends at the end of the event of the noun-phrase event e_2.

5 Incremental Dialogue Processing: principles and examples

With a basic means of representing utterances in progress, we can now formulate certain principles which will use to explicate several of the phenomena discussed in section 2.

5.1 Utterance Projection

The first principle we introduce corresponds to the 'stop option' in our utterance protocol (12b)—it says that if one projects that an utterance will continue in a certain way, then one can actually use this prediction to update one's DGB. This is of course a dangerous principle to apply in an unconstrained fashion, and would ideally be formulated using probabilities about the projection, for instance using the framework of Cooper et al. (2015), though we do not do so here. (31) is an update rule which moves a locutionary proposition from pending to LatestMove. (r^* represents the previous information state which is required to be of the type labelled 'preconds'.)

[7] Indeed he suggests that this might extend to non-linguistic event prediction among non-humans, e.g., the prediction by a dog playing Fetch that it should run after a stick which is held up.

[8] In terms of the traditional chart parsing terminology this corresponds to an *active edge* involving a *dotted rule*. The fact that the addition of this type to the chart type is triggered by finding something of an appropriate type to be the leftmost element in a string the would be an appropriate argument to the rule corresponds to what is called a *left-corner* parsing strategy.

(31) **Utterance Projection**

$$\begin{bmatrix} \text{preconds} = \begin{bmatrix} \text{pending.sit} : \textit{Sign} \\ \text{pending.sit-type.proj} : \textit{Type} \end{bmatrix} \\ \text{effects} = \text{TurnUnderspec} \boxed{\wedge} \begin{bmatrix} \text{LatestMove} = r^* : \textit{LocProp} \end{bmatrix} \end{bmatrix}$$

5.2 Forward-Looking Disfluencies

Forward-looking disfluencies are disfluencies where the moment of interruption is followed not by an alteration, but just by a completion of the utterance which is delayed by a filled or unfilled pause (hesitation) or a repetition of a previously uttered part of the utterance (repetitions). As we mentioned with respect to example (4) and in our discussion in section 4.1, we need a means of enabling at any point in the speech stream the emergence of a question about what is still to come in the current utterance. Forward Looking Disfluencies involve the update rule in (32)—given a context where an initial segment of utterance by A has taken place, the next speaker—underspecified between the current one and the addressee—may address the issue of what A intended to say next by providing a co-propositional utterance:

(32) Forward Looking Utterance Rule:

$$\begin{bmatrix} \text{preconds} = \begin{bmatrix} \text{spkr} : \textit{Ind} \\ \text{addr} : \textit{Ind} \\ \text{pending.sit-type} : \begin{bmatrix} \text{fnd} : \textit{Sign} \\ \text{req} : \textit{Sign} \end{bmatrix} \end{bmatrix} \\ \text{effects} = \textit{TurnUnderspec} \boxed{\wedge} \begin{bmatrix} \text{MaxQud} = \\ \begin{bmatrix} \text{q} = \lambda x{:}\textit{Ind} \,.\, \text{MeanNextUtt}(r^*.\text{spkr}, r^*.\text{fnd}, x) \\ \text{fec} = \{\} \end{bmatrix} : \textit{InfoStruc} \\ \text{LatestMove} : \textit{LocProp} \\ \text{c2: Copropositional}(\text{LatestMove}^{content}, \text{MaxQud}) \end{bmatrix} \end{bmatrix}$$

A consequence of (32), is that it offers the potential to explain cases like (33). In the aftermath of a filled pause an issue along the lines of the one we have *posited* as the *effect* of the conversational rule (32) actually gets uttered:

(33) a. Carol 133 Well it's (pause) it's (pause) er (pause) what's his name? Bernard Matthews' turkey roast. (BNC, KBJ)

b. They're pretty ... um, how can I describe the Finns? They're quite an unusual crowd actually.
`http://www.guardian.co.uk/sport/2010/sep/10/small-talk-steve-backley-interview`

On our account such utterances are licensed because these questions are co-propositional with the issue 'what did A mean to say after u0?'. This suggests that a different range of such questions will occur depending on the identity of (the syntactic/semantic type of) u0. This expectation is met, as discussed in Tian et al. (2017), who also discuss cross-linguistic variation with SAQs in English, Chinese, and Japanese.

6 Conclusions and further Work

That people process linguistic input incrementally is a widely shared view. But does this mean that the "competence grammar" must be formulated in a way that enables incremental (minimally word by word and even mid-word) semantic composition to be effected? Various frameworks have responded affirmatively to this question, but HPSG has over the years resisted such a conclusion, preferring to assume that incrementality is merely an aspect of performance. In this paper we have reiterated the view that grammars should be viewed as systems that classify an utterance as it occurs in conversation, a view that has recently been articulated in some detail (Ginzburg (2012), Ginzburg and Poesio (2016), Kempson et al. (2016)). Once one examines ongoing conversational data even in a fairly cursory fashion one discovers the pervasive nature of phenomena whose analysis requires incremental semantic composition. In the paper we sketch how this can be done in a way that allows one to utilize existing 'non–incremental' grammars such as HPSG as long as they interface in a radical way with the conversational context. This approach has parallels to Dynamic Syntax (Kempson et al., 2001), and particularly recent dialogue-friendly versions (Purver et al., 2011, Kempson et al., 2016), where the central idea is online, incremental construction of meaning representations. However, the incremental account presented here not only allows the representation of utterances, but the internal state of a dialogue agent, including background beliefs and the events in the situated context, to be updated online for entire interactions.

In a more detailed presentation we will present a small grammar/context fragment. In future work we hope to investigate experimentally the processing of data of the kind presented here.

References

Ades, Antony and Mark Steedman. 1982. On the order of words. *Linguistics and Philosophy* 6:517–558.

Asher, Nicholas and Alex Lascarides. 2003. *Logics of Conversation*. Cambridge: Cambridge University Press.

Barwise, Jon. 1989. *The Situation in Logic*. CSLI Lecture Notes. Stanford: CSLI Publications.

Clark, Herbert. 1996. *Using Language*. Cambridge: Cambridge University Press.

Cooper, Robin. 2012. Type theory and semantics in flux. In R. Kempson, N. Asher, and T. Fernando, eds., *Handbook of the Philosophy of Science*, vol. 14: Philosophy of Linguistics. Amsterdam: Elsevier.

Cooper, Robin. 2013. Update conditions and intensionality in a type-theoretic approach to dialogue semantics. *Proceedings of SemDial* pages 15–24.

Cooper, Robin. 2016. Type theory and language: From perception to linguistic communication. Book Draft.

Cooper, Robin, Simon Dobnik, Staffan Larsson, and Shalom Lappin. 2015. Probabilistic type theory and natural language semantics. *Linguistic Issues in Language Technology* 10.

Cooper, Robin and Jonathan Ginzburg. 2015. Type theory with records for natural language semantics. In C. Fox and S. Lappin, eds., *Handbook of Contemporary Semantic Theory, second edition*. Oxford: Blackwell.

Demberg, Vera. 2012. Incremental Derivations in CCG. pages 198–206.

Demberg, Vera, Frank Keller, and Alexander Koller. 2013. Incremental, predictive parsing with psycholinguistically motivated tree-adjoining grammar. *Computational Linguistics* 39(4):1025–1066.

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

Gardent, Claire and Michael Kohlhase. 1997. Computing parallelism in discourse. In *IJCAI*, pages 1016–1021.

Ginzburg, Jonathan. 1994. An update semantics for dialogue. In H. Bunt, ed., *Proceedings of the 1st International Workshop on Computational Semantics*. Tilburg: ITK, Tilburg University.

Ginzburg, Jonathan. 2012. *The Interactive Stance: Meaning for Conversation*. Oxford: Oxford University Press.

Ginzburg, Jonathan, Raquel Fernández, and David Schlangen. 2014. Disfluencies as intra-utterance dialogue moves. *Semantics and Pragmatics* 7(9):1–64.

Ginzburg, Jonathan and Massimo Poesio. 2016. Grammar is a system that characterizes talk in interaction. *Frontiers in Psychology* 7:1938.

Güngördü, Zelal. 1997. Incremental constraint-based parsing: An efficient approach for head-final languages .

Haugereid, Petter and Mathieu Morey. 2012. A left-branching grammar design for incremental parsing. In *Proceedings of the 19th international conference on head-driven phrase structure grammar, chungnam national university daejeon*, pages 181–194.

Hough, Julian. 2015. *Modelling Incremental Self-Repair Processing in Dialogue*. Ph.D. thesis, Queen Mary, University of London.

Hough, Julian, Casey Kennington, David Schlangen, and Jonathan Ginzburg. 2015. Incremental semantics for dialogue processing: Requirements, and a comparison of two approaches. In *Proceedings of the 11th International Conference on Computational Semantics (IWCS)*.

Jurafsky, Daniel and James H. Martin. 2009. *Speech and Language Processing*. New Jersey: Prentice Hall, 2nd edn.

Kempson, Ruth, Ronnie Cann, Eleni Gregoromichelaki, and Stergios Chatzikyriakidis. 2016. Language as mechanisms for interaction. *Theoretical Linguistics* 42(3-4):203–276.

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

Larsson, Staffan. 2002. *Issue based Dialogue Management*. Ph.D. thesis, Gothenburg University.

Levelt, Willem J. 1983. Monitoring and self-repair in speech. *Cognition* 14(4):41–104.

Milward, David. 1994. Dynamic dependency grammar. *Linguistics and Philosophy* 17:561–405.

Milward, David and Robin Cooper. 1994. Incremental interpretation: applications, theory, and relationship to dynamic semantics. In *Proceedings of the 15th conference on Computational linguistics-Volume 2*, pages 748–754. ACL.

Morrill, Glyn. 2000. Incremental processing and acceptability. *Computational Linguistics* 26(3):319–338.

Poesio, Massimo and Hannes Rieser. 2010. (prolegomena to a theory of) completions, continuations, and coordination in dialogue. *Dialogue and Discourse* 1:1–89.

Pollard, Carl and Ivan A. Sag. 1994. *Head Driven Phrase Structure Grammar*. Chicago: University of Chicago Press and CSLI.

Purver, M. 2006. Clarie: Handling clarification requests in a dialogue system. *Research on Language & Computation* 4(2):259–288.

Purver, Matthew, Arash Eshghi, and Julian Hough. 2011. Incremental semantic construction in a dialogue system. In J. Bos and S. Pulman, eds., *Proceedings of the 9th IWCS*, pages 365–369. Oxford, UK.

Sag, Ivan A and Thomas Wasow. 2015. Flexible processing and the design of grammar. *Journal of psycholinguistic research* 44(1):47–63.

Sato, Yo. 2011. Local ambiguity, search strategies and parsing in dynamic syntax. *The Dynamics of Lexical Interfaces* pages 205–233.

Schlangen, David and Gabriel Skantze. 2011. A General, Abstract Model of Incremental Dialogue Processing. *Dialogue & Discourse* 2(1):83–111.

Schlesewsky, M. and I. Bornkessel. 2004. On incremental interpretation: Degrees of meaning accessed during sentence comprehension. *Lingua* 114(9).

Steedman, Mark. 1996. *Surface Structure and Interpretation*. Linguistic Inquiry Monographs. Cambridge: MIT Press.

Tian, Ye, Takehiko Maruyama, and Jonathan Ginzburg. 2017. Self addressed questions and filled pauses: A cross-linguistic investigation. *Journal of psycholinguistic research* 46(4):905–922.

Vallduvì, Enric. 2015. Information structure. In M. Aloni and P. Dekker, eds., *The Cambridge Handbook of Semantics*. Cambridge: Cambridge University Press.