Incrementality and Clarification/Sluicing potential[1]

Jonathan GINZBURG — *Université Paris-Diderot (Paris 7)*
Robin COOPER — *University of Gothenburg*
Julian HOUGH — *Bielefeld University*
David SCHLANGEN — *Bielefeld University*

Abstract. 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 (2009); Hough et al. (2015)). Nonetheless, with a few important exceptions (see e.g., Kempson et al. (2016)), incrementality is viewed as an aspect of performance, not semantic meaning. Moreover, it seems to entail giving up on compositionality as a constraining principle on denotations. In this paper, we point to a variety of dialogical phenomena whose analysis incontrovertibly requires a semantics formulated in incremental terms. These include cases, above all with sluicing, that call into question existing assumptions about ellipsis resolution and argue for incremental updating of QUD. The incremental semantic framework we sketch improves on existing such accounts (reviewed in Peldszus and Schlangen (2012); Hough et al. (2015)) on both denotational and contextual fronts: the contents we posit are in fact tightly constrained by a methodological principle more restrictive than traditional compositionality, namely the Reprise Content Hypothesis (Purver and Ginzburg (2004); Ginzburg and Purver (2012); Cooper (2013a)), embedded within independently motivated dialogue states (Ginzburg (2012)).

Keywords: Incremental processing, dialogue, clarification potential, sluicing

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 (2009); Hough et al. (2015)). Nonetheless, with a few important exceptions (see e.g., Kempson et al. (2016)), incrementality is viewed as an aspect of performance, not semantic meaning. Moreover, it seems to entail giving up on compositionality as a constraining principle on denotations. In this paper, we point to a variety of dialogical phenomena whose analysis incontrovertibly requires a semantics formulated in incremental terms. These include cases, above all with sluicing, that call into question existing assumptions about ellipsis resolution and argue for incremental updating of QUD. The incremental semantic framework we sketch improves on existing such accounts (reviewed in Peldszus and Schlangen (2012); Hough et al.

[1] Many thanks to the insightful comments of three Sinn und Bedeutung reviewers, as well as to the audience at Sinn und Bedeutung, 2016. We acknowledge support by the French Investissements d'Avenir–Labex EFL program (ANR-10-LABX-0083) and by the Disfluences, Exclamations, and Laughter in Dialogue (DUEL) project within the Projets Franco-Allemand en sciences humaines et sociales of the Agence Nationale de Recherche (ANR) and the Deutsche Forschunggemeinschaft (DFG). For Hough and Schlangen: this work was supported by the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG).

(2015)) on both denotational and contextual fronts: the contents we posit are in fact tightly constrained by a methodological principle more restrictive than traditional compositionality, namely the Reprise Content Hypothesis (Purver and Ginzburg (2004); Ginzburg and Purver (2012); Cooper (2013a)), embedded within independently motivated dialogue states (Ginzburg (2012)).

The structure of the paper is as follows: in section 2 we introduce the data and draw from it basic specifications for incremental semantics. In section 3 we present the necessary background concerning KOS and 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 the data from section 2 in section 5. We end with some brief conclusions.

2. Why Semantics needs Incrementality : the Data and Initial specification

(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. / # burnt himself while cooking last night.
b. A(i): John . . . Oh never mind. B(ii): John what? A: # He's a lovely chap but a bit disconnected. / 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: A 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.] A: 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' (Merchant (2001)):

(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. A: Someone I'm not saying who / B: No, do say/Who?

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

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 Vallduví (2015), who relates the focus establishing constituent with a notion needed to capture *contrast*.

[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.

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 set(⟨question, FEC⟩)	*Live* *issues*
Pending	List(Locutionary propositions)	*Ungrounded utterances*

Table 1: Dialogue Gameboard

3.2. TTR

The logical underpinnings of KoS is Type Theory with Records (TTR) (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 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 RT if each value in r assigned to a given label l_i satisfies the typing constraints imposed by RT on l_i:

(6) record witnessing

The record:

$$\begin{bmatrix} l_1 & = & a_1 \\ l_2 & = & a_2 \\ \ldots & & \\ l_n & = & a_n \end{bmatrix} \text{ is of type: } \begin{bmatrix} l_1 & : & T_1 \\ l_2 & : & T_2(l_1) \\ \ldots & & \\ l_n & : & T_n(l_1, l_2, \ldots, l_{n-1}) \end{bmatrix} \text{ iff } a_1 : T_1, a_2 : T_2(a_1), \ldots, a_n : T_n(a_1, a_2, \ldots, a_{n-1})$$

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 three rules used subsequently. The first two concern the incrementation of QUD. (8a)[3] specifies that given the LatestMove being q, q becomes maximal in QUD, whereas (8b) concerns the effect of A asserting p: this raises the issue p?—the responder can then either decide to discuss this issue (as a consequence of the rule QSPEC introduced below as (9)) or accept it as positively resolved (as a consequence of a rule we do not mention here):

(8) a. Ask QUD-incrementation b. Assertion QUD-incrementation

$$\begin{bmatrix} \text{pre} = & \begin{bmatrix} \text{q}: \ \textit{Question} \\ \text{LatestMove} = \text{Ask(spkr,addr,q)}: \ \textit{IllocProp} \end{bmatrix} \\ \text{effects} = & \begin{bmatrix} \text{qud} = \left\langle \text{q},r^{*}.\text{qud} \right\rangle: \ \text{poset}(\textit{Question}) \end{bmatrix} \end{bmatrix} \begin{bmatrix} \text{pre} = & \begin{bmatrix} \text{p}: \ \textit{Prop} \\ \text{LatestMove} = \text{Assertion(spkr,addr,p)}: \ \textit{IllocProp} \end{bmatrix} \\ \text{effects} = & \begin{bmatrix} \text{qud} = \left\langle \text{p?},r^{*}.\text{qud} \right\rangle: \ \text{poset}(\textit{Question}) \end{bmatrix} \end{bmatrix}$$

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).[4]

[3] Throughout in update rules we will use r^{*} to refer to the immediately preceding information state which is required to be of the type in the field labelled by 'pre' or 'preconditions'.

[4] 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:

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

(9) QSPEC

$$\left[\begin{array}{l} \text{pre} = \left[\text{qud} = \left\langle \text{i, I} \right\rangle \text{: poset}(\textit{InfoStruc})\right] \\ \text{effects} = \textit{TurnUnderspec} \\ \boxed{\underset{\cdot}{\wedge}} \left[\begin{array}{l} \text{r} : \textit{AbSemObj} \\ \text{R: } \textit{IllocRel} \\ \text{LatestMove} = \\ \text{R(spkr,addr,r)} : \textit{IllocProp} \\ \text{c1 : Qspecific(r,i.q)} \end{array}\right] \end{array}\right]$$

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 (10).

(10) $$\left[\begin{array}{lll} \text{pre} & : & \textit{RecType} \\ \text{effects} & : & \textit{RecType} \end{array}\right]$$

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

(11) If $C_i : T_{C_i}$ and T_{C_i} is a subtype of R.pre,
then R licenses the conclusion that:
$C_{i+1} : T_{C_i} \boxed{\underset{\cdot}{\wedge}} R$.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. The Reprise Content Hypothesis and Generalized Quantifiers

As a means of tightly constraining semantic denotations, we adopt the Reprise Content Hypothesis (RCH, Purver and Ginzburg, 2004; Ginzburg and Purver, 2012; Cooper, 2013a):

(12) *A fragment reprise question queries exactly the standard semantic content of the fragment being reprised.*

This uses the data from responses to clarification questions about a constituent as indicative of its content (e.g., *A: Most students object to the proposal. B: Most students? A: Carl, Max, and Minnie.*). Purver and Ginzburg (2004) and Ginzburg and Purver (2012) use such data to argue in favour of witness sets rather than higher order entities as denotations of QNPs, whereas Cooper (2013a) refines Purver and Ginzburg's account and shows how the RCH can be maintained using a GQ-based perspective. Using the RCH as a methodological principle for positing denotations can be applied straightforwardly in an incremental setting. It offers a stronger constraint than Fregean/Montogovian compositionality which leaves underdetermined which part contributes what—it fulfills the criteria of what Milward (1991) calls *incremental representation* and *strongly incremental interpretation*.

3.4. 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.

KoS's treatment of repair involves two aspects. One is straightforward, drawing on an early insight of Conversation Analysis (Schegloff (2007)), 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 (13a) and will have the form of record (13b):

(13) a. $LocProp =_{def} \begin{bmatrix} \text{sit} : \mathit{Sign} \\ \text{sit-type} : \mathit{RecType} \end{bmatrix}$ 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 Head Driven Phrase Structure Grammar (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 (13b), 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 3.4 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 (14), 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.

(14) 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 (2013b). Starting at the word level—if Lex(T_w, C) is one of the lexical resources available to an agent *A* (e.g., Lex('Beethoven', NP) or Lex('a', Det)) and *A* judges an event *e* to be of type T_w, then *A* is licensed to update their 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 such as route finding. The licensing condition corresponding to lexical resources like (14) is given in (15). We will return below to how this relates to gameboard update. (15) 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.

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

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 activities such as route finding. 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$^\frown$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 type Det$^\frown$N.

The resource that licences this is a rule which modelled as the function in (16a) which we represent as (16b).

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

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

'RuleDaughters' is to be the function in (17). 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:

(17)
$$\begin{array}{l}\lambda T_1 : \mathit{Type} \,.\\ \quad \lambda T_2 : \mathit{Type} \,.\\ \qquad \lambda u : T_1 \,.\, T_2 \boxed{\underset{\cdot}{\wedge}} \left[\text{syn:} \left[\text{daughters=}u\text{:}T_1\right]\right]\end{array}$$

4.3. Semantic Composition using asymmetric merge

As we mentioned in section 3.2.1, 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 (18).

(18) [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-...

Before the first word we assume that the speaker has the question ‘who took the scissors’ (which we denote here with q_0) on his agenda, in the private part of his information state;[5] in his visual field he can see no scissors:[6]

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

$$\begin{bmatrix} \text{private.agenda} = \left\langle \text{ask(s,q}_0\text{)} \right\rangle : \left\langle \mathit{Type} \right\rangle \\ \text{DGB.FACTS} = \left\{ \ldots \neg\exists x In(Vis-sit, x.scissors(x)) \ldots \right\} : \left\{ \mathit{Type} \right\} \end{bmatrix}$$

We assume that an utterance, u_1, of an interrogative NP such as *who* results in the update in (20). 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* as in (20), a function from records that include a person x into propositions involving a predication $P(x)$.

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

P is of type *Pred*, that is (*Ind→RecType*), the type of functions from individuals to record types. The function, *w*, which serves as the incremental content (*cf.* Milward and Cooper, 1994) of *who* is given in (21).[7]

(21) $$w = \lambda P{:}\mathit{Pred}\ .\ \lambda r{:}\begin{bmatrix} \text{x:}\mathit{Ind} \\ \text{c:person(x)} \end{bmatrix}\ .\ P(r.\text{x})$$

Now the updated information state is characterized in (22).

(22) InfState$_1$: T_0 $\boxed{\underset{\cdot}{\wedge}}$

$$\begin{bmatrix} \text{DGB.Pending} = \begin{bmatrix} \text{sit} = u_1 : \mathit{Sit} \\ \text{sit-type}= \begin{bmatrix} \text{phon : who} \\ \text{cont} = w\text{: } (\mathit{Pred}\rightarrow \mathit{WhPQ}) \end{bmatrix} :\mathit{RecType} \end{bmatrix} : \mathit{RecType} \end{bmatrix}$$

We denote the type computed in (22) by T_1. We take content of the verb *took* to be (23a) (ignoring tense) of type (23b). We represent this content as ‘take′’.

(23) a. $\text{take}' = \lambda y{:}\mathit{Ind}\ .\ \lambda x{:}\mathit{Ind}\ .\ \begin{bmatrix}\text{e:take}(x,y)\end{bmatrix}$ b. $(\mathit{Ind}\rightarrow \mathit{Pred})$

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).

[5] This is not a necessary assumption—presumably many utterances are partially planned as their generation starts, hence the occurrence of some filled pauses to buy the speaker planning time.

[6] We assume this visual field is part of the speaker’s DGB, which is again a simplification, since it need not be (quasi)-shared.

[7] 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.

(24) a. $\lambda y{:}\mathit{Ind}\ .\ w(\mathrm{take}'(y))$ b. $\lambda y{:}\mathit{Ind}\ .\ \mathrm{InfoState}_1.\mathrm{DGB.Pending.sit\text{-}type.cont}(\mathrm{take}'(y))$

We abbreviate (24b) as *wt*. We can compute a type for InfState$_2$ as in (25).

(25) $$\mathrm{InfState}_2 : T_1 \boxed{\underset{\cdot}{\wedge}} \begin{bmatrix} \mathrm{DGB.Pending} = \begin{bmatrix} \mathrm{sit} = u_2{:}\mathit{Sit} \\ \mathrm{sit\text{-}type} = \begin{bmatrix} \mathrm{phon} : \text{who took} \\ \mathrm{cont} = \mathit{wt} : (\mathit{Ind} \rightarrow \mathit{WhPQ}) \end{bmatrix} {:}\mathit{RecType} \end{bmatrix} {:}\mathit{RecType} \end{bmatrix}$$

We use T_2 to represent the type computed in (25). J opens the drawer and sees the scissors there. This updates the DGB facts with a fact that the scissors are in the visual field. 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, InfState$_3$, as in (26).

(26) $$\mathrm{InfState}_3 = T_2 \boxed{\underset{\cdot}{\wedge}} \begin{bmatrix} \mathrm{private.agenda} = \left\langle\ \right\rangle : \left\langle \mathit{Type} \right\rangle \\ \mathrm{DGB.FACTS} = \left\{ \ldots \begin{bmatrix} \mathrm{x}{:}\mathit{Ind} \\ \mathrm{c{:}scissors(x)} \\ \mathrm{In(VisSit,x\ )} \end{bmatrix} \ldots \right\} : \left\{ \mathit{Type} \right\} \end{bmatrix}$$

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 (27). The issues that remains is how to explicate T_{chart} in order to understand how incremental content arises.

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

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 *Dudamel*. At this point we will say that the type of the chart is (28)

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

The main event of the chart type (represented by the e-field) breaks the phonological event of type “Dudamel” down into a string of two events, the start and the end of the “Dudamel”-event.[8] Thus (28) records that we have observed an event of the phonological type “Dudamel” and an event consisting of the start of that event followed by the end of that event. Given that we have the resource $\text{Lex}_{\text{PropName}}$(“Dudamel”, *d*) available, we can update (28) to (29):

(29) $$\begin{bmatrix} \mathrm{e}_1 & : & \text{“Dudamel”} \\ \mathrm{e}_2 & : & \text{Lex}_{\text{PropName}}(\text{“Dudamel”}, d)\ \boxed{\dot{\wedge}}\ [\text{s-event:}[\mathrm{e}{=}\mathrm{e}_1\text{:}\mathit{Phon}]\ ] \\ \mathrm{e} & : & \begin{bmatrix}\mathrm{e}_1\text{:start}(\mathrm{e}_1) \\ \mathrm{e}_2\text{:start}(\mathrm{e}_2)\end{bmatrix}{}^\frown\begin{bmatrix}\mathrm{e}_1\text{:end}(\mathrm{e}_1) \\ \mathrm{e}_2\text{:end}(\mathrm{e}_2)\end{bmatrix} \end{bmatrix}$$

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 “Dudamel” 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 $\text{Lex}_{\text{PropName}}$(“Dudamel”, *d*) 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 (30a), which has the type (30b).

(30) a. $S \longrightarrow NP\ VP \mid NP'(VP')$ b. $(NP^\frown VP \rightarrow \mathit{Type})$

Cooper (2016) argues for an analogy between non-linguistic event prediction and the prediction that occurs in parsing.[9] 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 (31).[10]

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

[9] 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.

[10] 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 that would be an appropriate argument to the rule corresponds to what is called a *left-corner* parsing strategy.

(31)
$$\left[\begin{array}{lll} \text{e}_1 & : & \text{“Dudamel”} \\ \text{e}_2 & : & \text{Lex}_{\text{PropName}}(\text{“Dudamel”}, d)\ \boxed{\dot{\wedge}}\ \left[\text{s-event:}\left[\text{e=e}_1{:}\mathit{Phon}\right]\ \right] \\ \text{e}_3 & : & \left[\begin{array}{l} \text{rule=}S \longrightarrow NP\ VP \mid NP'(VP'){:}(NP^\frown VP \rightarrow Type) \\ \text{fnd=e}_2{:}\mathit{Sign} \\ \text{req=}VP{:}\mathit{Type} \\ \text{e:required(req,rule)} \end{array}\right] \\ \text{e} & : & \left[\begin{array}{l} \text{e}_1{:}\text{start}(\text{e}_1) \\ \text{e}_2{:}\text{start}(\text{e}_2) \end{array}\right]^\frown \left[\begin{array}{l} \text{e}_1{:}\text{end}(\text{e}_1) \\ \text{e}_2{:}\text{end}(\text{e}_2) \\ \text{e}_3{:}\text{start}(\text{e}_3)^\frown\text{end}(\text{e}_3) \end{array}\right] \end{array}\right]$$

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.

In what follows, we will adopt a simplified version of (31), exemplified in (32). We will omit the ‘e field’.

(32)
$$\left[\begin{array}{l} \text{e1 : “Dudamel”} \\ \text{e2 : Lex}_{NP}(\text{“Dudamel”})\ \boxed{\dot{\wedge}}\ \left[\text{s-event : }\left[\text{e = e1 : Phon}\right]\right] \\ \text{e3 : } \left[\begin{array}{l} \text{fnd = e2 : Sign} \\ \text{req = } \left[\begin{array}{l} \text{cat = VP : Syncat} \\ \text{cont : (Ind} \rightarrow \text{Prop)} \end{array}\right]\text{: Type} \\ \text{proj = } \left[\begin{array}{l} \text{s-event :fnd.phon }^\frown\text{req.phon} \\ \text{cat = S} \\ \text{cont =req.cont(fnd.cont) : Prop} \end{array}\right]\text{: Type} \end{array}\right] \end{array}\right]$$

5. Incremental Dialogue Processing: principles and examples

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

5.1. Utterance Projection

The first principle we introduce corresponds to the ‘stop option’ in our utterance protocol (14b)—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. (33) 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’.)

(33) **Utterance Projection**

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

We exemplify an incremental view of the latest move that is being moved in (33) with a word-by-word evolution of the latest move, analogous to that in section 4.3, but this time for an initial segment of a declarative utterance: *Jo... saw...*

(34) a.

$$\left[\begin{array}{l} \text{sit} = u_1 \\ \text{sit-type=} \left[\begin{array}{l} \text{phon} : \text{“Jo”} \\ \text{dgb-params} : \left[\begin{array}{l} \text{j} : \textit{Ind} \\ \text{s0} : \textit{Rec} \end{array}\right] \\ \text{cont} = \lambda P{:}\textit{Pred}\ .\ \left[\begin{array}{l} \text{sit} = s_0 \\ \text{sit-type} = \left[\text{c1}{:}P(\text{j})\right] \end{array}\right] : (\textit{Pred}\rightarrow\textit{Prop}) \end{array}\right] \end{array}\right]$$

b.

$$\left[\begin{array}{l} \text{sit} = u_2 \\ \text{sit-type=} \left[\begin{array}{l} \text{phon} : \text{“Jo saw”} \\ \text{dgb-params} : \left[\begin{array}{l} \text{j} : \textit{Ind} \\ \text{s0} : \textit{Rec} \end{array}\right] \\ \text{cont} = \lambda x{:}\textit{Ind}\ .\ \left[\begin{array}{l} \text{sit} = \text{s0} \\ \text{sit-type} = \left[\text{c1:Saw(j,}x\text{)}\right] \end{array}\right] : (\textit{Ind}\rightarrow\textit{Prop}) \end{array}\right] \end{array}\right]$$

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 (35)—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:

(35) Forward Looking Utterance Rule:

$$\left[\begin{array}{l} \text{preconds} = \left[\begin{array}{l} \text{spkr} : \mathit{Ind} \\ \text{addr} : \mathit{Ind} \\ \text{pending.sit-type} : \left[\begin{array}{l} \text{fnd} : \mathit{Sign} \\ \text{req:}\ \mathit{Sign} \end{array}\right] \end{array}\right] \\ \text{effects} = \mathit{TurnUnderspec}\ \boxed{\underset{\cdot}{\wedge}}\ \left[\begin{array}{l} \text{MaxQud} = \\ \left[\begin{array}{l} \text{q} = \lambda x{:}\mathit{Ind}\ .\ \text{MeanNextUtt}(r^{*}.\text{spkr}, r^{*}.\text{fnd}, x) \\ \text{fec} = \left\{\right\} \end{array}\right] : \mathit{InfoStruc} \\ \text{LatestMove} : \mathit{LocProp} \\ \text{c2: Copropositional}(\text{LatestMove}^{content}, \text{MaxQud}) \end{array}\right] \end{array}\right]$$

A consequence of (35), is that it offers the potential to explain cases like (36). 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 (35) actually gets uttered:

(36) 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. (2016), who also discuss cross-linguistic variation with SAQs in English, Chinese, and Japanese.

5.3. Prediction and Clarification for incomplete utterances

We return now to (2a,b), repeated here as (37):

(37) a. A(i): John . . . Oh never mind. B(ii): What about John? A: He's a lovely chap but a bit disconnected. / # burnt himself while cooking last night.

b. A(i): John . . . Oh never mind. B(ii): John what? A: burnt himself while cooking last night. / # He'ss a lovely chap but a bit disconnected.

Whether (2a) or (2b) arise depends on whether one uses utterance projection or the forward looking utterance rule. For the former, as we showed in (34), an initial referential NP when prediction is applied results in (roughly) the projected content in (38). Thus, given the conversational rule *QSPEC* (the rule (9) above), B's follow up questions are justified as seeking elaboration of the existentially quantified proposition $\exists P\ \text{IllocRel}(spkr, P(j))$:

(38) LatestMove = $\begin{bmatrix} \text{sit} = u_1 & \\ \text{sit-type}= & \begin{bmatrix} \text{phon : “John”} \\ \text{dgb-params :} \begin{bmatrix} \text{j} : \mathit{Ind} \\ \text{s0} : \mathit{Rec} \end{bmatrix} \\ \text{cont} = \lambda P{:}\mathit{Pred}\ . \begin{bmatrix} \text{sit} = \text{s0} \\ \text{sit-type} = \begin{bmatrix} \text{c1}{:}P(\text{j}) \end{bmatrix} \end{bmatrix} : (\mathit{Pred}{\to}\mathit{Prop}) \end{bmatrix} \end{bmatrix}$

As for (2b), this follows by applying the forward looking utterance rule, where the addressee takes over.

5.4. Sluicing, incrementally

We assume, following Cooper (2013a) that a QNP such as ‘someone’ has a content of the form (39), where *q-params* constitute descriptive content that, in contrast to the dgb-params, does not require instantiation.

(39) $\begin{bmatrix} \text{q-params:} \begin{bmatrix} \text{restr = person: Ppty} \\ \text{witness : } \exists(\text{restr}) \end{bmatrix} \\ \text{P : Ppty} \\ \text{cont} = \begin{bmatrix} \text{scope = P :Ppty} \\ \text{c1 = witness : } \exists(\text{restr,scope}) \end{bmatrix} : \text{Rtype} \end{bmatrix}$

We assume a constructional specification for a sluice as in (40), deriving from Ginzburg (2012). A sluice denotes a question (i.e., a function from records into propositions) whose domain is the type denoted by the *wh*-phrase and whose range is that given by MaxQUD’s proposition where the *wh*-phrase’s variable is substituted for that associated with the antecedent:

(40) sluice-int-cl.cont = (whP.rest)MaxQUD.prop[antecedent.x $\mapsto$whP.x]

The sluice is triggered by utterance prediction that LatestMove is *A asserts that Someone P’ed.* This gives rise to QUD update, via Assertion QUD-incrementation with (41a) as maximal element of QUD and the antecedent for a sluice, as in (41b), which is predicted to mean (41c) immediately after it is uttered:

(41) a. $?\exists x, P[\text{Person}(x) \wedge \text{P}(x)]$
b. A: Someone ... B: Who?
c. ‘Who is that person (that has some as yet uninstantiated property)?’

6. Conclusions and further Work

In this paper we provide data related to the potential for clarification, repair, and sluicing in mid-utterance. This data shows 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. In particular, this data constitutes an argument for incremental access to the contextual repository QUD. 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

Barwise, J. (1989). *The Situation in Logic*. CSLI Lecture Notes. Stanford: CSLI Publications.

Clark, H. (1996). *Using Language*. Cambridge: Cambridge University Press.

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

Cooper, R. (2013a). Clarification and generalized quantifiers. *Dialogue and Discourse 4*, 1–25.

Cooper, R. (2013b). Update conditions and intensionality in a type-theoretic approach to dialogue semantics. In *Proceedings of SemDial 2013 (DialDam)*, pp. 15–24. Citeseer.

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

Cooper, R., S. Dobnik, S. Larsson, and S. Lappin (2015). Probabilistic type theory and natural language semantics. *Linguistic Issues in Language Technology 10*(4).

Cooper, R. and J. Ginzburg (2015). Type theory with records for natural language semantics. In C. Fox and S. Lappin (Eds.), *Handbook of Contemporary Semantic Theory, 2nd edition*. Oxford: Blackwell.

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

Gardent, C. and M. Kohlhase (1997). Computing parallelism in discourse. *IJCAI*, 1016–1021.

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

Ginzburg, J. (2012). *The Interactive Stance: Meaning for Conversation*. Oxford: Oxford University Press.

Ginzburg, J., R. Fernández, and D. Schlangen (2014). Disfluencies as intra-utterance dialogue moves. *Semantics and Pragmatics 7*(9), 1–64.

Ginzburg, J. and M. Purver (2012). Quantification, the reprise content hypothesis, and type theory. In L. Borin and S. Larsson (Eds.), *From Quantification to Conversation: Festschrift for Robin Cooper on the occasion of his 65th Birthday*, pp. 85–110. College Publications. This paper appeared in an online version of this collection in 2008.

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

Hough, J., C. Kennington, D. Schlangen, and J. 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, D. and J. H. Martin (2009). *Speech and Language Processing* (2nd ed.). New Jersey: Prentice Hall.

Kempson, R., R. Cann, E. Gregoromichelaki, and S. Chatzikyriakidis (2016). Language as mechanisms for interaction. *Theoretical Linguistics 42*(3-4), 203–276.

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

Larsson, S. (2002). *Issue based Dialogue Management*. Ph. D. thesis, Gothenburg University.

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

Merchant, J. (2001). *The Syntax of Silence*. Oxford: Oxford University Press.

Milward, D. (1991). *Axiomatic Grammar, Non-Constituent Coordination and Incremental Interpretation*. Ph. D. thesis, University of Cambridge.

Milward, D. and R. Cooper (1994). Incremental interpretation: Applications, theory, and relationship to dynamic semantics. In *Proceedings of the 15th conference on Computational linguistics-Volume 2*, pp. 748–754. ACL.

Peldszus, A. and D. Schlangen (2012). Incremental construction of robust but deep semantic representations for use in responsive dialogue systems. In *Proceedings of the Coling Workshop on Advances in Discourse Analysis and its Computational Aspects*.

Pollard, C. and I. 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, M., A. Eshghi, and J. Hough (2011). Incremental semantic construction in a dialogue system. In J. Bos and S. Pulman (Eds.), *Proceedings of the 9th IWCS*, pp. 365–369.

Purver, M. and J. Ginzburg (2004). Clarifying noun phrase semantics. *Journal of Semantics 21*(3), 283–339.

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

Schegloff, E. (2007). *Sequence Organization in Interaction*. Cambridge: Cambridge University Press.

Schlangen, D. and G. Skantze (2009). A general, abstract model of incremental dialogue processing. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 710–718. Association for Computational Linguistics.

Schlangen, D. and G. 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*(9–10), 1213–1234.

Tian, Y., T. Murayama, and J. Ginzburg (2016). Hesitation markers and self addressed questions. *Journal of Psycholinguistic Research*.

Vallduví, E. (2015). Information structure. In M. Aloni and P. Dekker (Eds.), *The Cambridge Handbook of Semantics*. Cambridge: Cambridge University Press.