incrementality and clarification/sluicing potential

Jonathan Ginzburg, Julian Hough, Robin Cooper, David Schlangen
September 6, 2016, Sinn und Bedeutung 2016, Edinburgh University
supported by the Disfluency, Exclamations, and Laughter in Dialogue (DUEL) project within the projets franco-allemand en sciences humaines et sociales funded by the ANR and the DFG
Why Semantics needs Incrementality: the Data

Background

Incremental Grammar as Incremental dialogue processing

References
• 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 & Bornkessel, 2004) and as an important feature for design of spoken dialogue systems (see e.g., Schlangen & Skantze, 2009; Hough et al., 2015).

• Nonetheless, with a few important exceptions (see e.g., Gregoromichelaki & Kempson, 2013), incrementality is viewed as an aspect of performance, not semantic competence.

• Moreover, it seems to entail giving up on compositionality as a constraining principle on denotations.
In this paper:

1. We point to a variety of dialogical phenomena whose analysis incontrovertibly requires a semantics formulated in incremental terms.

2. These include cases, with clarification requests, self-corrections, and sluicing, that call into question existing assumptions about ellipsis resolution and argue for incremental updating of QUD.
3. The incremental semantic framework we sketch improves on existing such accounts (reviewed in Peldszus & 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 & Ginzburg, 2004; Ginzburg & Purver, 2012; Cooper, 2013,
— embedded within independently motivated dialogue states Ginzburg, 2012.
(1a) 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 ([1]); (1bi), an instance of an ‘abandoned’ utterance (or fresh start, Levelt, 1983), licenses reactions such as (1bii) and (1c):

   c. A: Bill is . . . B: Yeah don’t say it, we know.
(2a,b) exemplify a contrast between two reactions to an ‘abandoned’ utterance:

(2) 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.

(3) is an attested example of an abandoned utterance in mid-word:

(3) a. [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-...
why semantics needs incrementality: the data IV

(4a) 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;

(4b) . . .

(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)
(5) 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);
   e. A: Someone I’m not saying who / B: No, do say/Who?
• (6a) requires us to be able to write a lexical entry for ‘Aha’ and ‘yeah’ and ‘mmh’ (and their counterparts cross linguistically, e.g., French: ouais, mmh, . . . ,) whose context is/includes “an incomplete utterance”.

• (6b,c,d) requires us to associate a content with A’s incomplete utterance which can either trigger an elaboration query (6b), a query about utterance completion (6c), or an acknowledgement.

   d. A: Bill is . . . B: Yeah don’t say it, we know.
(7) requires us to integrate within-utterance and dialogue context processing:

(7) a. [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-... (complete)
why semantics needs incrementality: initial spec III

- (8a) 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:

(8)  a. 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)

b. Oh I know especially if you get, what is it, Seclor i think that that just about breaks the bank right there. (sw2292B-ms98)
(9) requires us to enable either incomplete argument frames or QNPs immediately after their utterance to raise sluiced QUDs:

(9) a. The translation is by—who else? —Doris Silverstein (The TLS, Feb 2016)
b. He saw—can you guess who?—The Dude;
incremental semantics summary of initial spec I

- Context specified as (including) “an incomplete utterance” → type specification within dialogue context
- (1b-d) require us to associate a content with A’s incomplete utterance which can either trigger an elaboration query, a continuation query, or an acknowledgement. → operations on dialogue context
- incomplete argument frames or QNPs immediately after utterance raise sluiced QUDs → inference rule on dialogue context
Why Semantics needs Incrementality: the Data

Background

Incremental Grammar as Incremental dialogue processing

References
starting point: the dgb I


- A cognitive architecture in which there is no single common ground, but distinct yet coupled Dialogue GameBoards, one per conversationalist.
## Dialogue Gameboard

<table>
<thead>
<tr>
<th>Component</th>
<th>Type</th>
<th>Keeps track of</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spkr: Individual</td>
<td>Addressee: Individual</td>
<td>Turn ownership</td>
</tr>
<tr>
<td>Facts: Set(propositions)</td>
<td></td>
<td>Shared assumptions</td>
</tr>
<tr>
<td>VisualSit: Situation</td>
<td></td>
<td>Visual scene</td>
</tr>
<tr>
<td>Moves: List(Locutionary propositions)</td>
<td></td>
<td>Grounded utterances</td>
</tr>
<tr>
<td>QUD: partially ordered set(&lt;question,fec&gt; pairs)</td>
<td>fec=focus-establishing-constituent</td>
<td>Live issues</td>
</tr>
<tr>
<td>Pending: List(Locutionary propositions)</td>
<td></td>
<td>Ungrounded utterances</td>
</tr>
</tbody>
</table>
The logical underpinnings of our dialogue framework KoS is Type Theory with Records (TTR).


It can be used to model semantic ontologies, interaction, and to write dialogically-oriented grammars.
the type theoretic universe

(10) Type Theoretic World (Cooper, 2012, simplified)

TYPEDWORLD = ⟨Type^n, BasicType, ComplexType^n, RecType^n, ⟨A, F^n⟩⟩

a. Type^n is the set of types of order n, built up recursively using type construction operations.
b. BasicType: IND, TIME, LOC, . . .
c. ComplexType^n: tuples consisting of entities [from A] and predicates.
d. RecType^n: set of records, record types defined with respect to a set of objects used as labels.
e. ⟨A, F^n⟩ is a model (A^n assigning entities to BasicType and F^n assigns tuples to ComplexType^n).
As a means of tightly constraining semantic denotations, we adopt the Reprise Content Hypothesis (RCH) Purver & Ginzburg, 2004; Ginzburg & Purver, 2012; Cooper, 2013.

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.)
the reprise content hypothesis and generalized quantifiers

- Purver & Ginzburg, 2004; Ginzburg & Purver, 2012 use such data to argue in favour of witness sets rather than higher order entities as denotations of QNPs, whereas Cooper, 2013 refines P&G’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.
• 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*.

• 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 $q$).
conversational rules II

- **QSPEC**

\[
\begin{align*}
\text{pre:} & \quad \text{qud} = \langle i, l \rangle: \text{poset}(\text{InfoStruc}) \\
\text{effects:} & \quad \text{TurnUnderspec} \\
& \quad r: \text{AbSemObj} \\
& \quad R: \text{IllocRel} \\
& \quad \text{LatestMove} = R(\text{spkr, addr, r}) : \text{IllocProp} \\
& \quad c1: \text{Qspecific}(r, i.q)
\end{align*}
\]
Ginzburg, 2012: What object needed to represent utterances in dialogue context in their immediate aftermath?

Two essential branches:

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

Hence, need an entity off of which both ‘shared understanding’ (grounding conditions, Clark, 1996) AND clarification potential can be read.
● Concretely . . .

● Utterances are kept track of in a contextual attribute PENDING in the immediate aftermath of the speech event.

● Given a presupposition that $u$ is the most recent speech event and that $T_u$ is a grammatical type that classifies $u$, a record of the form

$$\begin{bmatrix}
\text{sit} = u \\
\text{sit-type} = T_u
\end{bmatrix}$$

(of type LocProp (locutionary proposition)), gets added to PENDING.
grounding and clarification interaction

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

- If \( \text{Lex}(T_w, C) \) is one of the lexical resources available to an agent \( A \) (e.g., \( \text{Lex}("Beethoven", \text{NP}) \) or \( \text{Lex}("a", \text{Det}) \) ) and \( A \) judges an event \( e \) to be of type \( T_w \), then \( A \) is licensed to update their gameboard with the type \( \text{Lex}(T_w, C) \).

- Phrase structure rules as functions from strings of sign types to a sign type (that is, dependent types returning a type of sign)

- A string of type \( \text{Det} \triangleleft \text{N} \) (that is, a concatenation of an event of type \( \text{Det} \) and an event of type \( \text{N} \) ) can lead us to the conclusion that we have observed a sign of type \( \text{NP} \) whose daughters are of the type \( \text{Det} \triangleleft \text{N} \).

- Formally:
  \[
  \lambda u : \text{Det} \triangleleft \text{N}. \text{NP} \left[ \text{cat:daughters=}u : \text{Det} \triangleleft \text{N} \right]
  \]
Why Semantics needs Incrementality: the Data

Background

Incremental Grammar as Incremental dialogue processing

References
The account of MCI relevance discussed earlier extended to self-repair (Ginzburg, Fernández and Schlangen 2014 ‘Disfluencies as intra-utterance moves’ Semantics and Pragmatics, 7(9)1–64).

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 $u_1$?) or what is still to come (e.g. what word does the speaker mean to utter after sub-utterance $u_2$?).

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 provided by self-addressed questions (*She saw the ... what’s the word?*, *Je suis comment dire?*), as explained below.

Thus, 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.

1. Ground: continue (Levelt, 1983)
2. Predict: stop since content is predicatable
3. (Self)Clarify: generate (self)CR given lack of expected utterance
The denotation associated with the root of the tree is an illocutionary proposition, hence compatible with declarative, interrogative, imperative etc utterances.

This gets refined as each word gets introduced using an operation of asymmetric merge of record types Cooper, 2012; Hough, 2015.

**asymmetric merge**: given two record types $R_1$ and $R_2$, $R_1 \triangledown R_2$ will yield a record type which is the union of all fields with labels not shared by $R_1$ and $R_2$ and the asymmetric merge of the remaining fields with the same labels, whereby $R_2$’s type values take priority over $R_1$’s fields, yielding a resulting record type with $R_2$’s fields only in those cases.
This enables us to effect a combinatory operation that synthesises function application and unification.

Generalizing this to KOS's update rules we have the following update process for a dialogue context $C$ and for some rule record type $R$:
if \( C : \begin{bmatrix}
\text{private:} & [] \\
\text{shared:} & [] 
\end{bmatrix} \) and \( C : R.\text{pre} \),

then \( R \) can apply to \( C \) such that \( C \) is updated:

\[
C := C \bigwedge R.\text{effects}
\]

- The updates operate on various levels of information which can be arbitrarily fine-grained (even phonetic). This gives us the requisite apparatus for incrementality.
semantic composition using asymmetric merge IV

- sketch of *who took the sci*—? example
semantic composition using asymmetric merge V

• **Before first word:** \( \text{InfState}_0 = \)
  
  \[
  \begin{align*}
  \text{private.agenda} &= \langle \text{ask}(s,q_0) \rangle \\
  \text{DGB.FACTS} &= \{ \ldots \neg \exists In(Vis \leftarrow \text{sit}, x.\text{scissors}(x)) \ldots \}
  \end{align*}
  \]

• **After who:** \( \text{InfState}_1 = \text{InfState}_0 \quad \land \quad e_1 : \text{who=} \)
  
  \[
  \begin{align*}
  \text{private.agenda} &= \langle \text{ask}(s,q_0) \rangle \\
  \text{DGB.FACTS} &= \{ \ldots \neg \exists In(VisSit, x.\text{scissors}(x)) \ldots \}
  \end{align*}
  \]

\[
\begin{align*}
\text{DGB.Pending} &= \left[ \\
\text{sit} &= e_1 \\
\text{sit-type} &= \text{phon} : \text{who} \\
\text{P:Pred} &= \text{cont} = ( \left[ x:\text{Ind} \right] ) \left[ c1:P(r.x) \right]
\end{align*}
\]
• After *took*: $\text{InfState}_2 = \text{InfState}_1 \bigwedge e_2 : \text{took} =$

\[
\begin{align*}
\text{private.agenda} &= \langle \text{ask}(s,q_0) \rangle \\
\text{DGB.FACTS} &= \{ \ldots \neg \exists \text{In}(\text{VisSit},x.\text{scissors}(x)) \ldots \} \\
\text{sit} &= e_2 \\
\text{DGB.Pending} &= \begin{cases} \\
\text{sit-type} = & \text{phon} : \text{who took} \\
\text{cont} = & (x:\text{Ind} \\
& c_1:\text{person}(x)) \rightarrow [c_1:T(r.x,y)] \\
\end{cases}
\end{align*}
\]

• After opening the drawer:
\[ \text{InfState}_3 = \text{InfState}_2 \bigwedge e_3 : \begin{bmatrix} x: \text{Ind} \\ c: \text{scissors}(x) \\ \text{See}(s, x) \end{bmatrix} = \]

\[
\begin{cases}
\text{private.agenda} = \langle \text{ask}(s, q_0) \rangle \\
\text{DGB.FACTS} = \left\{ \ldots \begin{bmatrix} x: \text{Ind} \\ c: \text{scissors}(x) \end{bmatrix} \ldots \right\} \\
\text{DGB.Pending} = \begin{bmatrix} \text{sit} = e_2 \\ \text{sit-type} = \end{bmatrix} \begin{bmatrix} \text{phon} : \text{who took} \\ y: \text{Ind} \\ \text{cont} = (\begin{bmatrix} x: \text{Ind} \\ c1: \text{person}(x) \end{bmatrix})[c1: T(r, x, y)] \end{bmatrix} 
\end{cases}
\]
• (via constraint forbidding posing resolved questions) InfState$_4$

$\text{private.agenda} = \langle \rangle$

$\text{DGB.FACTS} = \{ \ldots [x: \text{Ind}, c: \text{scissors}(x)] \ldots \}
\{ \ln(\text{VisSit}, x) \}$

$\text{DGB.Pending} = \begin{cases} 
\text{sit} = \text{e2} \\
\text{sit-type} = \begin{cases} 
\text{phon : who took} \\
y: \text{Ind} \\
\text{cont} = ( [x: \text{Ind}]
[\text{c1: person}(x)] ) \text{[c1: T}(r.x, y)]
\end{cases}
\end{cases}$
Pending and charts I

- Pending information on the dialogue gameboard includes a chart-type representing the agent's view of the parse chart during the progress of the utterance.

- For Pending the type remains LocProp

\[
\begin{bmatrix}
sit = l \\
sit-type = T_{chart}
\end{bmatrix}
\]

- We focus on $T_{chart}$ to understand how incremental content arises
• charts as used in computational parsing systems can be updated incrementally for each word.

• they can represent several live possibilities in a single data structure

• a chart can encode that we have heard a NP followed by a transitive verb and make a prediction that there will be an object NP without requiring that the subject and the verb be combined into a single constituent

• charts can be modelled as record types facilitating reasoning about speech events already perceived to type of speech events expected in the continuation
charts: a simplified example

\[
\begin{align*}
e_1 & : \text{"Jo"} \\
e_2 & : \text{Lex}_{NP}(\text{"Jo")} \land \left[ \text{s-event} : e = e_1 : \text{Phon} \right] \\
f\text{nd} & = e_2 : \text{Sign} \\
\text{req} & = \begin{cases} 
\text{cat} = \text{VP} : \text{Syncat} \\
\text{cont} : (\text{Ind} \rightarrow \text{Prop}) 
\end{cases} : \text{Type} \\
e_3 & : \begin{cases} 
\text{s-event} : f\text{nd}.\text{phon} \bowtie \text{req}.\text{phon} \\
\text{proj} = \begin{cases} 
\text{cat} = S : \text{Syncat} \\
\text{cont} = \text{req}.\text{cont}(f\text{nd}.\text{cont}) : \text{Prop} 
\end{cases} : \text{Type}
\end{cases}
\end{align*}
\]
Chart projection I

- Basic rule:

\[
(11) \begin{cases}
\text{preconds} : [\text{pending.sit-type.proj} = a : \text{Type}] \\
\text{effects} : \text{TurnUnderspec} \land_{\text{merge}} \\
\quad e_1 : \text{sign} \\
\quad \text{LatestMove} = [\text{sit} = e_1] : \text{LocProp}
\end{cases}
\]
chart projection II

- After: A: Jo ...

(12) LatestMove =

\[
\begin{bmatrix}
\text{sit} = u1 \\
\text{sit-type} = \\
\text{phon} : Jo \\
\text{P:Pred} \\
\text{dgb-params} : \\
\text{cont} = \\
\text{sit-type} = \\
\end{bmatrix}
\begin{bmatrix}
\text{sit} = s0 \\
\text{c1:P(j)} \\
\end{bmatrix}
\]
chart projection III

- After A: Jo saw ...

(13) LatestMove =

\[
\begin{align*}
\text{sit} &= u1 \\
\text{phon} : Jo & \\
x : \text{Ind} & \\
\text{dgb-params} : & \\
\text{cont} = & \\
\text{sit-type} &
\end{align*}
\]

\[
\begin{align*}
\text{sit-type} & = [c1: \text{Saw}(j, x)] \\
\text{sit} & = s0 \\
\text{sit-type} & = [c1: \text{Saw}(j, x)] \\
\end{align*}
\]
forward-looking disfluencies

- *forward-looking* disfluencies: 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).

(14) a. Show flights arriving in uh Boston. (Shriberg 1994)  
    b. And also the- the dog was old. (Besser and Alexandersson (2007))
forward-looking disfluencies

- FLDs involve the update rule in 15—given a context where the LatestMove is a forward looking editing phrase by A, 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:
(15) Forward Looking Utterance Rule:

preconds:
\[
\begin{cases}
\text{spkr} : \text{Ind} \\
\text{addr} : \text{Ind} \\
\text{pending.sit-type} : [\text{fnd} : \text{sign}, \text{req} : \text{sign}]
\end{cases}
\]

effects: \text{TurnUnderspec} \land \text{merge}

MaxQud =
\[
\begin{cases}
q = \lambda x \text{MeanNextUtt}(\text{pre.spkr}, \text{pre.fnd}, x) \\
fec = \emptyset
\end{cases}
\]

LatestMove : \text{LocProp}

c2: \text{Copropositional}(\text{LatestMove}^{\text{content}}, \text{MaxQud})
• we can propose a lexical entry for continuative particles like ‘mmh’ or ‘yeah’, as in (16).
• The crucial assumption here is that Pending is also associated with incomplete utterances.

(16)
\[
\begin{align*}
\text{cat} = \text{interjection} : \text{syncat} & \\
\text{dgb-params} : & \\
\text{spkr} : \text{IND} \\
\text{addr} : \text{IND} \\
\text{MaxPending} : \text{LocProp} \\
\text{presupp1} : \text{address(addr,spkr,MaxPending)} & \\
\text{cont} = & \left[ c1 : \text{Understand(spkr,addr,MaxPending)} \right] : \text{RecType}
\end{align*}
\]


• Whether (17a) or (17b) arise depends on whether one uses projection or the forward–looking utterance rule.
As we showed earlier, on our account, a bare referential NP when prediction is applied, as exemplified in (17), results in (roughly) the content in (18):

Thus, given the conversational rule \( QSPEC \) B’s follow up questions are justified as seeking elaboration of the existentially quantified proposition \( \exists Q I l l o c R e l (s p k r, Q(j)) \):

\[
(18) \quad \begin{bmatrix}
S.\text{cont} = R(spk r, P) : \text{Illocprop} \\
Q : \text{Pred} \\
\text{dgb-params: } [spkr : \text{Ind}] \\
P = Q(j)
\end{bmatrix}
\]
(19) A: Jo uh B: Jo what?

- This follows by applying the FLD rule, where the addressee takes over

- Input: \[
\begin{align*}
\text{e1 : "Jo"} \\
\text{e2 : } \text{Lex}_{NP}("Jo") \land \left[ \text{s-event : } \left[ e = e1 : \text{Phon} \right] \right] \\
\text{fnd} = e2 : \text{Sign} \\
\text{req} = \left[ \begin{array}{c}
\text{cat} = \text{VP} : \text{Syncat} \\
\text{cont} : (\text{Ind} \rightarrow \text{Prop})
\end{array} \right] : \text{Type} \\
\text{e3 : } \left[ \begin{array}{c}
\text{s-event} : \text{fnd.phon} \sim \text{req.phon} \\
\text{proj} = \left[ \begin{array}{c}
\text{cat} = \text{S} : \text{Syncat} \\
\text{cont} = \text{req.cont} (\text{fnd.cont}) : \text{Prop}
\end{array} \right] : \text{Type}
\end{align*}
\]

- Output: What utterance was to follow ‘Jo’?
sluicing, incrementally I


- We assume a constructional specification for a sluice as in (21d), deriving from (Ginzburg, 2012):

(21) \[
\text{sluice-int-cl.cont} = \\
(\text{whP.rest}\text{MaxQUD.prop}[\text{antecedent}.x \mapsto \text{whP}.x] \\
\]

- The sluice denotes a question (i.e., a function from records into propositions) whose domain is the type denoted by the \textit{wh}-phrase and whose range is that given by MaxQUD’s proposition where the \textit{wh}-phrase’s variable is substituted for that associated with the antecedent.
We assume, following Cooper, 2013 that a QNP such as ‘Someone’ has a content of the form (22a), where q-params constitute descriptive content that, in contrast to the dgb-params, does not require instantiation.

Thus, applying prediction to (22b) will lead to roughly (22c) as maximal element of QUD and the antecedent for a sluice.

(22) a. \[
\begin{align*}
\text{q-params:} & \quad \text{restr} = \text{person: Ppty} \\
\text{witness:} & \quad \exists (\text{restr}) \\
\end{align*}
\]
\[
\begin{align*}
P : \text{Ppty} \\
\text{cont} = & \quad \text{scope} = P : \text{Ppty} \\
& \quad c1 = \text{witness: } \exists (\text{restr}, \text{scope}) : \text{Rtype}
\end{align*}
\]

b. A: Someone— B: who?

c. QUD: \(?\exists x, P[\text{Person}(x) \land P(x)]\)
More in detail: The sluice is triggered by prediction that latest-move is "A asserts that Someone P’ed" which then gives rise to QUD update, after which normal sluice occurs.
• Data that shows the need for incrementality in “competence grammar”
• In particular, an argument for incremental QUD.
• The approach has parallels to Dynamic Syntax (Kempson, Meyer-Viol, & Gabbay, 2001), and particularly recent dialogue-friendly versions (Purver, Eshghi, & Hough, 2011). The central idea is online, incremental construction of meaning representations.
conclusions and further work II

- However, this incremental account 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.
• experimental work on incremental sluicing
Many Thanks


