CI via DTS

Daisuke Bekki[1,2,3(✉)] and Elin McCready[4]

[1] Graduate School of Humanities and Sciences, Ochanomizu University, 2-1-1 Ohtsuka, Bunkyo-ku, Tokyo 112-8610, Japan
[2] National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
[3] CREST, Japan Science and Technology Agency, 4-1-8 Honcho, Kawaguchi, Saitama 332-0012, Japan
bekki@is.ocha.ac.jp
[4] Aoyama Gakuin University, 4-4-25 Shibuya, Shibuya-ku, Tokyo 150-8366, Japan

Abstract. It has been observed that conventionally implicated content interacts with at-issue content in a number of different ways. This paper focuses on the existence of anaphoric links between content of these two types, something disallowed by the system of Potts (2005), the original locus of work on these issues. The problem of characterizing this interaction has been considered by a number of authors. This paper proposes a new system for understanding it in the framework of Dependent Type Semantics. It is shown that the resulting system provides a good characterization of how "cross-dimensional" anaphoric links can be supported from a proof-theoretic perspective.

1 Conventional Implicatures

Conventional implicature (CI) is a kind of pragmatic content first discussed by Grice [7], which is taken to be (one part of the) nonasserted content conveyed by particular lexical items or linguistic constructions. Examples include appositives, non-restrictive relative clauses (NRRCs), expressive items, and speaker-oriented adverbs. Such content has been a focus of a great deal of research in linguistics and philosophy since the work of Potts [19]. According to Potts (who takes a position followed by much or most subsequent research), CIs have at least the following characteristics:

(1) a. CI content is independent from at-issue content (in the sense that the two are scopeless with respect to each other)
  b. CIs do not modify CIs.[1]
  c. Presupposition filters do not filter CIs

Our sincere thanks to the anonymous reviewers of LENLS11 who gave us insightful comments. Elin McCready and Daisuke Bekki are partially supported by a Grant-in-Aid for Scientific Research (C) (No. 25370441) from the Ministry of Education, Science, Sports and Culture. Daisuke Bekki is partially supported by JST, CREST.

[1] Though see footnote 3 for some necessary qualification.

T. Murata et al. (Eds.): JSAI-isAI 2014 Workshops, LNAI 9067, pp. 23–36, 2015.
DOI: 10.1007/978-3-662-48119-6_3

Potts models these features in a two-dimensional semantics for CIs in which CIs are associated with special semantic types. First, since CI content enters a dimension of meaning distinct from that of at-issue content, no scope relations are available, modeling (1a); characteristic (1b) follows from a lack of functional types with CI inputs in the type system; placing filters in the at-issue dimension also accounts for (1c). Although this system has been criticized for various reasons, it seems to be adequate for modeling the basic data associated with CIs.

2 Problem: Interaction Between At-Issue and CI Content

Potts's two-dimensional semantics, which utilizes distinct and dimensionally independent representations for at-issue and CI content, aims to capture their supposed mutual semantic independence. However, a fully separated multidimensional semantics is not fully satisfactory from an empirical perspective, as can be seen by focusing attention on the interaction between CIs and anaphora/presupposition (as also noted by Potts himself). In particular, the following facts are problematic for a treatment in which no interdimensional interaction is allowed:

1. CI content may serve as antecedent for later anaphoric items and presupposition triggers, meaning that discourse referents introduced in CI contexts are accessible to anaphora/presupposition triggers, as exemplified in the mini-discourse (2) ((3.15b) in [19], slightly simplified).
2. Anaphora/presupposition triggers introduced in CI environments may find their antecedents in the preceding discourse (their *local contexts*), i.e. anaphora/ presupposition triggers inside a conventional content require access to their left contexts, as exemplified in the mini-discourse (3) (see also [25].).

(2) a. Mary counseled John, who killed a coworker.
b. Unfortunately, Bill knows that he killed a coworker.

(3) a. John killed a coworker.
b. Mary, who knows that he killed a coworker, counseled him.

In both (2) and (3), the factive presupposition "he (=John) killed a coworker" can be bound by the antecedent in the first sentence. This behaviour of CIs fails to be explained by Potts's [19] analysis, where at-issue content and CI content are fully independent of each other, at least in their sentential representations. With regard to the cases such as (2), at minimum we require a mechanism to collect the CIs hanging in a sentential tree, and pass them to the succeeding discourse, where they can play the role of antecedents. In order to deal with the cases like (3), we also need a mechanism to pass the local context of a sentence to the collection of CIs which has been collected from it. (Here we assume an analysis which collects CIs from the (syntactic or semantic) tree in a Pottsian style, putting aside the arguments about compositionality raised by

Gutzmann [10] and others.) Neither extension, however, seems straightforward in Potts's [19] framework, nor in other frameworks that have been proposed for the analysis of CIs, but which have not attempted to account for the present set of phenomena (excluding theories using dynamic semantics, such as that of AnderBois et al. (2014) or Nouwen [18]).

3 Dependent Type Semantics

Dynamic solutions to these puzzles exist, such as the work of AnderBois et al. cited above; here, we take a different line, and propose a compositional analysis of conventional implicatures in the framework of Dependent Type Semantics (DTS; Bekki(2014)). DTS is based on dependent type theory (Martin-Löf [16]), Coquand and Huet [6]) which provides a proof-theoretic semantics in terms of the Curry-Howard correspondence between types and propositions, following the line of Sundholm [24]. This approach has been proved useful for linguistic analysis, especially in Ranta's [20] Type Theoretical Grammar and its successors. Krahmer and Piwek [14] found that anaphora resolution and presupposition binding/accommodation can be reduced to *proof search* (which is known as the "anaphora resolution as proof construction" paradigm). Bekki's [5] DTS inherits this paradigm and reformalizes the whole setting in a compositional manner; the resulting system can serve as the semantic component of any lexical grammar.

For example, the semantic representation of a classical relative donkey sentence as (4a) is calculated as (4b) in DTS.

(4) a. Every farmer who owns a donkey beats it_1.

b. $$\lambda c. \left(u{:} \begin{bmatrix} x{:}\mathbf{entity} \\ \begin{bmatrix} \mathbf{farmer}(x) \\ \begin{bmatrix} y{:}\mathbf{entity} \\ \begin{bmatrix} \mathbf{donkey}(y) \\ \mathbf{own}(x,y) \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \right) \rightarrow \mathbf{beat}(\pi_1(u), @_1(c,u))$$

The semantic representation (4b) for the sentence (4a) contains an *underspecified term* $@_1$, which corresponds to the referent of the pronoun "it_1". Anaphora resolution in DTS then proceeds as follows: (1) the representation is given an *initial context* (which is () of type $\top$), (2) the resulting representation undergoes *type checking* (cf. Löh [15]) to check whether it has a type type (i.e. the type of types (=propositions)), which in turn requires (3) that the underspecified term $@_1$ satisfies the following judgment:

(5) $$\varGamma, u : \begin{bmatrix} x{:}\mathbf{entity} \\ \begin{bmatrix} \mathbf{farmer}(x) \\ \begin{bmatrix} y{:}\mathbf{entity} \\ \begin{bmatrix} \mathbf{donkey}(y) \\ \mathbf{own}(x,y) \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \vdash @_1 : \begin{bmatrix} \top \\ \begin{bmatrix} x{:}\mathbf{entity} \\ \begin{bmatrix} \mathbf{farmer}(x) \\ \begin{bmatrix} y{:}\mathbf{entity} \\ \begin{bmatrix} \mathbf{donkey}(y) \\ \mathbf{own}(x,y) \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \rightarrow \mathbf{entity}$$

Given the above, we arrive at a choice point. The first option: if the hearer chooses to *bind* $@_1$, he/she has to find a proof term of the specified type, to replace $@_1$. Here $\lambda c.\pi_1\pi_2\pi_2\pi_2(c)$ is a candidate for such a term, which corresponds to the intended donkey. Alternatively, the hearer may choose not to execute proof search and instead *accommodate* $@_1$, in which case he/she just assumes that there is such a term $@_1$ and uses it in the subsequent inferences. In either case, the semantic representation (4b) does not need drastic reconstruction, unlike van der Sandt's [22] DRT-based approach.

In intersentential composition cases, two sentential representations are merged into one by the *dynamic conjunction* operation defined below:

(6)
$$M; N \stackrel{def}{\equiv} \lambda c. \begin{bmatrix} u{:}Mc \\ N(c,u) \end{bmatrix}$$

4 Representations of CIs in DTS

Our proposal is that a given bit of CI content A (again of type type) can be properly represented in terms of DTS in the following way:

4.1 The CI Operator

Definition 1 (The CI operator). *Let A be a type and $@_i$ be an underspecified term with an index i:*

$$\mathsf{CI}(@_i : A) \stackrel{def}{\equiv} \mathbf{eq}_A(@_i, @_i)$$

Let us call CI the *CI operator*, and a type of the form $\mathsf{CI}(@_i : A)$ a *CI type*. The CI operator is used with an underspecified term $@_i$ and a type A as its arguments (as will be demonstrated in the next section) to form a CI type. The CI type is defined in a rather technical way, but the content is simple: $\mathbf{eq}_A(M, N)$, with M, N any terms, is a type for equations between M and N in DTS, namely, it is the type of proofs of the proposition that M equals N ($@_i = @_i$, informally), both of which are of type A.

Thus, $\mathsf{CI}(@_i : A)$ is always true by the reflexivity law, under any context. In terms of DTS, $\mathsf{CI}(@_i : A)$ inhabits a canonical proof refl_A (i.e. $\vdash \mathsf{refl}_A : \mathbf{eq}_A(@_i, @_i)$).

This means that the CI operator $\mathsf{CI}(@_i : A)$ does not contribute anything to at-issue content, since we know that it is always inhabited by the term refl_A. However, the type checking of a semantic representation which contains $\mathsf{CI}(@_i : A)$ requires that the $\mathbf{eq}_A(@_i, @_i)$ has a type type, which in turn requires that the underspecified term $@_i$ has the type A. Therefore, the proposition A must have a proof term $@_i$ of type A (i.e. A must be true), which projects, regardless of the configuration in which it is embedded.

Moreover, unlike the cases of anaphora and presupposition, an underspecified term for a CI does not take any local context as its argument. This explains why CIs do not respect their left contexts.

Let us examine how our analysis how the CI operators are used to represent CIs and how they predict the set of benchmarks (1a)–(1c) for CIs.

4.2 Independence from At-Issue Content

The property (1a) is supported by the fact that both the sentences (7a) and (7b) entail the CI content *Lance Armstrong is an Arkansan*. Thus the CI content is not affected by, or projects through, logical operators such as negation that take scope over it.

(7) a. Lance Armstrong, an Arkansan, has won the 2003 Tour de France!
  b. It is not the case that Lance Armstrong, an Arkansan, has won the 2003 Tour de France!

The proposition *Lance Armstrong is an Arkansan* is represented in DTS as a type (=proposition) $\textbf{arkansan}(\mathit{lance})$. If it is embedded within the CI type as in $\mathsf{CI}(@_1 : \textbf{arkansan}(\mathit{lance}))$, this proposition is a CI content, and $@_1$ is its proof term. This embedding for an indefinite appositive construction is done by applying the following Indefinite Appositive Rule.

Definition 2 (Indefinite Appositive Rule).

$$(IA_i)\ \frac{S\backslash NP \\ : M}{S/(S\backslash NP)\backslash NP : \lambda x.\lambda p.\lambda c.\begin{bmatrix} pxc \\ \mathsf{CI}(@_i : Mxc) \end{bmatrix}}$$

This rule applies to an indefinite predicative noun phrase.[2] For example, the sentence (7a) is derived as follows,

(8)

$$\dfrac{\dfrac{\dfrac{\text{Lance}}{NP : \mathit{lance}} \quad (IA_1)\dfrac{\dfrac{\dfrac{\text{an}}{S\backslash NP/N : \mathit{id}} \quad \dfrac{\text{Arkansan}}{N : \lambda x.\lambda c.\textbf{arkansan}(x)}}{S\backslash NP : \lambda x.\lambda c.\textbf{arkansan}(x)}>}{S/(S\backslash NP)\backslash NP : \lambda x.\lambda p.\lambda c.\begin{bmatrix} pxc \\ \mathsf{CI}(@_1 : \textbf{arkansan}(x)) \end{bmatrix}}}{S/(S\backslash NP) : \lambda p.\lambda c.\begin{bmatrix} p(\mathit{lance})c \\ \mathsf{CI}(@_1 : \textbf{arkansan}(\mathit{lance})) \end{bmatrix}}< \quad \dfrac{\text{has won the 2003 Tour de France}}{S\backslash NP : \lambda x.\lambda c.\textbf{won}(x)}}{S : \lambda c.\begin{bmatrix} \textbf{won}(\mathit{lance}) \\ \mathsf{CI}(@_1 : \textbf{arkansan}(\mathit{lance})) \end{bmatrix}}>$$

The resulting SR entails that Lance is an Arkansan, because it contains the CI type $\mathsf{CI}(@_1 : \textbf{arkansan}(\mathit{lance}))$ and type checking of this SR requires $\vdash @_1 : \textbf{arkansan}(\mathit{lance})$, namely, the underspecified term $@_1$ is

[2] We should specify some features of S both on the predicate side and the rule side, in order to prevent this rule to apply to other kinds of phrases of category $S\backslash NP$, such as verb phrases, which is a routine task we will not perform here.

of type **arkansan**(*lance*). In other words, the proposition that Lance is an Arkansan is inhabited. In contrast, a derivation of (7b) is shown in (9).

(9)

$$\dfrac{\dfrac{\text{It is not the case that}}{S/S : \lambda p.\lambda c.\neg pc} \qquad \dfrac{\text{Lance, an Arkansan, has won the 2007 Tour de France}}{S : \lambda c.\begin{bmatrix}\mathbf{won}(lance)\\ \mathsf{CI}(@_1 : \mathbf{arkansan}(lance))\end{bmatrix}}}{S : \lambda c.\neg\begin{bmatrix}\mathbf{won}(lance)\\ \mathsf{CI}(@_1 : \mathbf{arkansan}(lance))\end{bmatrix}}>$$

Here again, the resulting SR contains the CI type $\mathsf{CI}(@_1 : \mathbf{arkansan}(lance))$. Since type checking of this SR is not affected by the existence of the negation operator $\neg$ that encloses it, it also requires that the proposition that Lance is an Arkansan is inhabited. This way, the CI content is predicted to be independent from at-issue content, as expected.

4.3 Presupposition Filters Do Not Filter CIs

The contrast between (10a) and (10b) exemplifies (1b): in the sentence (10a), where the definite description *the cyclist* induces a presupposition that *Lance is a cyclist*, the presupposition is filtered by the antecedent of the conditional that entails the presupposition, so the whole sentence does not have any presupposition. On the other hand, in the sentence (10b), where the indefinite appositive *a cyclist* induces the CI that *Lance is a cyclist*, the CI is not filtered by the same antecedent thus projects over it, and moreover the whole sentence is infelicitous for Gricean reasons. There are various ways in which this infelicity could be viewed, but to us it is a violation of Quantity or Manner, in that the conditional clause is uninformative, as it is pre-satisfied by the appositive content.

(10) a. If Lance is a cyclist, then the Boston Marathon was won by the cyclist.

b. If Lance is a cyclist, then the Boston Marathon was won by Lance, a cyclist.

Let us explain how this contrast is predicted in DTS. First, the derivation of (10a) is as (11).

(11)

$$\dfrac{\text{was won by}}{S\backslash NP/NP : \lambda y.\lambda x.\lambda c.\mathbf{win}(y,x)} \qquad \dfrac{\text{the cyclist}}{S\backslash NP\backslash(S\backslash NP/NP) : \lambda p.\lambda x.\lambda c.p\left(\pi_1\left(@_1 c : \begin{bmatrix}y{:}\mathbf{entity}\\ \mathbf{cyclist}(y)\end{bmatrix}\right)\right)xc}$$

$$\dfrac{\text{was won by} \quad \text{the cyclist}}{S\backslash NP : \lambda x.\lambda c.\mathbf{win}\left(\pi_1\begin{bmatrix}y{:}\mathbf{entity}\\ @_1 c : \mathbf{cyclist}(y)\end{bmatrix}, x\right)}<$$

$$\dfrac{\dfrac{\text{the BM}}{NP : bm} \quad S\backslash NP}{S : \lambda c.\mathbf{win}\left(\pi_1\left(@_1 c : \begin{bmatrix}y{:}\mathbf{entity}\\ \mathbf{cyclist}(y)\end{bmatrix}, BM\right)\right)}<$$

$$\dfrac{\dfrac{\text{If Lance is a cyclist}}{S/S : \lambda p.\lambda c.\,(u{:}\mathbf{cyclist}(lance)) \rightarrow p(c,u)} \quad S}{\lambda c.\,(u{:}\mathbf{cyclist}(lance)) \rightarrow \mathbf{win}\left(\pi_1\left(@_1(c,u) : \begin{bmatrix}y{:}\mathbf{entity}\\ \mathbf{cyclist}(y)\end{bmatrix}, BM\right)\right)}>$$

Then the type checking rules apply to the resulting SR under the initial context (), which require that the underspecified term $@_1$ satisfies the following judgment.

(12) $$\Gamma, u : \mathbf{cyclist}(lance) \vdash @_1 : \begin{bmatrix} \top \\ \mathbf{cyclist}(lance) \end{bmatrix} \to \begin{bmatrix} y{:}\mathbf{entity} \\ \mathbf{cyclist}(y) \end{bmatrix}$$

In other words, the type checking launches a proof search, which tries to find a term of type:

(13) $$\begin{bmatrix} \top \\ \mathbf{cyclist}(lance) \end{bmatrix} \to \begin{bmatrix} y{:}\mathbf{entity} \\ \mathbf{cyclist}(y) \end{bmatrix}$$

under a global context $\Gamma, u : \mathbf{cyclist}(lance)$. We assume that the hearer knows that Lance exists, i.e. we assume that the global context Γ includes the entry $lance : \mathbf{entity}$.

At a first glance, one may think that there are at least two different resolutions (14) and (15), and so that there are two different terms that satisfy (12):

(14) $$\dfrac{\dfrac{lance : \mathbf{entity} \qquad \dfrac{u : \mathbf{cyclist}(lance) \qquad \dfrac{}{x : \begin{bmatrix} \top \\ \mathbf{cyclist}(lance) \end{bmatrix}}{\scriptstyle (1)}}{u : \mathbf{cyclist}(lance)}{\scriptstyle (w)}}{(lance, u) : \begin{bmatrix} y{:}\mathbf{entity} \\ \mathbf{cyclist}(y) \end{bmatrix}}{\scriptstyle (\Sigma I)}}{\lambda x.(lance, u) : \begin{bmatrix} \top \\ \mathbf{cyclist}(lance) \end{bmatrix} \to \begin{bmatrix} y{:}\mathbf{entity} \\ \mathbf{cyclist}(y) \end{bmatrix}}{\scriptstyle (\Pi I),\,(1)}$$

(15) $$\dfrac{\dfrac{lance : \mathbf{entity} \qquad \dfrac{\dfrac{}{x : \begin{bmatrix} \top \\ \mathbf{cyclist}(lance) \end{bmatrix}}{\scriptstyle (1)}}{\pi_2 x : \mathbf{cyclist}(lance)}{\scriptstyle (\Sigma E)}}{(lance, \pi_2 x) : \begin{bmatrix} y{:}\mathbf{entity} \\ \mathbf{cyclist}(y) \end{bmatrix}}{\scriptstyle (\Sigma I)}}{\lambda x.(lance, \pi_2 x) : \begin{bmatrix} \top \\ \mathbf{cyclist}(lance) \end{bmatrix} \to \begin{bmatrix} y{:}\mathbf{entity} \\ \mathbf{cyclist}(y) \end{bmatrix}}{\scriptstyle (\Pi I),\,(1)}$$

However, only (15) is licenced, because the underspecified term $@_1$ must not contain u as a free variable. The reason, which is a bit technical but empirically important, is that we implicitly assumed it in the derivation (11): more precisely, in the functional application between "If Lance is a cyclist" and "the BM was won by the cyclist", the following β-reduction took place.

(16) $$(\lambda p.\lambda c.\,(u{:}\mathbf{cyclist}(\mathit{lance})) \to p(c,u)) \left(\lambda c.\mathbf{win}\left(\pi_1\left(@_1 c : \begin{bmatrix} y{:}\mathbf{entity} \\ \mathbf{cyclist}(y) \end{bmatrix}, \mathit{BM}\right)\right)\right)$$

$$\longrightarrow_\beta \lambda c.\,(u{:}\mathbf{cyclist}(\mathit{lance})) \to \mathbf{win}\left(\pi_1\left(@_1(c,u) : \begin{bmatrix} y{:}\mathbf{entity} \\ \mathbf{cyclist}(y) \end{bmatrix}, \mathit{BM}\right)\right)$$

If the variable u occurs free in the underspecified term $@_1$, the variable u in the $\lambda p.\lambda c.\,(u{:}\mathbf{cyclist}(\mathit{lance})) \to p(c,u)$ part should have been renamed. This means that $@_1$ does not contain u as a free variable, thus $\lambda x.(\mathit{lance}, u)$ is not a candidate to replace $@_1$.

Therefore, if one wants to bind $@_1$, then $@_1 = \lambda x.(\mathit{lance}, \pi_2 x)$. This shows that the presupposition triggered by "the cyclist" is bound by the local context, the information given by the antecedent of the conditional.

In the case of CI, the situation is different. The derivation of (10b) is as (17).

(17)

was won by: $S\backslash NP/NP : \lambda y.\lambda x.\lambda c.\mathbf{win}(y,x)$

Lance, a cyclist: $S/(S\backslash NP) : \lambda p.\lambda x.\lambda c.\begin{bmatrix} p(\mathit{lance})xc \\ \mathsf{CI}(@_2 : \mathbf{cyclist}(\mathit{lance})) \end{bmatrix}$

$$\frac{\text{was won by} \quad \text{Lance, a cyclist}}{S\backslash NP : \lambda x.\lambda c.\begin{bmatrix} \mathbf{win}(\mathit{lance},x) \\ \mathsf{CI}(@_2 : \mathbf{cyclist}(\mathit{lance})) \end{bmatrix}} <$$

the BM: $NP : \mathit{bm}$

$$\frac{\text{the BM} \quad S\backslash NP}{S : \lambda c.\begin{bmatrix} \mathbf{win}(\mathit{lance},\mathit{BM}) \\ \mathsf{CI}(@_2 : \mathbf{cyclist}(\mathit{lance})) \end{bmatrix}} <$$

If Lance is a cyclist: $S/S : \lambda p.\lambda c.\,(u{:}\mathbf{cyclist}(\mathit{lance})) \to p(c,u)$

$$\frac{\text{If Lance is a cyclist} \quad S}{\lambda c.\,(u{:}\mathbf{cyclist}(\mathit{lance})) \to \begin{bmatrix} \mathbf{win}(\mathit{lance},\mathit{BM}) \\ \mathsf{CI}(@_2 : \mathbf{cyclist}(\mathit{lance})) \end{bmatrix}} >$$

Type checking rules apply to the resulting SR under the initial context (), which requires that the underspecified term $@_2$ satisfies the following judgment.

(18) $$\Gamma, u : \mathbf{cyclist}(\mathit{lance}) \vdash @_2 : \mathbf{cyclist}(\mathit{lance})$$

It seems that the variable u is an immediate candidate that can replace $@_2$, but this is not licenced, for the same reason as the case of definite descriptions: The underspecified term $@_2$ should not contain the free occurrence of u, since it is implicitly assumed in the β-reduction that took place at the bottom of (17).

Thus, there is no binding option for the CI in (10b) unless the global context Γ provides some knowledge that allows its inference. Otherwise, the hearer has to update Γ accordingly, i.e. accommodate it. The simplest way is to use the following updated global context Γ' (x is some variable chosen so that $x \notin \Gamma$).

(19) $$\Gamma' \overset{def}{\equiv} \Gamma,\ x : \mathbf{cyclist}(\mathit{lance})$$

The difference between the two cases (10a) and (10b) is that in the formulation of CIs, the underspecified term does not take a local context as its argument, and so cannot refer to it, while in the formulation of presuppositions, the underspecified term is given a local context as its argument, and so is able to bind it by means of information deduced from the local context. This way, DTS predicts that antecedents of conditionals, which are of course presupposition filters, do not filter CI contents, thus deriving one of the empirical differences between these types of content.

It is also predicted in DTS that the sentence (10b) is pragmatically infelicitous. In order to *accept* (10b) as a felicitous sentence, one has to add the entry $x : \mathbf{cyclist}(\mathit{lance})$ to his/her global context in most cases. It is then inappropriate to assume that Lance is a cyclist, as in (10b), is redundant, since it is immediately derivable from the global context. This is one way to implement the idea of the infelicity of (10b) as a Gricean violation of the kind mentioned above.

4.4 CIs Do Not Modify CIs

Typical cases that exemplify (1c) are examples like (20), where the speaker-oriented adverb *surprisingly* does not modify the expressive content induced by *the bastard*, i.e. the bastardhood of Jerry is not surprising for the speaker.

(20) Surprisingly, Jerry, the bastard, showed up with no money.

The derivation of (20) in DTS is as follows, assuming that the definite appositive is analyzed in the same way as the indefinite appositives, and the speaker-oriented adverb *surprisingly* takes a proof of the sentence it modifies.

(21)

$$
\dfrac{\dfrac{\text{surprisingly}}{S/S : \lambda p.\lambda c.\begin{bmatrix} u{:}pc \\ \mathsf{CI}(@_1 : \mathbf{surprising}(u)) \end{bmatrix}} \quad \dfrac{\dfrac{\text{Jerry, the bastard}}{S\backslash(S/NP) : \lambda p.\lambda c.\begin{bmatrix} p(\mathit{jerry})c \\ \mathsf{CI}(@_2 : \mathbf{bastard}(\mathit{jerry})) \end{bmatrix}} \quad \dfrac{\text{showed up with no money}}{S\backslash NP : \lambda x.\lambda c.\mathbf{showedUpNoMoney}(x)}}{S : \lambda c.\begin{bmatrix} \mathbf{showedUpNoMoney}(\mathit{jerry}) \\ \mathsf{CI}(@_2 : \mathbf{bastard}(\mathit{jerry})) \end{bmatrix}}>}{S : \lambda c.\begin{bmatrix} u{:}\begin{bmatrix} \mathbf{showedUpNoMoney}(\mathit{jerry}) \\ \mathsf{CI}(@_2 : \mathbf{bastard}(\mathit{jerry})) \end{bmatrix} \\ \mathsf{CI}(@_1 : \mathbf{surprising}(u)) \end{bmatrix}}>
$$

Type checking of the resulting semantic representation in (20) under the initial context (), requires that the two underspecified terms are of the following types:

(22) a. $\Gamma,\ u : \begin{bmatrix} \mathbf{showedUpNoMoney}(\mathit{jerry}) \\ \mathsf{CI}(@_2 : \mathbf{bastard}(\mathit{jerry})) \end{bmatrix} \vdash @_1 : \mathbf{surprising}(u)$

b. $\Gamma \vdash @_2 : \mathbf{bastard}(\mathit{jerry})$

The judgment (22b) immediately requires the update of Γ, if it does not entail the bastardhood of Jerry. The case of judgment (22a), on the other hand, is more complex, since the **surprising** predicate is about a variable u, which is a proof term of type:

(23)
$$
\begin{bmatrix} \mathbf{showedUpNoMoney}(\mathit{jerry}) \\ \mathsf{CI}(@_2 : \mathbf{bastard}(\mathit{jerry})) \end{bmatrix}
$$

However, since the type $\mathsf{CI}(@_2 : \mathbf{bastard}(\mathit{jerry}))$ inhabits only one term $\mathsf{refl}_{\mathbf{bastard}(\mathit{jerry})}$, the value of u only varies over the terms of type:

(24) $$\mathbf{showedUpNoMoney}(\mathit{jerry})$$

and so states that Jerry showed up with no money. Thus, whether it is surprising only depends on how Jerry showed up with no money, and not on how the equality between two identical $@_2$ results in identity.

Thus DTS predicts that there is no interactions between different bits of CI content.[3]

5 Solution to the Puzzles

Let us now proceed to show how our analysis solves the puzzles regarding the interaction between CI contents and anaphora/presuppositions.

5.1 A CI can Serve as an Antecedent for the Subsequent Anaphora/presuppositions

The semantic representation for (2) is derived as (25). We assume a distinct lexical entry for "who" for NRRCs, which contains the CI operator for specifying their CI content.

The resulting discourse representation contains three underspecified terms: $@_1$ for the CI content, $@_2$ for the factive presupposition of "knows", and $@_3$ for the pronoun "he".

Type checking requires the term $@_1$ to be of type $\mathbf{KC}(\mathit{john})$, which will be accommodated as new information to the hearer. The term $@_3$ can be independently resolved if it is intended to be coreferential to "John", namely, as $@_3 = \lambda c.\mathit{john}$. Then the term $@_2$, which is required to have type $\mathbf{KC}(\mathit{john})$, can be bound just by being identified with $@_1$. In this way, what is introduced as a CI can bind the subsequent presuppositions, although it does not participate in the at-issue content.

[3] There is a possible problem with attributing the property (1c) to CIs. Gutzmann [9,10] argues that sentences such as (1) is a possible counter-example for (1c) in the sense that *fucking* in (1) serves to intensify the degree to which Jerry has the property of being an asshole, which is CI content that is induced by *asshole*; thus, the adjective works to strengthen not-at-issue content in cases of this kind.

(1) Jerry is a fucking asshole.

The current version of DTS, however, predicts that the target of the modification performed by *fucking* does not include the CI content of *asshole*, just as in the case of (20). We believe this issue relates to the sort of variance in what counts as "at-issue" discussed by Horn [12,13], and, as such, exhibits a level of complexity that requires a more detailed look at the pragmatics of these constructions (cf. Amaral et al. [1]). This difficult project is beyond the scope of the present paper.

(25)

$$\text{John} \vdash NP : \mathit{john}$$
$$\text{, who}_1 \vdash \boldsymbol{T}\backslash(\boldsymbol{T}/NP)\backslash NP/(S\backslash NP) : \lambda r.\lambda z.\lambda p.\lambda \boldsymbol{x}.\lambda c.\begin{bmatrix} pz\boldsymbol{x}c \\ \mathsf{CI}(@_1 : rzc) \end{bmatrix}$$
$$\text{killed a coworker} \vdash S\backslash NP : \lambda x.\lambda c.\mathbf{KC}(x)$$
$$> \quad \boldsymbol{T}\backslash(\boldsymbol{T}/NP)\backslash NP : \lambda z.\lambda p.\lambda \boldsymbol{x}.\lambda c.\begin{bmatrix} pz\boldsymbol{x}c \\ \mathsf{CI}(@_1 : \mathbf{KC}(z)) \end{bmatrix}$$
$$< \quad \boldsymbol{T}\backslash(\boldsymbol{T}/NP) : \lambda p.\lambda x.\lambda c.\begin{bmatrix} p\,\mathit{john}\,xc \\ \mathsf{CI}(@_1 : \mathbf{KC}(\mathit{john})) \end{bmatrix}$$
$$\text{counselled} \vdash S\backslash NP/NP : \lambda y.\lambda x.\lambda c.\mathbf{counsel}(x, y)$$
$$< \quad S\backslash NP : \lambda x.\lambda c.\begin{bmatrix} \mathbf{counsel}(x, \mathit{john}) \\ \mathsf{CI}(@_1 : \mathbf{KC}(\mathit{john})) \end{bmatrix}$$
$$\text{Mary} \vdash NP : \mathit{mary}$$
$$< \quad S : \lambda c.\begin{bmatrix} \mathbf{counsel}(\mathit{mary}, \mathit{john}) \\ \mathsf{CI}(@_1 : \mathbf{KC}(\mathit{john})) \end{bmatrix};$$

$$\text{Bill} \vdash NP : \mathit{bill}$$
$$\text{knows}_2 \text{ that} \vdash S\backslash NP/S : \lambda p.\lambda x.\lambda c.\mathbf{know}(x, @_2 : pc)$$
$$\text{he}_3 \text{ killed a coworker} \vdash S : \lambda c.\mathbf{KC}(@_3 c)$$
$$> \quad S\backslash NP : \lambda x.\lambda c.\mathbf{know}(x, @_2 : \mathbf{KC}(@_3 c))$$
$$> \quad S : \lambda c.\mathbf{know}(\mathit{bill}, @_2 : \mathbf{KC}(@_3 c))$$

$$\xrightarrow{\text{Dynamic conjunction}} \quad \lambda c.\begin{bmatrix} u : \begin{bmatrix} \mathbf{counsel}(\mathit{mary}, \mathit{john}) \\ \mathsf{CI}(@_1 : \mathbf{KC}(\mathit{john})) \end{bmatrix} \\ \mathbf{know}(\mathit{bill}, @_2 : \mathbf{KC}(@_3(u, c))) \end{bmatrix}$$

5.2 Anaphora/Presuppositions Inside CIs Receive Their Left Contexts

The semantic representation for (3) is derived as follows.

(26)

$$\text{John} \vdash NP : \mathit{john} \qquad \text{killed a coworker} \vdash S\backslash NP : \lambda x.\lambda c.\mathbf{KC}(x)$$
$$> \quad S : \lambda c.\mathbf{KC}(\mathit{john});$$

$$\text{knows}_2 \text{ that} \vdash S\backslash NP/S : \lambda p.\lambda x.\lambda c.\mathbf{know}(x, @_2 c : pc)$$
$$\text{John killed a coworker} \vdash S : \lambda c.\mathbf{KC}(\mathit{john})$$
$$> \quad S\backslash NP : \lambda x.\lambda c.\mathbf{know}(x, @_2 c : \mathbf{KC}(\mathit{john}))$$
$$\text{, who}_1 \vdash \boldsymbol{T}/(\boldsymbol{T}\backslash NP)\backslash NP/(S\backslash NP) : \lambda r.\lambda z.\lambda p.\lambda \boldsymbol{x}.\lambda c.\begin{bmatrix} pz\boldsymbol{x}c \\ \mathsf{CI}(@_1 : rzc) \end{bmatrix}$$
$$> \quad \boldsymbol{T}/(\boldsymbol{T}\backslash NP)\backslash NP : \lambda z.\lambda p.\lambda c.\begin{bmatrix} pzc \\ \mathsf{CI}(@_1 : \mathbf{know}(z, @_2 c : \mathbf{KC}(\mathit{john}))) \end{bmatrix}$$
$$\text{Mary} \vdash NP : \mathit{mary}$$
$$< \quad \boldsymbol{T}/(\boldsymbol{T}\backslash NP) : \lambda p.\lambda c.\begin{bmatrix} p\,\mathit{mary}\,c \\ \mathsf{CI}(@_1 : \mathbf{know}(\mathit{mary}, @_2 c : \mathbf{KC}(\mathit{john}))) \end{bmatrix}$$
$$\text{counselled him}_3 \vdash S\backslash NP : \lambda x.\lambda c.\mathbf{counsel}(x, @_3 c)$$
$$> \quad S : \lambda c.\begin{bmatrix} \mathbf{counsel}(\mathit{mary}, @_3 c) \\ \mathsf{CI}(@_1 : \mathbf{know}(\mathit{mary}, @_2 c : \mathbf{KC}(\mathit{john}))) \end{bmatrix}$$

$$\xrightarrow{\text{Dynamic conjunction}} \quad \lambda c.\begin{bmatrix} u{:}\mathbf{KC}(\mathit{john}) \\ \begin{bmatrix} \mathsf{CI}(@_1 : \mathbf{know}(\mathit{mary}, @_2(c, u) : \mathbf{KC}(\mathit{john}))) \\ \mathbf{counsel}(\mathit{mary}, @_3(c, u)) \end{bmatrix} \end{bmatrix}$$

The resulting discourse representation contains three underspecified terms: $@_2$ for the factive presupposition triggered by "know" which states that John killed a coworker, $@_1$ for the NRRC that Mary knows it, and $@_3$ for the pronoun "him".

The factive presupposition $@_2$, which is embedded within the CI for NRRC, still receives its left context (c, u) that is a pair of the left context for this mini discourse and the proof of the first sentence. Obviously, the most salient resolution of this underspecification is $@_2 = \lambda c.\pi_2 c$, which returns the proof of the first sentence. What enables this solution is the flexibility of DTS in which the lexical entry of "who" can pass the left context it receives to the relative clause, while the CI content $@_1$ that it introduces does not receive it.

6 Conclusion

In this paper, we have given an analysis of conventional implicature in the framework of Dependent Type Semantics. In this framework, phenomena such as anaphora resolution and presupposition are viewed in terms of proof search; we have shown that this viewpoint, together with suitable constraints on conventional implicature, naturally derive certain observed behavior of conventional implicature with respect to semantic operators and interaction between at-issue and conventionally implicated content. We think the resulting picture is attractive, not least in that it is fully integrated with compositional, subsentential aspects of meaning derivation.

As we observed above, there are many other competing approaches to the derivation of conventional implicatures, and other analyses of their interaction with anaphora and presupposition. Analyses of the first type are generally based on type theory of the kind more standard in linguistic theory, as exemplified by [11,17,19]; the second sort of work tends to be set in dynamic semantics, in line with the majority of formal work on anaphora and presupposition in recent years. This paper removes the explicit focus on dynamics and works with a different view of type theory; as such, it can be placed directly within the recent movement to use continuations and other non-dynamic techniques to simultaneously model intersentential phenomena and to provide a compositional analysis of problems traditional views of composition have found difficult (e.g. [2–4,8]. We leave a full comparison of our theory here with existing views for future work.

This work exhibits many directions for future expansion. We would like to close with one that we believe is of general interest and that shows the power of the current approach. Roberts et al. [23] suggest that the projection behavior of not-at-issue content – i.e. that content which includes presupposition, conventional implicature, and possibly other types which do not play a direct role in the determination of truth conditions – depends on the relation of that content to the current Question Under Discussion, or QUD [21]. We are somewhat agnostic about the precise way in which this claim could or should be formalized, especially given the currently somewhat mysterious ontological status of QUDs; but we are highly sympathetic to the idea that projection behavior should be relativized in some manner to the discourse context, and possibly to the goals and desires of the participants (e.g. as realized by a QUD). But this view is clearly close to what we have set forward here. Plainly the discourse context makes various sorts of content available; if that content contains such things as goals and QUDs, then they ought to play a role in proof search as well, and so we might expect that different computations could be carried out in different contexts, yielding different projection behavior for not-at-issue content. The exact form by which this idea should be spelled out depends on a number of factors: the analysis of questions, probably the proper analysis of denial and other relational speech acts, the form of QUDs and the manner in which they are derived, and of course empirical facts about the projection behavior of not-at-issue content and its relation to contextual elements. We believe that exploring these issues is an exciting next step for the present project.

References

1. Amaral, P., Roberts, C., Smith, E.: Review of 'the logic of conventional implicatures' by Christopher Potts. Linguist. Philos. **30**, 707–749 (2008)
2. Asher, N., Pogodalla, S.: SDRT and continuation semantics. In: Onoda, T., Bekki, D., McCready, E. (eds.) JSAI-isAI 2010. LNCS, vol. 6797, pp. 3–15. Springer, Heidelberg (2011)
3. Barker, C., Bernardi, R., Shan, C.: Principles of interdemensional meaning interaction. In: Li, N., Lutz, D. (eds.) Semantics and Linguistic Theory (SALT) 20, pp. 109–127. eLanguage (2011)
4. Barker, C., Shan, C.C.: Donkey anaphora is in-scope binding. Semantics and Pragmatics **1**(1), 1–46 (2008)
5. Bekki, D.: Representing anaphora with dependent types. In: Asher, N., Soloviev, S. (eds.) LACL 2014. LNCS, vol. 8535, pp. 14–29. Springer, Heidelberg (2014)
6. Coquand, T., Huet, G.: The calculus of constructions. Inf. Comput. **76**(2–3), 95–120 (1988)
7. Grice, H.P.: Logic and conversation. In: Cole, P., Morgan, J.L. (eds.) Syntax and Semantics 3: Speech Acts, pp. 41–58. Academic Press, London (1975)
8. de Groote, P.: Towards a montagovian account of dynamics. In: Gibson, M., Howell, J. (eds.) 16th Semantics and Linguistic Theory Conference (SALT16), pp. 148–155. CLC Publications, University of Tokyo (2006)
9. Gutzmann, D.: Expressive modifiers & mixed expressives. In: Bonami, O., Cabredo Hofherr, P. (eds.) Empirical Issues in Syntax and Semantics 8, pp. 143–165 (2011)
10. Gutzmann, D.: Use-Conditional Meaning: studies in multidimensional semantics. Ph.D. thesis, Universität Frankfurt (2012)
11. Heim, I., Kratzer, A.: Semantics in Generative Grammar. Blackwell, Oxford (1998). No. 13 in Blackwell Textbooks in Linguistics
12. Hom, C.: The semantics of racial epithets. J. Philos. **105**, 416–440 (2008)
13. Hom, C.: A puzzle about pejoratives. Philos. Stud. **159**(3), 383–405 (2010)
14. Krahmer, E., Piwek, P.: Presupposition projection as proof construction. In: Bunt, H., Muskens, R. (eds.) Computing Meanings: Current Issues in Computational Semantics. Kluwer Academic Publishers, Dordrecht (1999). Studies in Linguistics Philosophy Series

15. Löh, A., McBride, C., Swierstra, W.: A tutorial implementation of a dependently typed lambda calculus. Fundamenta Informaticae - Dependently Typed Program. **102**(2), 177–207 (2010)
16. Martin-Löf, P.: Intuitionistic type theory. In: sambin, G. (ed.) vol. 17. Bibliopolis, Naples (1984)
17. Montague, R.: The proper treatment of quantification in ordinary english. In: Hintikka, J., Moravcsic, J., Suppes, P. (eds.) Approaches to Natural Language, pp. 221–242. Reidel, Dordrecht (1973)
18. Nouwen, R.: On appositives and dynamic binding. Res. Lang. Comput. **5**, 87–102 (2007)
19. Potts, C.: The Logic of Conventional Implicatures. Oxford University Press, New York (2005)
20. Ranta, A.: Type-Theoretical Grammar. Oxford University Press, New York (1994)
21. Roberts, C.: Information structure: towards an integrated formal theory of pragmatics. In: OSUWPL, Papers in Semantics, vol. 49. The Ohio State University Department of Linguistics (1996)
22. van der Sandt, R.: Presupposition projection as anaphora resolution. J. Seman. **9**, 333–377 (1992)
23. Simons, M., Tonhauser, J., Beaver, D., Roberts, C.: What projects and why. In: Proceedings of SALT 20, pp. 309–327. CLC Publications (2011)
24. Sundholm, G.: Proof theory and meaning. In: Gabbay, D., Guenthner, F. (eds.) Handbook of Philosophical Logic, vol. III, pp. 471–506. Kluwer, Reidel, Dordrecht (1986)
25. Wang, L., Reese, B., McCready, E.: The projection problem of nominal appositives. Snippets **11**, 13–14 (2005)