

Context-Passing and Underspecification in Dependent Type Semantics

Daisuke Bekki and Koji Mineshima

Abstract Dependent type semantics (DTS) is a framework of discourse semantics based on dependent type theory, following the line of Sundholm (Handbook of Philosophical Logic, 1986) and Ranta (Type-Theoretical Grammar, 1994). DTS attains compositionality as required to serve as a semantic component of modern formal grammars including variations of categorial grammars, which is achieved by adopting mechanisms for local contexts, context-passing, and underspecified terms. In DTS, the calculation of presupposition projection reduces to type checking, and the calculation of anaphora resolution and presupposition binding both reduce to proof search in dependent type theory, inheriting the paradigm of anaphora resolution as proof construction.

1 Introduction

1.1 *Natural Language Semantics via Dependent Type Theory*

In the late 1980s, against the backdrop of the rapid development of model-theoretic discourse semantics such as Discourse Representation Theory (DRT) (Kamp 1981), File Change Semantics (FCS) (Heim 1982), and Dynamic Predicate Logic (DPL)

Our sincere thanks to Kenichi Asai, Nicholas Asher, Kentaro Inui, Yusuke Kubota, Sadao Kurohashi, Robert Levine, Zhaohui Luo, Ribeka Tanaka and Ayumi Ueyama for many insightful comments. We also thank to Youyou Cong, Yuri Ishishita, Ayako Nakamura, Yuki Nakano, Miho Sato and Maika Utsugi for helpful discussions. This research is partially supported by JST, CREST.

D. Bekki (✉) · K. Mineshima

Ochanomizu University/CREST, Japan Science and Technology Agency (JST),
Tokyo, Japan
e-mail: bekki@is.ocha.ac.jp

K. Mineshima

e-mail: mineshima.koji@ocha.ac.jp

(Groenendijk and Stokhof 1991), Martin-Löf and Sundholm noticed that *dependent type theory* (DTT),¹ which extends simply typed lambda calculus by adding *dependent types*, may provide semantic representations of discourses involving dynamic binding, which are parallel to their syntactic structures. This idea can be elaborated as a solution to the *compositionality problem*, that is, the discrepancy between syntactic structures and semantic representations (SRs) of certain sentences: a sentence including donkey anaphora (Geach 1962) as the sentence (1); E-type anaphora (Evans 1980) as the sentences (2); and, more generally, discourse referents as discussed in Karttunen (1976).²

- (1) Every farmer who owns [a donkey]_i beats it_i.
- (2) a. [A man]_j entered.
b. He_j whistled.

1.2 Compositionality Problem of Discourse Anaphora

Let us briefly summarize the compositionality problem of discourse anaphora, which has been repeatedly discussed in the literature, starting from Geach (1962) and Evans (1980). For the donkey sentence (1), a first-order formula (3), whose truth condition is the same as that of (1), is a candidate of its SR.

$$(3) \forall x(\mathbf{farmer}(x) \rightarrow \forall y(\mathbf{donkey}(y) \wedge \mathbf{own}(x, y) \rightarrow \mathbf{beat}(x, y)))$$

The problem of (3) as the SR of (1) is that translation from the sentence (1) to (3) is not straightforward since (i) the indefinite noun phrase *a donkey* is translated into a universal quantifier in (3) instead of an existential quantifier, and (ii) the syntactic structure of (3) does not corresponds to that of (1).

The syntactic parallel of (1) is, rather, the SR (4), in which the indefinite noun phrase is translated into an existential quantification. However, (4) does not represent the truth condition of (1) correctly since the variable y in $\mathbf{beat}(x, y)$ fails to be bound by \exists .

$$(4) \forall x(\mathbf{farmer}(x) \wedge \exists y(\mathbf{donkey}(y) \wedge \mathbf{own}(x, y)) \rightarrow \mathbf{beat}(x, y))$$

Therefore, neither (3) nor (4) qualifies as the SR of (1). Similar arguments apply to the case of the E-type anaphora in (2) as well. The first-order SR (5), which represents the truth condition of (2), is a candidate of the SR of (2), but the syntactic structure

¹The representative version of dependent type theory is Martin-Löf Type Theory (MLTT) (Martin-Löf 1984), which is also known as Constructive Type Theory or Intensional Type Theory. In this article, we use the term “dependent type theory” as a term to refer to any type theory with dependent types, including MLTT, λP (Barendregt 1992), Calculus of Construction (CoC) (Coquand and Huet 1988), and Unified Type Theory (UTT) (Luo 2012b).

²The subscripts i and j signify that we focus on judgments under a specified reading in which the antecedent of *it* is *a donkey* in (1), and the antecedent of *He* is *A man* in (2).

of the SR (5) does not correspond to that of (2) either, since the mini-discourse (2) consists of two independent sentences.

(5) $\exists x(\mathbf{man}(x) \wedge \mathbf{enter}(x) \wedge \mathbf{whistle}(x))$

The sentential boundary of (2) should prefer the first-order representation (6), but the truth condition of (6) is different from that of the mini-discourse (2) since the variable x in **whistle**(x) is not bound by \exists .

(6) $\exists x(\mathbf{man}(x) \wedge \mathbf{enter}(x)) \wedge \mathbf{whistle}(x)$

We should elaborate on the difficulty of composing (5) from the SRs of (2a) and (2b), which may be decomposed into the following three questions.

Question 1: What is the SR of (2a)?

Question 2: What is the SR of (2b)? In particular, what is the SR of *He*?

Question 3: How is the SR (5) compositionally obtained from the answers for Questions 1 and 2?

Recall that, until the emergence of discourse semantics such as DRT, FCS, and DPL, it was not straightforward to give a single solution to these questions, since the three questions are entangled with each other. This is revealed by putting the following three naïve assumptions together.

Assumption 1: The SR of (2a) is $\exists x(\mathbf{man}(x) \wedge \mathbf{enter}(x))$

Assumption 2: The SR of (2b) is **whistle**(x)

Assumption 3: The SR of two assertive sentences is obtained by conjoining their SRs with \wedge .

If we maintain all three assumptions, we obtain (6). So we have to abandon at least one of these assumptions or other hidden assumptions behind this naïve analysis. For example, DRT abandons Assumptions 1 and 3, and also the direct compositionality of meaning. DPL abandons Assumption 1, and also the standard model-theoretic interpretation of first-order logic, so that (5) and (6) become equivalent.

As will be seen, dependent type theory succeeds in solving the compositionality problem of discourse by abandoning Assumption 2, and by substituting model-theoretic interpretations of SRs with *proof-theoretic* interpretations, which provides not only a key idea for solving the particular problem of anaphora, but an alternative perspective for the theory of meaning.

1.3 Partial Solutions in Dependent Type Theory

In natural language semantics based on dependent type theory, the meaning of a declarative sentence is represented by a type, which is a collection of proofs under a given context. This is a major divergence from the model-theoretic semantics dating back to Montague (1974), in which a proposition *denotes* a truth value or a set of possible worlds. In dependent type theory, a type has no denotation; instead, its meaning

Table 1 DTS-style versus standard notations for dependent types

	DTS-style notation	Standard notation
Dependent function type (Π -type)	$(x : A) \rightarrow B$	$(\Pi x : A)B$
Dependent product type (Σ -type)	$(x : A) \times B, \begin{bmatrix} x : A \\ B \end{bmatrix}$	$(\Sigma x : A)B$

is defined by the inference rules, which consist of formation rules, introduction rules, and elimination rules, as shown in Definitions 1 and 2. These rules specify how a type (as a proposition) can be formed and proved under a given context. In particular, introduction rules and elimination rules provide *verificational* and *pragmatist* accounts of a given constructor, and the former is considered as primary, according to Gentzen (1935), in the sense that the latter can be derived from the former. Thus, the meaning of a sentence in proof-theoretic semantics lies in its *verification condition*, in line with the philosophy of language that originates in Dummett (1975, 1976) and Prawitz (1980).³

Definition 1 (*Dependent function type*) For any $(s_1, s_2) \in \{(\text{type}, \text{type}), (\text{type}, \text{kind}), (\text{kind}, \text{type}), (\text{kind}, \text{kind})\}$, $s \in \{\text{type}, \text{kind}\}$,

$$\frac{\overline{x : A}^i \vdots \dots A : s_1 \quad B : s_2}{(x : A) \rightarrow B : s_2} (\Pi F), i \quad \frac{\overline{x : A}^i \vdots \dots A : s \quad M : B}{\lambda x. M : (x : A) \rightarrow B} (\Pi I), i \quad \frac{M : (x : A) \rightarrow B \quad N : A}{MN : B[N/x]} (\Pi E)$$

Definition 2 (*Dependent product type*) For any $(s_1, s_2) \in \{(\text{type}, \text{type}), (\text{type}, \text{kind}), (\text{kind}, \text{kind})\}$,

$$\frac{\overline{x : A}^i \vdots \dots A : s_1 \quad B : s_2}{\begin{bmatrix} x : A \\ B \end{bmatrix} : s_2} (\Sigma F), i \quad \frac{M : A \quad N : B[M/x]}{(M, N) : \begin{bmatrix} x : A \\ B \end{bmatrix}} (\Sigma I) \quad \frac{M : \begin{bmatrix} x : A \\ B \end{bmatrix}}{\pi_1 M : A} (\Sigma E) \quad \frac{M : \begin{bmatrix} x : A \\ B \end{bmatrix}}{\pi_2 M : B[\pi_1 M/x]} (\Sigma E)$$

In dependent type theory, two kinds of dependent types are added to simply-typed lambda calculus: *dependent function type* or Π -type (notation $(x : A) \rightarrow B$) and *dependent product type* or Σ -type (notation $(x : A) \times B$) as shown in (Table 1).⁴ By

³Francez and Dyckhoff (2010) and Francez et al. (2010) also pursued a proof-theoretic semantics of natural language. The difference in their approach is that the meaning of a *word* itself is defined via its verification conditions, whereas in our approach the meaning of a word is represented by a term in dependent type theory, as a contribution to the meaning of a sentence it may participate in. Luo (2014) provides a comparison between Francez's approach and dependent-theoretic approaches, together with an interesting discussion on the proof-theoretic and model-theoretic status of natural language semantics via dependent type theory.

⁴DTS also employs a two-dimensional notation for Σ -types as shown in Definition 2, which is reminiscent of the notation for record types in Cooper (2005).

Curry–Howard correspondence between types and propositions, a type $(x : A) \rightarrow B$ corresponds to a universal quantification $(\forall x : A)B$, and also an implication $A \rightarrow B$ when $x \notin fv(B)$.⁵ A type $(x : A) \times B$ corresponds to an existential quantification $(\exists x : A)B$, and also a conjunction $A \wedge B$ when $x \notin fv(B)$.

In a standard setting of dependent type theory, more types are employed: intensional equality type, disjoint union type, enumeration types (including \top and \perp ; the latter is used to define negation) and natural number type. We assume that dependent type theory includes such types that are necessary for representing logical operators in natural language semantics, and also the basic rules such as the **type** formation rule, the conversion rule and the weakening rule given as Definition 3. For details, please refer to Nordström et al. (1990).⁶

Definition 3 (*Basic rules*)

$$\frac{}{\text{type} : \text{kind}} \text{ (typeF)} \quad \frac{M : A}{M : B} \text{ (CONV)} \quad \frac{M : A \quad N : B}{M : A} \text{ (WK)} \\ \text{where } A =_{\beta} B$$

The SR of a donkey sentence (1) in our analysis is as in (7).⁷

$$(1) \text{ Every farmer who owns [a donkey]}_i \text{ beats it}_i. \\ (7) \left(u : \left[\left[\left[\begin{array}{c} x : \text{entity} \\ \text{farmer}(x) \end{array} \right] \right] \left[\left[\left[\begin{array}{c} y : \text{entity} \\ \text{donkey}(y) \end{array} \right] \right] \left[\text{own}(x, \pi_1 v) \right] \right] \right] \right) \rightarrow \text{beat}(\pi_1 u, \pi_1 \pi_1 \pi_2 \pi_2 u)$$

The syntactic structure of the SR (7) parallels that of (1). Moreover, the universal and existential quantifications are uniformly translated to dependent function types and dependent product types, respectively. Recall that the SR (3) translates the indefinite noun phrase to \forall and fails to preserve the constituent structure of (1).

It follows from the inference rules in Definitions 1 and 2 that a proof of $(x : A) \rightarrow B$ is a (fibred) function from A to B , while a proof of $(x : A) \times B$ is a (fibred) pair of A and B . The operators π_1 and π_2 are, respectively, the first and second projections from a given pair. Thus, in the SR (7), the type

⁵We denote the set of free variables in B by $fv(B)$.

⁶DTS employs two sorts: **type** and **kind**, and its terms are stratified into three levels: terms of type A where A is a type, types of sort **type**, kinds of sort **kind**. The only axiom is (typeF) in Definition 3. The (ΠF) rule allows the four patterns (type, type), (type, kind), (kind, type) and (kind, kind) as in Definition 1, and the (ΣF) rule allows the three patterns (type, type), (type, kind) and (kind, kind) as in Definition 2. Thus, in this article, DTS employs dependent type theory in which **type** is an impredicative universe with respect to Π . This setting is stronger than the predicative dependent type theory that Bekki (2014) is founded on, but not too strong to construct a proof of Girard’s paradox (Girard 1972; Coquand 1986; Hook and Howe 1986). We are grateful to Zhaohui Luo (personal communication) for discussions and comments on this issue.

⁷Following the notation in logic, we write **farmer**(x) for (**farmer** x) and **own**(x, y) for (**own** y) x , and so on. More generally, for an n -place predicate **f**, we often write **f**(x_1, \dots, x_n) for $(\dots (\mathbf{f} x_n) \dots x_1)$.

$$\left[\left[\left[\begin{array}{l} x : \text{entity} \\ \text{farmer}(x) \end{array} \right] \left[\begin{array}{l} v : \left[\begin{array}{l} y : \text{entity} \\ \text{donkey}(y) \end{array} \right] \\ \text{own}(x, \pi_1 v) \end{array} \right] \right] \right]$$

represents a collection of a nested pair, each comprising an entity, its proof of being a farmer, another entity, its proof of being a donkey, and a proof of an owning relation between them. This setting lets the representation of the pronoun *it*, namely, $\pi_1 \pi_1 \pi_2 \pi_2 u$ —which stays in the scope of u but is outside the scope of x —refer to the donkey in question.

This analysis naturally extends to the semantics of discourse including E-type anaphora. The SR of the mini-discourse (2) in our analysis is (8).

- (2) a. [A man]_{*i*} entered.
b. He_{*i*} whistled.

$$(8) \left[\begin{array}{l} v : \left[\begin{array}{l} u : \left[\begin{array}{l} x : \text{entity} \\ \text{man}(x) \end{array} \right] \\ \text{enter}(\pi_1 u) \\ \text{whistle}(\pi_1 \pi_1 v) \end{array} \right] \end{array} \right]$$

Note that (8) preserves the constituent structure of (2). The representation of *He* in (2b) is $\pi_1 \pi_1 v$, which correctly picks up the first element of a proof of the first sentence, even though it stays outside the scope of x . However, coming back to the compositionality problem and the questions in Sect. 1.2, the adequacy of (8) requires coherent answers to the following questions:

- (i) Does (8) correctly represent the meaning of (2)? This question needs particular attention, given that the meaning of a sentence is not its truth condition in proof-theoretic semantics.
(ii) If the answer to (i) is positive, and we adopt Assumption 1 in Sect. 1.2, namely, that the SR of (2a) is (9) below, then what are the answers to Questions 2 and 3?

$$(9) \left[\begin{array}{l} u : \left[\begin{array}{l} x : \text{entity} \\ \text{man}(x) \end{array} \right] \\ \text{enter}(\pi_1 u) \end{array} \right]$$

We will answer (i) positively in Sect. 2, by advocating a methodology which we call “inferences as tests”. As for (ii), for which no previous approaches in dependent type theory succeeds in providing a satisfactory answer, we will present the context-passing mechanism of dependent type semantics (DTS) in Sect. 3.

1.4 The Interpretation of Common Nouns in Dependent Type Theory

The SRs given in (7) and (8) are different from those proposed in the previous literature on natural language semantics using dependent type theory. Thus, according to the original proposal in Sundholm (1986), Ranta (1994) and Dávila-Pérez (1994), the SR of the sentence (1) is as given in (10) and that of (2) is as given in (11).⁸

$$(10) \left[u : \left[\begin{array}{l} x : \text{farmer} \\ y : \text{donkey} \\ \text{own}(x, y) \end{array} \right] \right] \left[\text{beat}(\pi_1 u, \pi_1 \pi_2 u) \right]$$

$$(11) \left[u : \left[\begin{array}{l} x : \text{man} \\ \text{enter}(x) \end{array} \right] \right] \left[\text{whistle}(\pi_1 u) \right]$$

The crucial difference between our approach and these previous approaches lies in the interpretation of common nouns; in our approach, common nouns such as *farmer*, *donkey*, and *man* are analyzed as predicates of type **entity** \rightarrow **type**.⁹ In the previous approaches with dependent types, by contrast, common nouns are treated as types; thus, the common noun *man* corresponds to a type **man**, not to a predicate. One attractive feature of the common-nouns-as-types view is that it can assign simplified SRs as shown in (10) and (11), as compared to the DTS-style SRs given in (7) and (8). This view has also been adopted by Modern Type Theory (MTT) (Luo 2012a,b; Chatzikiyiakidis and Luo 2014) and applied to a variety of issues in lexical semantics such as selectional restriction and coercion.

Despite its initial attractions, however, there is a problem with this approach.¹⁰ Consider the following example:

⁸Here we use the notation in DTS.

⁹Note that the notion of predicate in a dependently typed setting is different from that used in a simply typed setting—the type theory that underlies Montague semantics (Montague 1974) and the standard framework of formal semantics (Heim and Kratzer 1998). In the simply typed setting, we usually use base type e for the type of entities and t for the type of truth-values, that is, we have $e : \text{type}$ and $t : \text{type}$; given these base types, a one-place predicate is assigned type $e \rightarrow t$ and a two-place predicate type $e \rightarrow e \rightarrow t$, and so on. In our dependently typed setting, by contrast, we have **entity** : **type** and assign type **entity** \rightarrow **type** to one-place predicates and type **entity** \rightarrow **entity** \rightarrow **type** to two-place predicates. In this sense, a predicate in our setting is not a function from entities to truth-values (or, equivalently, a set of entities) but a function from entities to types (that is, propositions); note also that the meanings of types are specified in terms of inference rules, not in terms of their denotation.

¹⁰The problem of negated and conditional forms of predicational sentences is discussed in Tanaka et al. (2015). See also Chatzikiyiakidis and Luo (2016) and footnote 1 of that paper for more information.

(12) John is a student.

This is a predication sentence and the NP *a student* is a predicate nominal.¹¹ One way of looking at a predication sentence from the common-nouns-as-types view is to analyze it as a judgement

(13) *john* : **student**

where the common noun *student* corresponds to the type **student**. However, it is then not clear how to represent the negated sentence in (14) and the conditional sentence in (15), since a judgement itself cannot be negated nor appear in the antecedent of implication.

(14) John is not a student.

(15) If John is a student, I will be surprised.

Also, it is not clear how to represent complex constructions involving predicate nominals, such as (16a–c).

(16) a. John might be a doctor.

b. Susan became a painter.

c. Bob considers Mary a genius.

For instance, it seems natural to take the judgement *john* : **doctor** to be involved in the SR for the modal construction in (16); however, it is not evident how to give such an SR, or more generally, how to model the interaction of the common-nouns-as-types-view with the semantics of modals. Similarly for (16b) and (16c).

Another potential analysis is to adopt the Russell-Montague's analysis of predication sentences (Russell 1919; Montague 1974), according to which the predication of the form *t is an F* is analyzed as having the logical form $\exists x(Fx \wedge x = t)$. We can import this analysis into dependent type theory in the following way:

(17) a. John is a student. $\left[\begin{array}{l} x : \mathbf{student} \\ john =_{\mathbf{student}} x \end{array} \right]$
 b. John is not a student. $\neg \left[\begin{array}{l} x : \mathbf{student} \\ john =_{\mathbf{student}} x \end{array} \right]$
 c. If John is a student, then ... $\left[\begin{array}{l} x : \mathbf{student} \\ john =_{\mathbf{student}} x \end{array} \right] \rightarrow \dots$

This analysis allows us to represent the SR for *John is a student* as a type (i.e., a proposition), not a judgement, hence we can represent the negation and the implication as (17b) and (17c), respectively.

This analysis immediately faces a serious problem, however. Note that the equality in dependent type theory has the formation rule of the form:

$$\frac{A : \mathbf{type} \quad t : A \quad u : A}{t =_A u : \mathbf{type}} (=F)$$

¹¹For a recent survey on the interpretation of predication sentences and predicate nominals, see Mikkelsen (2011).

Accordingly, $john =_{\text{student}} x$ is well-formed only if $john : \text{student}$ is provable. Note also that negation and implication have the following formation rules¹²:

$$\frac{A : \text{type}}{\neg A : \text{type}} (\neg F) \qquad \frac{A : \text{type} \quad B : \text{type}}{A \rightarrow B : \text{type}} (\rightarrow F)$$

This means that if the negative form of SR in (17b) and the implicational form of SR in (17c) are well-formed, the judgement $john : \text{student}$ must be provable. In other words, the SRs in (17b) and (17c) presuppose that John is a student.¹³ Clearly, this is an undesirable consequence.

It is easily seen that the common-nouns-as-predicates view in our dependently typed setting avoids all these problems. Overall, an advantage of using type **entity** and assuming SRs like (7) and (8) rather than (10) and (11) is that it makes relatively easy to combine rich type structures and proof-theoretic machinery of dependent type theory with various analyses proposed in formal semantics of natural language. The DTS-style approach can make use of the expressive power of dependent type theory to analyze recalcitrant problems about discourse anaphora without losing the possibility of combining it with well-understood theories of formal semantics.¹⁴

Chatzikyriakidis and Luo (2016) propose a new analysis of negation and conditional in the context of MTT that sets out to avoid the problem of negated and conditional forms of predicational sentences.¹⁵ This proposal introduces the predicational form of a categorical (non-hypothetical) judgement as in (13) and then extends it to negated and hypothetical judgements, thereby avoiding the undesirable consequences. A detailed comparison between the two approaches has to be left for another occasion.

2 Verification Conditions of Discourse and Empirical Tests

Regardless of whether a theory states the meaning of a given sentence in the form of truth or verification conditions, its adequacy cannot be *directly* checked by our intuition nor linguistic data; what we can test are its predictions. Verification conditions, along with a proof theory that introduces them, predict entailment relations

¹²See Sect. 4.2 for more discussion on the formation rule of negation.

¹³We will give a more detailed discussion of the notion of presupposition in the context of dependent type theory in Sect. 4.

¹⁴Sundholm (1989) gives an analysis of generalized quantifiers in the framework of dependent type theory in which common nouns are treated as types. Tanaka (2014) points out that Sundholm's approach faces an "over-counting" problem in the interpretation of the proportional quantifier *most*, and provides a refined analysis by interpreting common nouns as predicates in the framework of DTS. Also, Tanaka et al. (2014) combines the framework of DTS with the semantics of modals that allows explicit quantification over possible worlds and applies it to the analysis of modal subordination phenomena.

¹⁵The analysis of negation goes back to Chatzikyriakidis and Luo (2014).

between sentences. Since we may judge an arbitrary entailment between sentences that includes a sentence in question, a set of such judgments serves as a set of tests for a semantic theory. We call this paradigm of testing a semantic theory as the *inferences as tests* paradigm (see also Sect. 3.7). For example, the sentences in (2) participate in the entailment relations listed in (18).

- (18) a. $[A \text{ man}]_i \text{ entered. He}_i \text{ whistled.} \Rightarrow A \text{ man entered.}$
 b. $[A \text{ man}]_i \text{ entered. He}_i \text{ whistled.} \Rightarrow A \text{ man whistled.}$
 c. $A \text{ man entered and whistled.} \Rightarrow [A \text{ man}]_i \text{ entered. He}_i \text{ whistled.}$

In DTS, (18a), (18b) and (18c) are predicted by constructing proofs for the inferences in (19) respectively,¹⁶ where \mathcal{K} is a set of background knowledge represented as a global context.¹⁷

$$\begin{aligned}
 (19) \text{ a. } \mathcal{K}, w : & \left[v : \left[u : \left[\begin{array}{l} x : \text{entity} \\ \text{man}(x) \\ \text{enter}(\pi_1 u) \\ \text{whistle}(\pi_1 \pi_1 v) \end{array} \right] \right] \right] \vdash \left[u : \left[\begin{array}{l} x : \text{entity} \\ \text{man}(x) \\ \text{enter}(\pi_1 u) \end{array} \right] \right] \text{ true} \\
 \text{b. } \mathcal{K}, w : & \left[v : \left[u : \left[\begin{array}{l} x : \text{entity} \\ \text{man}(x) \\ \text{enter}(\pi_1 u) \\ \text{whistle}(\pi_1 \pi_1 v) \end{array} \right] \right] \right] \vdash \left[u : \left[\begin{array}{l} x : \text{entity} \\ \text{man}(x) \\ \text{whistle}(\pi_1 u) \end{array} \right] \right] \text{ true} \\
 \text{c. } \mathcal{K}, w : & \left[u : \left[\begin{array}{l} x : \text{entity} \\ \text{man}(x) \\ \text{enter}(\pi_1 u) \\ \text{whistle}(\pi_1 u) \end{array} \right] \right] \vdash \left[v : \left[u : \left[\begin{array}{l} x : \text{entity} \\ \text{man}(x) \\ \text{enter}(\pi_1 u) \\ \text{whistle}(\pi_1 \pi_1 v) \end{array} \right] \right] \right] \text{ true}
 \end{aligned}$$

The inference (19a) is provable in a straightforward manner since the consequence of (19a) is just the first projection of the last premise. Assuming that the premise is inhabited by a term t , we obtain the following proof diagram in dependent type theory.

¹⁶The definition of the judgment of the form $\Gamma \vdash M : A$ is that there exists a proof diagram from the assumptions Γ to the consequence $M : A$. The judgment of the form $\Gamma \vdash A \text{ true}$ holds if and only if there exists a proof term M such that $\Gamma \vdash M : A$.

¹⁷In DTS, we assume that the global context \mathcal{K} at least includes:

- The basic ontological commitment (e.g. $\text{entity} : \text{type}$)
- The arities of predicates (e.g. $\text{whistle} : \text{entity} \rightarrow \text{type}$)
- Ontological knowledge (e.g. $\text{john} : \text{entity}$, $f : \left(u : \left[\begin{array}{l} x : \text{entity} \\ \text{cat}(x) \end{array} \right] \right) \rightarrow \text{animal}(\pi_1 u)$).

(20)

$$\frac{t : \left[\begin{array}{c} v : \left[\begin{array}{c} u : \left[\begin{array}{c} x : \text{entity} \\ \text{man}(x) \end{array} \right] \\ \text{enter}(\pi_1 u) \\ \text{whistle}(\pi_1 \pi_1 v) \end{array} \right] \end{array} \right]}{\pi_1 t : \left[\begin{array}{c} u : \left[\begin{array}{c} x : \text{entity} \\ \text{man}(x) \end{array} \right] \\ \text{enter}(\pi_1 u) \end{array} \right]} \quad (\Sigma E)$$

The entailments in (18b) and (18c) are even more complex, but we have proofs as shown in (21) and (22).¹⁸

(21)

$$\frac{t : \left[\begin{array}{c} v : \left[\begin{array}{c} u : \left[\begin{array}{c} x : \text{entity} \\ \text{man}(x) \end{array} \right] \\ \text{enter}(\pi_1 u) \\ \text{whistle}(\pi_1 \pi_1 v) \end{array} \right] \end{array} \right]}{\pi_1 t : \left[\begin{array}{c} u : \left[\begin{array}{c} x : \text{entity} \\ \text{man}(x) \end{array} \right] \\ \text{enter}(\pi_1 u) \end{array} \right]} \quad (\Sigma E) \quad \frac{t : \left[\begin{array}{c} v : \left[\begin{array}{c} u : \left[\begin{array}{c} x : \text{entity} \\ \text{man}(x) \end{array} \right] \\ \text{enter}(\pi_1 u) \\ \text{whistle}(\pi_1 \pi_1 v) \end{array} \right] \end{array} \right]}{\pi_2 t : \text{whistle}(\pi_1 \pi_1 \pi_1 t)} \quad (\Sigma E)$$

$$\frac{\pi_1 \pi_1 t : \left[\begin{array}{c} x : \text{entity} \\ \text{man}(x) \end{array} \right]}{\pi_1 \pi_1 t, \pi_2 t : \left[\begin{array}{c} x : \text{entity} \\ \text{man}(x) \\ \text{whistle}(\pi_1 u) \end{array} \right]} \quad (\Sigma I)$$

(22)

$$\frac{t : \left[\begin{array}{c} u : \left[\begin{array}{c} x : \text{entity} \\ \text{man}(x) \\ \text{enter}(\pi_1 u) \\ \text{whistle}(\pi_1 u) \end{array} \right] \end{array} \right]}{\pi_1 t : \left[\begin{array}{c} x : \text{entity} \\ \text{man}(x) \end{array} \right]} \quad (\Sigma E) \quad \frac{t : \left[\begin{array}{c} u : \left[\begin{array}{c} x : \text{entity} \\ \text{man}(x) \\ \text{enter}(\pi_1 u) \\ \text{whistle}(\pi_1 u) \end{array} \right] \end{array} \right]}{\pi_2 t : \left[\begin{array}{c} \text{enter}(\pi_1 \pi_1 t) \\ \text{whistle}(\pi_1 \pi_1 t) \end{array} \right]} \quad (\Sigma E)$$

$$\frac{\pi_1 t : \left[\begin{array}{c} x : \text{entity} \\ \text{man}(x) \end{array} \right]}{\pi_1 t, \pi_1 \pi_2 t : \left[\begin{array}{c} x : \text{entity} \\ \text{man}(x) \\ \text{enter}(\pi_1 u) \end{array} \right]} \quad (\Sigma I) \quad \frac{t : \left[\begin{array}{c} u : \left[\begin{array}{c} x : \text{entity} \\ \text{man}(x) \\ \text{enter}(\pi_1 u) \\ \text{whistle}(\pi_1 u) \end{array} \right] \end{array} \right]}{\pi_2 t : \left[\begin{array}{c} \text{enter}(\pi_1 \pi_1 t) \\ \text{whistle}(\pi_1 \pi_1 t) \end{array} \right]} \quad (\Sigma E)$$

$$\frac{\pi_2 \pi_2 t : \text{whistle}(\pi_1 \pi_1 t)}{\pi_2 \pi_2 t : \text{whistle}(\pi_1 \pi_1 (\pi_1 t, \pi_1 \pi_2 t))} \quad (\text{CONV})$$

$$\frac{(\pi_1 t, \pi_1 \pi_2 t) : \left[\begin{array}{c} x : \text{entity} \\ \text{man}(x) \\ \text{enter}(\pi_1 u) \end{array} \right]}{((\pi_1 t, \pi_1 \pi_2 t), \pi_2 \pi_2 t) : \left[\begin{array}{c} v : \left[\begin{array}{c} u : \left[\begin{array}{c} x : \text{entity} \\ \text{man}(x) \\ \text{enter}(\pi_1 u) \\ \text{whistle}(\pi_1 \pi_1 v) \end{array} \right] \end{array} \right] \end{array} \right]} \quad (\Sigma I)$$

Thus, all the inferences in (19) are provable. This gives a proof-theoretic account of the data in (18). A more precise formulation of the inferences-as-tests paradigm will be given in Sect. 3.7.

¹⁸The use of the (CONV) rule in (22) depends on the β -equivalence $\text{whistle}(\pi_1 \pi_1 t) =_\beta \text{whistle}(\pi_1 \pi_1 (\pi_1 t, \pi_1 \pi_2 t))$, which is omitted for the sake of space.

3 Toward Dependent Type Semantics

Regarding how one gets to a representation in dependent type theory from a given sentence (or a discourse), earlier works have provided different approaches. Ahn and Kolb (1990) proposed a translation algorithm from discourse representation structures to SRs in terms of dependent type theory. Dávila-Pérez (1995) proposed an integration of dependent type theory and Montagovian categorial grammar, and tried to provide a compositional setting (this has not been entirely successful, as discussed in Bekki 2014).

Then, the seminal work of Ranta (1994)—a compilation of this discipline in the mid-1990s—appeared as providing a theory that covers a broad range of linguistic phenomena including anaphora inaccessibility (see Sect. 3.6), descriptions, tense, and modality. However, Ranta’s work is initially formulated as a theory of sentence generation, which needs to be reformulated if one is to adopt it as a semantic component of a modern formal syntactic theory. This problem further involves how to formulate a problem of anaphora resolution and presupposition binding/accommodation as achieved in van der Sandt (1992), Geurts (1999), and Bos (2003) within the DRT framework.

Since then, researchers including Ranta himself have proposed various solutions, such as in Ranta (1994, Chap. 9), Krahmer and Piwek (1999), Piwek and Krahmer (2000), Mineshima (2008, 2013), and Bekki (2013, 2014). With regard to the problems of earlier approaches, please refer to the discussions in Bekki (2014).

Interestingly, the pursuit of this problem led to a paradigm called “anaphora resolution as proof construction” (Krahmer and Piwek 1999), which unified analyses of anaphora resolution and presupposition binding/accommodation, and analyses of sentential entailments.

The notable features of DTS, which are absent in other approaches using dependent type theory, or any other dynamic semantics, are its compositionality and the *double role* of SRs: On one hand, *the meaning* of a given sentence, which we assume is its verification condition as discussed in Sect. 1.3, is purely composed of lexical contributions of its words, in a standard way that most lexical grammars assume. On the other hand, *the context* for any proof construction for anaphora resolution or presupposition binding triggered within or around the sentence, is also composed of the same lexical contributions of words. Thus, in DTS, the SR of a word represents its contribution to both the meaning of a sentence and to the contexts for anaphora resolution/presupposition binding that the sentence is involved in.

DTS obtains these features, which provide a complete solution to the compositionality problem mentioned in Sect. 1.2, by employing two apparatuses: the context-passing mechanism and underspecified terms.

3.1 Context-Passing Mechanism

According to the discussion in Sect. 2, we assume that the SR of the mini-discourse (2) is (8). Moreover, if we maintain Assumption 1 in Sect. 1.2, the SR of (2a) is (9). Then Question 2 is understood as what is the SR of (2b), and Question 3 as how to construct the SR (8) from the SR (9) and the SR of (2b).

$$\begin{aligned}
 (2) \quad & \text{a. [A man]}_i \text{ entered.} \\
 & \text{b. He}_i \text{ whistled.} \\
 (8) \quad & \left[v : \left[u : \left[x : \mathbf{entity} \right. \right. \right. \\
 & \quad \left. \left. \mathbf{man}(x) \right] \right. \\
 & \quad \left. \mathbf{enter}(\pi_1 u) \right] \\
 & \quad \left. \mathbf{whistle}(\pi_1 \pi_1 v) \right] \\
 (9) \quad & \left[u : \left[x : \mathbf{entity} \right. \right. \\
 & \quad \left. \mathbf{man}(x) \right] \right] \\
 & \quad \left. \mathbf{enter}(\pi_1 u) \right]
 \end{aligned}$$

Since (9) is a subformula of the SR (8), the first guess for an answer to Question 2 is that the SR of (2b) should be its remainder, namely (23).

$$(23) \quad \mathbf{whistle}(\pi_1 \pi_1 v)$$

However, this does not work since v appears free in (23). Suppose we adopt a discourse composition rule (as an answer to Question 3) that takes the SRs M, N of two consecutive sentences and returns the following SR as a conjunction of these two sentences.

$$(24) \quad \left[\begin{array}{l} v : M \\ N \end{array} \right]$$

Then the variable-name convention of lambda calculus would rename this v if it appears free in N , which makes v in (23) unbound.

Since v is a proof of the first sentence (9), the immediate remedy for the first guess is to revise (23) and (24) so that the proof of the first sentence is passed to the SR of the second sentence. Let us tentatively assume that the SR of (2b) is as (25), a λ -abstraction of the type $\mathbf{whistle}(\pi_1 \pi_1 c)$ by the variable c .

$$(25) \quad \lambda c. \mathbf{whistle}(\pi_1 \pi_1 c)$$

Moreover, let us revise (24) as (26), by which the SR (8) is obtained in a compositional way from (9) and (25).

$$(26) \quad \left[\begin{array}{l} u : M \\ Nu \end{array} \right]$$

This remedy works well in this particular case but is not satisfactory if we further consider the following two cases:

- (I) M may also contain occurrences of discourse anaphora.

(II) The antecedent of a discourse anaphora in N may not be found in M and instead be found in the discourse that precedes M .

(I) suggests that we should λ -abstract not only N but M as well and pass M a proof of the discourse that precedes M . We call this the *local context* of M . (II) implies that what should be passed to N is not just the proof of M , but the local context of M plus the proof of M . Thus, the SR of (2a) is not as simple as that of (9), but should be revised as (27). This is the answer to Question 2 in DTS.

$$(27) \lambda c. \left[\begin{array}{l} u : \left[\begin{array}{l} x : \text{entity} \\ \text{man}(x) \end{array} \right] \\ \text{enter}(\pi_1 u) \end{array} \right]$$

The answer to Question 3 is that two sentential SRs are merged into one by the following *dynamic conjunction* operation.¹⁹

Definition 4 (*Dynamic conjunction*)

$$M; N \stackrel{\text{def}}{=} \lambda c. \left[\begin{array}{l} u : Mc \\ N(c, u) \end{array} \right] \quad \text{where } u \notin fv(N)$$

A local context c is a device to compose the SRs M and N of two consecutive sentences. First, the local context c for $M; N$ is passed to M , u being a proof of Mc , then the pair (c, u) is passed to N . This predicts the following asymmetry between M and N : discourse anaphora in N can refer to both antecedents in the local context and M , while discourse anaphora in M can only refer to antecedents in the local context.²⁰

Since the SR of (2a) is (27) and the SR of (2b) is as (25), the SR of the mini-discourse (2) is obtained by the dynamic conjunction between (27) and (25), which is calculated (and then β -reduced) as follows:

(28)

$$\begin{aligned} & \lambda c. \left[\begin{array}{l} u : \left[\begin{array}{l} x : \text{entity} \\ \text{man}(x) \end{array} \right] \\ \text{enter}(\pi_1 u) \end{array} \right]; \lambda c. \text{whistle}(\pi_1 \pi_1 \pi_2 c) \\ & \stackrel{\text{def}}{=} \lambda c. \left[\begin{array}{l} v : \left[\begin{array}{l} u : \left[\begin{array}{l} x : \text{entity} \\ \text{man}(x) \end{array} \right] \\ \text{enter}(\pi_1 u) \end{array} \right] \\ \text{whistle}(\pi_1 \pi_1 \pi_2(c, v)) \end{array} \right] =_{\beta} \lambda c. \left[\begin{array}{l} v : \left[\begin{array}{l} u : \left[\begin{array}{l} x : \text{entity} \\ \text{man}(x) \end{array} \right] \\ \text{enter}(\pi_1 u) \end{array} \right] \\ \text{whistle}(\pi_1 \pi_1 v) \end{array} \right] \end{aligned}$$

¹⁹The dynamic conjunction rule is an extension of the *progressive conjunction rule* in Ranta (1994) with a context-passing mechanism.

²⁰The types of the context c and the pair of contexts (c, u) are different. Thus, the two dynamic propositions M and N should be assigned different types. However, this does not require a polymorphic setting at the object-language level since M and N are preterms, and polymorphism is handled at the metalanguage level when type inference takes place.

3.2 Underspecified Terms

The analysis in the previous section that the SR of (2b) is given as (25) still has the following problems:

1. It is as if the hearer knew the antecedent of a pronoun before a preceding discourse is provided.
2. An antecedent of anaphora is ambiguous in general. For example, in the most *natural* readings of the following two sentences, *it* in (29a) refers to *a lion*, while *it* refers to *a zebra* in (29b).²¹

- (29) a. A Lion hunted a zebra. It was hungry.
 b. A Lion hunted a zebra. It was delicious.

According to our discussion so far, the sentences (29a) and (29b) have the SRs (30) and (31):

$$(30) \left[\begin{array}{c} u_3 : \left[\begin{array}{c} u_1 : \left[\begin{array}{c} x : \text{entity} \\ \text{lion}(x) \end{array} \right] \\ u_2 : \left[\begin{array}{c} y : \text{entity} \\ \text{zebra}(y) \end{array} \right] \\ \text{hunt}(\pi_1 u_1, \pi_1 u_2) \end{array} \right] \\ \text{hungry}(\pi_1 \pi_1 u_3) \end{array} \right]$$

$$(31) \left[\begin{array}{c} u_3 : \left[\begin{array}{c} u_1 : \left[\begin{array}{c} x : \text{entity} \\ \text{lion}(x) \end{array} \right] \\ u_2 : \left[\begin{array}{c} y : \text{entity} \\ \text{zebra}(y) \end{array} \right] \\ \text{hunt}(\pi_1 u_1, \pi_1 u_2) \end{array} \right] \\ \text{delicious}(\pi_1 \pi_1 \pi_2 u_3) \end{array} \right]$$

This means that the SRs of the second sentences of (29a) and (29b) are given as follows:

- (32) a. $\lambda c. \text{hungry}(\pi_1 \pi_1 \pi_2 c)$
 b. $\lambda c. \text{delicious}(\pi_1 \pi_1 \pi_2 \pi_2 c)$

How can we specify an SR of the pronoun *it* that incorporates the difference between (32a) and (32b)? What do the two terms $\pi_1 \pi_1 \pi_2 c$ and $\pi_1 \pi_1 \pi_2 \pi_2 c$ have in common? The answer to the latter is that they are of the same type under the same global context:

²¹Examples taken from “The Winograd Schema Challenge” (Levesque 2011), slightly adapted.

(33)

$$\begin{aligned}
& \mathcal{K}, u_3 : \left[\begin{array}{l} u_1 : \left[\begin{array}{l} x : \mathbf{entity} \\ \mathbf{lion}(x) \end{array} \right] \\ u_2 : \left[\begin{array}{l} y : \mathbf{entity} \\ \mathbf{zebra}(y) \end{array} \right] \\ \mathbf{hunt}(\pi_1 u_1, \pi_1 u_2) \end{array} \right] \vdash \pi_1 \pi_1 u_3 : \mathbf{entity} \\
& \mathcal{K}, u_3 : \left[\begin{array}{l} u_1 : \left[\begin{array}{l} x : \mathbf{entity} \\ \mathbf{lion}(x) \end{array} \right] \\ u_2 : \left[\begin{array}{l} y : \mathbf{entity} \\ \mathbf{zebra}(y) \end{array} \right] \\ \mathbf{hunt}(\pi_1 u_1, \pi_1 u_2) \end{array} \right] \vdash \pi_1 \pi_1 \pi_2 u_3 : \mathbf{entity}
\end{aligned}$$

Now we are ready to give a full answer to Question 2, including specifying the SR of a pronoun. The idea, which plays a central role in the discourse representation of DTS, is that anaphora (and presupposition triggers) are represented by *underspecified terms* $@_i$, which obey a certain typing judgment. In the above example, the SRs of the second sentences of (29a) and (29b) are the following:

- (34) a. $\lambda c.\mathbf{hungry}(@_1 c)$
 b. $\lambda c.\mathbf{delicious}(@_2 c)$

where $@_1$ and $@_2$ are different underspecified terms, but both of them obey the following type judgment.

$$(35) \quad \mathcal{K}, u_3 : \left[\begin{array}{l} u_1 : \left[\begin{array}{l} x : \mathbf{entity} \\ \mathbf{lion}(x) \end{array} \right] \\ u_2 : \left[\begin{array}{l} y : \mathbf{entity} \\ \mathbf{zebra}(y) \end{array} \right] \\ \mathbf{hunt}(\pi_1 u_1, \pi_1 u_2) \end{array} \right] \vdash @_i u_3 : \mathbf{entity}$$

Thus, the SR of (2b) is, finally, fixed as follows:

- (36) $\lambda c.\mathbf{whistle}(@_1 c)$

3.3 Syntactic Calculus and Semantic Composition

Along with a syntactic calculus, through the disambiguation process if necessary, the SR of a sentence is composed. This is a preterm of dependent type theory extended with underspecified terms.

The lexical items required to derive these sentences are listed in Table 2. Throughout this paper, DTS is presented as a semantic component of combinatory categorial grammar (Steedman 1996), but it is naturally available for other lexical grammars as well.

Table 2 Lexical items in DTS

PF	CCG categories	Semantic representations in DTS
if	$S/S/S$	$\lambda p.\lambda q.\lambda c.(u : pc) \rightarrow (q(c, u))$
every_{nom}	$T/(T \backslash NP)/N$	$\lambda n.\lambda p.\lambda c.\left(u : \begin{bmatrix} x : \text{entity} \\ nxc \end{bmatrix}\right) \rightarrow (p(\pi_1 u)(c, u))$
every_{acc}	$T \backslash (T/NP)/N$	$\lambda n.\lambda p.\lambda x.\lambda c.\left(v : \begin{bmatrix} y : \text{entity} \\ nyc \end{bmatrix}\right) \rightarrow (p(\pi_1 v)x(c, v))$
a_{nom}	$T/(T \backslash NP)/N$	$\lambda n.\lambda p.\lambda c.\left[\begin{array}{c} u : \begin{bmatrix} x : \text{entity} \\ nxc \end{bmatrix} \\ p(\pi_1 u)(c, u) \end{array}\right]$
a_{acc}	$T \backslash (T/NP)/N$	$\lambda n.\lambda p.\lambda x.\lambda c.\left[\begin{array}{c} v : \begin{bmatrix} y : \text{entity} \\ nyc \end{bmatrix} \\ p(\pi_1 v)x(c, v) \end{array}\right]$
man	N	$\lambda x.\lambda c.\text{man}(x)$
who	$N \backslash N/(S \backslash NP)$	$\lambda p.\lambda n.\lambda x.\lambda c.\left[\begin{array}{c} nxc \\ pxc \end{array}\right]$
whom	$N \backslash N/(S/NP)$	$\lambda p.\lambda n.\lambda x.\lambda c.\left[\begin{array}{c} nxc \\ pxc \end{array}\right]$
entered	$S \backslash NP$	$\lambda x.\lambda c.\text{enter}(x)$
whistled	$S \backslash NP$	$\lambda x.\lambda c.\text{whistle}(x)$
he_{nom}^i	$T/(T \backslash NP)$	$\lambda p.\lambda c.p(@_i c)$
it_{acc}^j	$T \backslash (T/NP)$	$\lambda p.\lambda c.p(@_j c)$

Here, $@_i$ and $@_j$ are underspecified terms

The conditional *if* and the universal quantifier *every* are constructed from dependent function types, while the indefinite article *a* is constructed from a dependent product type, following Sundholm (1986). The relativizer *who* takes a subjectless sentence and a common noun, and statically conjoins them.

Following the “presupposition as anaphora” paradigm advocated in van der Sandt and Geurts (1991), van der Sandt (1992) and Geurts (1999) that anaphora resolution and presupposition binding are the same operation, DTS uniformly represents anaphora and presupposition triggers as underspecified terms.

To see this, let us take an example of a derivation of (2). The sentences (2a) and (2b) are derived as (37) and (38), respectively.

$$\begin{array}{c}
 (37) \quad \frac{\frac{\frac{A}{T/(T \setminus NP)/N} \quad \frac{\text{man}}{N}}{\lambda n. \lambda p. \lambda c. \left[u : \begin{array}{c} x : \text{entity} \\ nxc \\ p(\pi_1 u)(c, u) \end{array} \right]} \quad \lambda x. \lambda c. \text{man}(x)}{>} \\
 \frac{\frac{\frac{T/(T \setminus NP)}{\lambda p. \lambda c. \left[u : \begin{array}{c} x : \text{entity} \\ \text{man}(x) \\ p(\pi_1 u)(c, u) \end{array} \right]} \quad \frac{\frac{\text{entered}}{S \setminus NP}}{\lambda x. \lambda c. \text{enter}(x)}}{>} \\
 S \\
 \lambda c. \left[u : \begin{array}{c} x : \text{entity} \\ \text{man}(x) \\ \text{enter}(\pi_1 u) \end{array} \right]
 \end{array}$$

$$\begin{array}{c}
 (38) \quad \frac{\frac{\text{He}}{T/(T \setminus NP)} \quad \frac{\text{whistled}}{S \setminus NP}}{\lambda p. \lambda c. p(@_i c) \quad \lambda x. \lambda c. \text{whistle}(x)} > \\
 S \\
 \lambda c. \text{whistle}(@_i c)
 \end{array}$$

Then, the dynamic conjunction operation is applied to (37) and (38), yielding an SR for the mini-discourse (2), as follows.

$$(39) \quad \lambda c. \left[u : \begin{array}{c} x : \text{entity} \\ \text{man}(x) \\ \text{enter}(\pi_1 u) \end{array} \right]; \lambda c. \text{whistle}(@_1 c) = \lambda c. \left[v : \begin{array}{c} u : \begin{array}{c} x : \text{entity} \\ \text{man}(x) \\ \text{enter}(\pi_1 u) \end{array} \\ \text{whistle}(@_1(c, v)) \end{array} \right]$$

3.4 Type Checking as the Felicity Condition

The anaphora resolution for the SR s is launched by type checking of the judgment $\mathcal{K}, \delta : \text{type} \vdash s : \delta \rightarrow \text{type}$. This reflects a requirement that the SR of a sentence must be of the sort **type** under an assumption that the SR of the preceding discourse is of type δ , which we call the *felicity condition* of a sentence. The variable δ will be instantiated with the type \top when there is no preceding discourse for s .

Following Mineshima (2008, 2013), Bekki (2013, 2014), and Bekki and Sato (2015), the presupposition projection is calculated via type-checking. In DTS, the type checking calculates, as a side effect, the judgment that each of $@_i$ must satisfy.²² This reflects a view that presupposition is about the well-formedness or the felicity of a sentence, not about its verification condition.

²²Bekki and Sato (2015) defined a fragment of dependent type theory with underspecified terms which has a decidable type-checking and inference algorithms.

The felicity condition invokes the type-checking algorithm presented in Bekki and Sato (2015), which returns the type that the underspecified term $@_1$ contained in the above SR must be assigned under a given global context, as (40):

$$(40) \quad \mathcal{K}, \delta : \mathbf{type}, c : \delta \vdash @_1 : \left[\begin{array}{c} \delta \\ \left[\begin{array}{c} u : \left[\begin{array}{c} x : \mathbf{entity} \\ \mathbf{man}(x) \end{array} \right] \\ \mathbf{enter}(\pi_1 u) \end{array} \right] \end{array} \right] \rightarrow \mathbf{entity}$$

The underspecified term $@_1$ could be any term that satisfies (40), but the type (40) must be inhabited for this mini-discourse to be felicitously uttered. Now the hearer of this mini-discourse has the two options: *binding* or *accommodation*.

3.5 Anaphora Resolution and Presupposition Binding

Following the “anaphora resolution as proof construction” paradigm in Krahmer and Piwek (1999) and Piwek and Krahmer (2000), anaphora resolution and presupposition binding are uniformly treated as a proof search for a term that can replace each underspecified term.

The proof search for (40) finds that the type (40) inhabits a proof term $\lambda c.\pi_1\pi_1\pi_2c$ as shown in (41).

(41)

$$\frac{\frac{\frac{\frac{\frac{\frac{\delta}{c : \left[\begin{array}{c} u : \left[\begin{array}{c} x : \mathbf{entity} \\ \mathbf{man}(x) \end{array} \right] \\ \mathbf{enter}(\pi_1 u) \end{array} \right] \end{array} \right] \quad (\Sigma E)}{\pi_2 c : \left[\begin{array}{c} u : \left[\begin{array}{c} x : \mathbf{entity} \\ \mathbf{man}(x) \end{array} \right] \\ \mathbf{enter}(\pi_1 u) \end{array} \right] \quad (\Sigma E)}{\pi_1 \pi_2 c : \left[\begin{array}{c} x : \mathbf{entity} \\ \mathbf{man}(x) \end{array} \right] \quad (\Sigma E)}{\pi_1 \pi_1 \pi_2 c : \mathbf{entity} \quad (\Sigma E)}{\lambda c.\pi_1 \pi_1 \pi_2 c : \left[\begin{array}{c} \delta \\ \left[\begin{array}{c} u : \left[\begin{array}{c} x : \mathbf{entity} \\ \mathbf{man}(x) \end{array} \right] \\ \mathbf{enter}(\pi_1 u) \end{array} \right] \end{array} \right] \rightarrow \mathbf{entity} \quad (\Pi I), 1$$

The first option for the hearer is to assume the following equation (at the top level of inferences).

$$(42) \ @_1 = \lambda c. \pi_1 \pi_1 \pi_2 c : \left[\begin{array}{c} \delta \\ u : \left[\begin{array}{c} x : \text{entity} \\ \text{man}(x) \end{array} \right] \\ \text{enter}(\pi_1 u) \end{array} \right] \rightarrow \text{entity}$$

This process corresponds to the *binding* of the presupposition triggered by $@_1$. In words, there is an entity in a given context to which the pronoun *He* can refer. This is exactly the presupposition that the pronoun triggers. Anaphora in (30) and (31) can be resolved in this way, and these resolutions correspond, respectively, to the anaphoric links established in (29a) and (29b).

More formally, anaphora resolution and presupposition binding are the processes defined as below (Bekki 2014).

Definition 5 (*Anaphora resolution/presupposition binding in DTS*) Suppose that $\Gamma \vdash @_i : A$ and $\Gamma \vdash M : A$. Then a resolution of $@_i$ by M under the context Γ is an equation $@_i = M : A$.

Another option in sentence understanding is to choose not to search for a proof, and just assume that there is a term $@_1$ that satisfies the judgment (40). This process corresponds to the *accommodation* of the presupposition triggered by $@_1$.

Note that the DTS version of accommodation does not involve any *transformation of the representations*, unlike the case in van der Sandt (1992) and Krahmer and Piwek (1999).

3.6 (In)accessibility

While the accessible anaphoric links are well represented, the inaccessible anaphora such as (43), listed in Karttunen (1976), are simply not representable with dependent types, as argued in Ranta (1994), Dávila-Pérez (1994), and Fox (1994a).

- (43) a. Everybody bought [a car]_i. *It_i stinks.
 b. If John bought [a car]_i, it_i must be a Porsche. *It_i stinks.
 c. John didn't buy [a car]_i. *It_i stinks.

This is because universal quantification, implication, and negation are represented by dependent function types that are data types of functions, from which the intended antecedents cannot be picked up. This is an explanation based purely on the structures of proofs, which is fundamentally different from the explanation in DRT and other dynamic semantics.

3.7 Inferences as Tests

Now we can formally state the *inferences as tests* paradigm in DTS as follows: Let S_1, \dots, S_n, S_{n+1} ($n \geq 1$) be a sequence of sentences such that $S_1, \dots, S_n \Rightarrow$

S_{n+1} empirically (i.e., one cannot conceive of a situation in which S_1, \dots, S_n is true and S_{n+1} is not). This inference relation is explained if their respective SRs $S'_1, \dots, S'_n, S'_{n+1}$ satisfy the relation $\mathcal{K}, u : (S'_1; \dots; S'_n)() \vdash S'_{n+1}(), u \text{ true}$, where $()$ is the proof term for \top . Since there is no discourse that precedes S_1 , the type of the preceding discourse for S_1, \dots, S_n, S_{n+1} is given as \top as mentioned in Sect. 3.4.

4 Presuppositions as Type Inferences

What is characteristic of our treatment of anaphora and presupposition is that the process of resolving anaphora and presupposition is analyzed as the process of type-checking/inference. In this section, we will see in more detail how our underspecified semantics can account for various presupposition phenomena discussed in the formal semantics literature.

4.1 Presupposition Phenomena

We first focus on the existence presupposition triggered by a definite description. Some other type of presupposition triggers will be discussed subsequently.

There are two characteristic properties of presuppositions.²³ First, a presupposition *projects* out of certain embedded contexts. Thus, we can naturally infer that France has a king not only from (44a) but also when the description occurs in the negated sentence (44b) or in the antecedent of a conditional (44c).

- (44) a. The king of France is wise. \Rightarrow France has a king.
- b. The king of France is not wise. \Rightarrow France has a king.
- c. If the king of France is wise, I will be happy. \Rightarrow France has a king.

These examples show a striking contrast with the case of entailment as exemplified in the following examples.

- (45) a. John is an American pianist. \Rightarrow John is American.
- b. John is not an American pianist. \nRightarrow John is American.
- c. If John is an American pianist, he is skillful. \nRightarrow John is American.

The sentence (45a) entails that John is American, but this entailment does not survive in the environments (45b, c), in contrast to the case of presupposition in (45b, c)

Second, a presupposition is *filtered* when it occurs in contexts such as the second sentence in the conjunction (46a) or in the conditional (46b).

- (46) a. France has a king and the king of France is wise.
- b. If France has a king, the king of France is wise.

²³See Soames (1989) and Beaver (2001) for useful surveys on the topic.

The problem posed by these examples is to account for the fact that while a simple sentence *The king of France is wise* presupposes that France has a king, neither (46a) nor (46b) inherits this presupposition.

4.2 Projection

The projection and filtering inferences of presupposition can be naturally accounted for within the framework of DTS. We will take a look at each in turn.

Consider first how to derive the presupposition projected out of the negated sentence in (44b). Note first that as is standard in constructive logic, negation is defined to be an implication of the form $\neg A \equiv A \rightarrow \perp$, where \perp is the absurdity type, i.e., the type that has no inhabitants. Given the formation rule for the absurdity type shown on the left below, the formation rule for negation can be derived as on the right:

$$\frac{}{\perp : \text{type}} \quad (\perp F) \qquad \frac{A : \text{type}}{\neg A : \text{type}} \quad (\neg F)$$

We analyze the definite article *the* as follows. Here, a lexical entry is specified in the form *Surf*; *Syn*; *Sem*, where *Surf* is a surface form, *Syn* a CCG syntactic category, and *Sem* a semantic representation in DTS.

$$(47) \text{ the}; (S/(S \setminus NP))/N; \lambda n. \lambda p. \lambda c. p \left(\pi_1 \left(@_i c : \left[\begin{smallmatrix} x : \mathbf{entity} \\ nxc \end{smallmatrix} \right] \right) \right) c$$

A term of the form $@_i c : \Lambda$ is called *type annotation* and specifies that the term $@_i c$ has type Λ . In the case of (47), the term $@_i c$ is annotated with a Σ -type $(x : \mathbf{entity}) \times nxc$. This means that the underspecified term $@_i$ is a function that takes a local context c as argument and returns a term having the Σ -type. In this case, such a term is a pair of an entity x and a proof that x satisfies the condition n . Then its first projection, i.e., an entity x , is applied to a given predicate p .

The relevant part of the derivation tree for the sentence (44b) runs as follows.²⁴

$$(48) \quad \frac{\frac{\frac{\text{The}}{(S/(S \setminus NP))/N} \quad \frac{\text{king of France}}{N}}{\lambda npc. p \left(\pi_1 \left(@_1 c : \left[\begin{smallmatrix} x : \mathbf{entity} \\ nxc \end{smallmatrix} \right] \right) \right) c} \quad \lambda xc. \mathbf{kof}(x)}{NP/(S \setminus NP)} \quad > \quad \frac{\frac{\text{is not}}{S \setminus NP/(S \setminus NP)} \quad \frac{\text{wise}}{S \setminus NP}}{\lambda pxc. \neg(pxc)} \quad \lambda xc. \mathbf{wise}(x)}{S \setminus NP} \quad > \\ \lambda pc. p \left(\pi_1 \left(@_1 c : \left[\begin{smallmatrix} x : \mathbf{entity} \\ \mathbf{kof}(x) \end{smallmatrix} \right] \right) \right) c & \quad \lambda xc. \neg \mathbf{wise}(x) \\ \hline S & \\ \lambda c. \neg \mathbf{wise} \left(\pi_1 \left(@_1 c : \left[\begin{smallmatrix} x : \mathbf{entity} \\ \mathbf{kof}(x) \end{smallmatrix} \right] \right) \right) & \quad >$$

As we saw in Sect. 3.4, the anaphora/presupposition resolution for the SR A is triggered by the judgement $\mathcal{K}, \delta : \text{type} \vdash A : \delta \rightarrow \text{type}$, where \mathcal{K} is a global context

²⁴We abbreviate $\lambda x_1 \dots \lambda x_n. M$ as $\lambda x_1 \dots x_n. M$.

Assuming that **wise** : **entity** \rightarrow **type** is in the global context, the proof that the SR yielded by (48) is well-formed runs as follows.

$$\begin{array}{c}
(49) \quad \frac{\frac{\frac{}{\text{wise} : \text{entity} \rightarrow \text{type}}}{@_1 : \delta \rightarrow \left[\begin{array}{l} x : \text{entity} \\ \mathbf{kof}(x) \end{array} \right]} \overline{c : \delta}^1}{@_1 c : \left[\begin{array}{l} x : \text{entity} \\ \mathbf{kof}(x) \end{array} \right]} \quad (\Pi E)}{\pi_1 \left(@_1 c : \left[\begin{array}{l} x : \text{entity} \\ \mathbf{kof}(x) \end{array} \right] \right) : \text{entity}} \quad (\Sigma E)} \\
\frac{\text{wise} \left(\pi_1 \left(@_1 c : \left[\begin{array}{l} x : \text{entity} \\ \mathbf{kof}(x) \end{array} \right] \right) \right) : \text{type}}{\neg \text{wise} \left(\pi_1 \left(@_1 c : \left[\begin{array}{l} x : \text{entity} \\ \mathbf{kof}(x) \end{array} \right] \right) \right) : \text{type}} \quad (\neg F)} \\
\frac{}{\lambda c. \neg \text{wise} \left(\pi_1 \left(@_1 c : \left[\begin{array}{l} x : \text{entity} \\ \mathbf{kof}(x) \end{array} \right] \right) \right) : \delta \rightarrow \text{type}} \quad (\Pi I), 1
\end{array}$$

What is presupposed by the original sentence in (44b) can be read off from the open branch ending with the judgment having the underspecified term @₁. For the given SR to be well-formed, one has to find a term that can replace @₁ in (50).

$$(50) \quad @_1 : \delta \rightarrow \begin{bmatrix} x : \mathbf{entity} \\ \mathbf{kof}(x) \end{bmatrix}$$

That is to say, given the input context represented by δ , one has to find a proof term for the proposition that there is a king of France. In this way, we can derive the existence presupposition for the negated sentence (44b), as well as for the positive counterpart (44a). As is easily seen by the definition $\neg A \equiv A \rightarrow \perp$, the same inference is triggered for the antecedent of a conditional sentence in (44c). Thus we can also account for presupposition projection for conditionals as exemplified in (44c).

4.3 Filtering

The present account can explain the filtering phenomena in (46) without further stipulation. The relevant derivation for (46a) goes in the same way as the case of anaphora resolution for the mini-discourse (2) discussed in Sect. 3. Here we will take a brief look at the case of a conditional sentence in (46b).

To begin with, the SR of the sentence (46b) is compositionally obtained via the following derivation tree.

(51)

$$\begin{array}{c}
\frac{\text{If}}{S/S/S} \quad \frac{\text{France has a king}}{S} \\
\frac{\lambda p q c. (u : pc) \rightarrow q(c, u) \quad : \lambda c. \left[\begin{array}{c} x : \text{entity} \\ \text{kof}(x) \end{array} \right]}{S/S} > \frac{\text{the king of France is wise}}{S} \\
\frac{\lambda q c. \left(u : \left[\begin{array}{c} x : \text{entity} \\ \text{kof}(x) \end{array} \right] \right) \rightarrow q(c, u) \quad : \lambda c. \text{wise} \left(\pi_1 \left(@_1 c : \left[\begin{array}{c} x : \text{entity} \\ \text{kof}(x) \end{array} \right] \right) \right)}{S} > \\
\frac{}{S} \\
: \lambda c. \left(u : \left[\begin{array}{c} x : \text{entity} \\ \text{kof}(x) \end{array} \right] \right) \rightarrow \text{wise} \left(\pi_1 \left(@_1(c, u) : \left[\begin{array}{c} x : \text{entity} \\ \text{kof}(x) \end{array} \right] \right) \right)
\end{array}$$

Then the following type inference is triggered:

(52)

$$\begin{array}{c}
\frac{\frac{\frac{}{c : \delta} \quad \frac{\frac{}{u : \left[\begin{array}{c} x : \text{entity} \\ \text{kof}(x) \end{array} \right]}{1}}{2}}{\frac{}{c, u : \left[\begin{array}{c} x : \text{entity} \\ \text{kof}(x) \end{array} \right]}{(\Sigma I)}}}{\frac{}{@_1 : \left[\begin{array}{c} \delta \\ \left[\begin{array}{c} x : \text{entity} \\ \text{kof}(x) \end{array} \right] \end{array} \right] \rightarrow \left[\begin{array}{c} x : \text{entity} \\ \text{kof}(x) \end{array} \right]}{(\Pi E)}}} \\
\frac{}{@_1(c, u) : \left[\begin{array}{c} x : \text{entity} \\ \text{kof}(x) \end{array} \right]} \\
\frac{\frac{}{\text{wise} : \text{entity} \rightarrow \text{type}} \quad \frac{}{\pi_1 \left(@_1(c, u) : \left[\begin{array}{c} x : \text{entity} \\ \text{kof}(x) \end{array} \right] \right) : \text{entity}}{(\Sigma E)} \quad \frac{}{(\Pi E)} \\
\frac{\frac{\vdots}{\left[\begin{array}{c} x : \text{entity} \\ \text{kof}(x) \end{array} \right] : \text{type}} \quad \frac{}{\text{wise} \left(\pi_1 \left(@_1(c, u) : \left[\begin{array}{c} x : \text{entity} \\ \text{kof}(x) \end{array} \right] \right) \right) : \text{type}}{(\Pi F), 1}} \\
\frac{\left(u : \left[\begin{array}{c} x : \text{entity} \\ \text{kof}(x) \end{array} \right] \right) \rightarrow \text{wise} \left(\pi_1 \left(@_1(c, u) : \left[\begin{array}{c} x : \text{entity} \\ \text{kof}(x) \end{array} \right] \right) \right) : \text{type}}{(\Pi I), 2} \\
\lambda c. \left(u : \left[\begin{array}{c} x : \text{entity} \\ \text{kof}(x) \end{array} \right] \right) \rightarrow \text{wise} \left(\pi_1 \left(@_1(c, u) : \left[\begin{array}{c} x : \text{entity} \\ \text{kof}(x) \end{array} \right] \right) \right) : \delta \rightarrow \text{type}
\end{array}$$

In this case, one can find a term that can replace $@_1$ without using the information in the context δ , namely, the term $\lambda c. \pi_2 c$. This accounts for the fact that the presuppositional inference is filtered out in sentences like (46a, b). By substituting $\lambda c. \pi_2 c$ for $@_1$, one can obtain a fully specified representation for the sentence (46b), which captures the intended reading.

$$(53) \quad \lambda c. \left(u : \left[\begin{array}{c} x : \text{entity} \\ \text{kof}(x) \end{array} \right] \right) \rightarrow \text{wise}(\pi_1 u)$$

4.4 Bridging Inferences and Gender Presuppositions of Pronouns

It is often the case that the information that is not explicitly provided in a discourse plays a role in the process of presupposition resolution. There are two important examples. One is the so-called *bridging* inference (Clark 1975).

(54) John bought a car. He checked the motor.

The definite description *the motor* in the second sentence does not have an overt antecedent, but the hearer can easily infer the existence of a motor using the implicit knowledge that a car has a motor. Such a bridging inference is special in that the antecedent is inferred using some relevant background knowledge, with the help of the information explicitly provided in a previous discourse (Krahmer and Piwek 1999). Due to this inferential character, it is not straightforward to handle bridging inferences in standard dynamic theories of anaphora such as DRT (van der Sandt 1992; Geurts 1999; Kamp et al. 2011).

The other is concerned with the gender information of pronouns. It has been widely observed that pronouns introduce gender information as presupposition.²⁵ In the case of (54), the assumption that John is male plays a role in identifying the antecedent of *he* with *John*.

In DTS, the process of anaphora/presupposition resolution essentially involves a process of proof search. As a consequence, it can treat presupposition resolution and inference with implicit world knowledge in a unified way.

As an illustration, consider how to handle the example in (54). In a similar way to the example (2) discussed in Sect. 3, the SRs for the first and the second sentences in (54) can be derived as (55a) and (55b), respectively.

$$(55) \text{ a. } \lambda c. \left[\begin{array}{l} v : \left[\begin{array}{l} x : \mathbf{entity} \\ \mathbf{car}(x) \end{array} \right] \\ \mathbf{buy}(j, \pi_1 v) \end{array} \right] \\ \text{ b. } \lambda c. \mathbf{check} \left(\pi_1 \left(@_1 c : \left[\begin{array}{l} x : \mathbf{entity} \\ \mathbf{male}(x) \end{array} \right] \right), \pi_1 \left(@_2 c : \left[\begin{array}{l} x : \mathbf{entity} \\ \mathbf{motor}(x) \end{array} \right] \right) \right)$$

Here the pronoun *he* introduces the underspecified term $@_1$ to which the Σ -type $(x : \mathbf{entity}) \times \mathbf{male}(x)$ is annotated. Then by combining the two SRs using the dynamic conjunction and then simplifying the resulting expression, the SR for the whole discourse in (54) is derived as follows.

$$(56) \lambda c. \left[\begin{array}{l} u : \left[\begin{array}{l} v : \left[\begin{array}{l} x : \mathbf{entity} \\ \mathbf{car}(x) \end{array} \right] \\ \mathbf{buy}(j, \pi_1 v) \end{array} \right] \\ \mathbf{check} \left(\pi_1 \left(@_1(c, u) : \left[\begin{array}{l} x : \mathbf{entity} \\ \mathbf{male}(x) \end{array} \right] \right), \pi_1 \left(@_2(c, u) : \left[\begin{array}{l} x : \mathbf{entity} \\ \mathbf{motor}(x) \end{array} \right] \right) \right) \end{array} \right]$$

It is easily checked that for the SR (56) to have the type $\delta \rightarrow \mathbf{type}$ given the context \mathcal{K} , $\delta : \mathbf{type}$, the underspecified terms $@_1$ and $@_2$ are required to have the types in (57a) and (57b), respectively.

²⁵The treatment of gender information of pronoun as presuppositions goes back at least to Cooper (1983). See Sudo (2012) for a recent discussion.

$$\begin{aligned}
 (57) \quad & \text{a. } @_1 : \left[\begin{array}{c} \delta \\ v : \left[\begin{array}{c} x : \text{entity} \\ \text{car}(x) \end{array} \right] \\ \text{buy}(j, \pi_1 v) \end{array} \right] \rightarrow \left[\begin{array}{c} x : \text{entity} \\ \text{male}(x) \end{array} \right] \\
 & \text{b. } @_2 : \left[\begin{array}{c} \delta \\ v : \left[\begin{array}{c} x : \text{entity} \\ \text{car}(x) \end{array} \right] \\ \text{buy}(j, \pi_1 v) \end{array} \right] \rightarrow \left[\begin{array}{c} x : \text{entity} \\ \text{motor}(x) \end{array} \right]
 \end{aligned}$$

Let us assume that the global context \mathcal{K} contains the judgements in (58) which represent the background knowledge.

$$(58) \quad j : \text{entity}, k : \text{male}(j), f : \left(u : \left[\begin{array}{c} x : \text{entity} \\ \text{car}(x) \end{array} \right] \right) \rightarrow \left[\begin{array}{c} v : \left[\begin{array}{c} y : \text{entity} \\ \text{motor}(y) \end{array} \right] \\ \text{have}(\pi_1 u, \pi_1 v) \end{array} \right]$$

Then one can construct a term having the type in (57a) as $\lambda c. (j, k)$ and one having the type in (57b) as $\lambda c. \pi_1(f(\pi_1(\pi_2 c)))$. Substituting these terms for $@_1$ and $@_2$ in (56), respectively, we can obtain the SR in (59), which captures the correct information derivable from the discourse in (54).

$$(59) \quad \lambda c. \left[\begin{array}{c} u : \left[\begin{array}{c} v : \left[\begin{array}{c} x : \text{entity} \\ \text{car}(x) \end{array} \right] \\ \text{buy}(j, \pi_1 v) \end{array} \right] \\ \text{check}(j, \pi_1(f(\pi_1 u))) \end{array} \right]$$

These examples suggest that presuppositions are resolved in various ways. In simple cases, the presupposed information is merely identified with some element present in the previous discourse via presupposition binding or copied in a suitable place via presupposition accommodation. These possibilities are accounted for within the framework of DRT (van der Sandt 1992; Geurts 1999; Kamp et al. 2011). In general cases, however, the antecedents of presuppositions need to be *inferred* using the assumptions that are not explicitly established in a previous discourse. The presupposition-as-type-inference view formulated within our proof-theoretic framework correctly captures this essentially inferential character of presupposition resolution.

4.5 Factive Presupposition

Factive presuppositions triggered by predicates like *know* and *regret* can also be handled using underspecified terms.²⁶ For instance, as the following set of examples shows, the factive predicate *know* presupposes that the embedded proposition is true.

²⁶See Tanaka et al. (2015) for more details. Earlier work using dependent type theory to analyze factivity includes Fox (1994b), Ranta (1994), and Krahmer and Piwek (1999).

- (60) a. John knows that Mary came. \Rightarrow Mary came.
 b. John does not know that Mary came. \Rightarrow Mary came.
 c. If John knows that Mary came, she will be surprised. \Rightarrow Mary came.

This fact can be captured by assuming that while a non-factive predicate like *believe* takes an entity and a proposition as argument, a factive predicate takes a proof term for the embedded proposition as an extra argument. We can read **believe**(x, P) as “the agent x believes the proposition P ”, and **know**(x, P, t) as “the agent x has evidence t of the proposition P ”. To capture the presuppositional inference, we use an underspecified term for the position t in **know**(x, P, t) which is to be filled by a proof term for P . Thus, the non-factive predicate *believe* and the factive predicate *know* have the following lexical entries:

- (61) *believe*; $(S \setminus NP)/S$; $\lambda p. \lambda x. \lambda c. \mathbf{believe}(x, pc)$
 (62) *know*; $(S \setminus NP)/S$; $\lambda p. \lambda x. \lambda c. \mathbf{know}(x, pc, @_1c : pc)$

The SR for the sentence (60a) is derived as in (63).

(63)

$$\begin{array}{c}
 \begin{array}{c} \text{John} \\ \hline NP \\ : j \end{array} \quad \begin{array}{c} \text{knows} \\ \hline (S \setminus NP)/S \\ : \lambda p. \lambda x. \lambda c. \mathbf{know}(x, pc, @_1c : pc) \end{array} \quad \begin{array}{c} \text{that} \\ \hline S/S \\ : \lambda p. p \end{array} \quad \begin{array}{c} \text{Mary came} \\ \hline S \\ : \lambda c. \mathbf{came}(m) \end{array} \quad > \\
 \hline
 \begin{array}{c} S \\ : \lambda c. \mathbf{know}(j, \mathbf{came}(m), @_1c : \mathbf{came}(m)) \end{array} <
 \end{array}$$

It is easily checked that the underspecified term $@_1$ has the type $\delta \rightarrow \mathbf{came}(m)$, where $\delta : \mathbf{type}$. This is the case even when the factive predicate appears in sentences like (60b) and (60c). Thus, in the same way as the examples in the previous sections, presuppositional inferences triggered by factive predicates can be derived as type inferences.

There are other important classes of presupposition triggers which cannot be discussed in this paper, including additive particles like *too* (Kripke 2009), cleft constructions (Atlas and Levinson 1981), and selection restrictions of predicates (Asher 2011; Magidor 2013). The framework of DTS is general enough to accommodate these cases as well. However, a detailed discussion has to be left for another occasion.

5 Conclusion

The dynamic setting of DTS, which consists of a context-passing mechanism and underspecified terms, solves the problem of proper formulation of anaphora resolution/presupposition binding and provides a compositional framework of discourse semantics based on dependent type theory.

As DTS is established as a semantic component of modern formal grammars due to the compositionality it attains, particularly (various kinds of) categorial grammars, the empirical coverage of DTS has been broadened to include linguistic phenomena such as generalized quantifiers (Tanaka et al. 2013; Tanaka 2014), modal subordination (Tanaka et al. 2014), conventional implicatures (or expressive content) (Bekki and McCready 2014), honorification in Japanese (Watanabe et al. 2014), and factive presuppositions (Tanaka et al. 2015).

References

- Ahn, R., & Kolb, H.-P. (1990). Discourse representation meets constructive mathematics. In L. Kalman & L. Polos (Eds.), *Papers from the Second Symposium on Logic and Language*. Akademiai Kiado.
- Asher, N. (2011). *Lexical Meaning in Context: A Web of Words*. Cambridge: Cambridge University Press.
- Atlas, J., & Levinson, S. (1981). It-clefts, informativeness and logical form: Radical pragmatics. In P. Cole (Ed.), *Radical Pragmatics* (pp. 1–61). Cambridge: Academic Press.
- Barendregt, H. P. (1992). Lambda calculi with types. In S. Abramsky, D. M. Gabbay, & T. Maibaum (Eds.), *Handbook of Logic in Computer Science* (Vol. 2, pp. 117–309). Oxford: Oxford Science Publications.
- Beaver, D. I. (2001). *Presupposition and Assertion in Dynamic Semantics*. Studies in Logic, Language and Information. Stanford: CSLI Publications & FoLLI.
- Bekki, D. (2013). Dependent type semantics: an introduction. In *the 2012 Edition of the LIRA Yearbook: A Selection of Papers*. Amsterdam: University of Amsterdam.
- Bekki, D. (2014). Representing anaphora with dependent types. In N. Asher & S. V. Soloviev (Eds.), *Proceedings of the Logical Aspects of Computational Linguistics (8th International Conference, LACL2014, Toulouse, France, June 2014), LNCS* (Vol. 8535, pp. 14–29). Springer, Heiderburg.
- Bekki, D., & McCready, E. (2014). CI via DTS. In *Proceedings of LENLS11* (pp. 110–123). Tokyo.
- Bekki, D., & Sato, M. (2015). Calculating projections via type checking. In *The Proceedings of Type Theory and Lexical Semantics (TYTLES), ESSLLI2015 Workshop*. Barcelona, Spain.
- Bos, J. (2003). Implementing the binding and accommodation theory for anaphora resolution and presupposition projection. *Computational Linguistics*, 29(2), 179–210.
- Chatzikyriakidis, S., & Luo, Z. (2014). Natural language inference in Coq. *Journal of Logic, Language and Information*, 23(4), 441–480.
- Chatzikyriakidis, S., & Luo, Z. (2016). On the Interpretation of Common Nouns: Types v.s. Predicates. In *Modern Perspectives in Type Theoretical Semantics, Studies of Linguistics and Philosophy*. Heidelberg: Springer.
- Clark, H. H. (1975). Bridging. In S. Roger, & B. L. Nash-Webber (Eds.), *In the Proceedings of TINLAP'75: Proceedings of the 1975 Workshop on Theoretical Issues in Natural Language Processing* (pp. 169–174). Cambridge, Massachusetts. (Association for Computational Linguistics, Stroudsburg, PA, USA).
- Cooper, R. (1983). *Quantification and Syntactic Theory*. Dordrecht: Reidel.

- Cooper, R. (2005). Austinian truth, attitudes and type theory. *Research on Language and Computation*, 3, 333–362.
- Coquand, T. (1986). An analysis of Girard's paradox. In *The Proceedings of the First Symposium on Logic in Computer Science* (pp. 227–236). IEEE Computer Society: Washington, D.C.
- Coquand, T., & Huet, G. (1988). The calculus of constructions. *Information and Computation*, 76(2–3), 95–120.
- Dávila-Pérez, R. (1994). Translating English into Martin-Löf's Theory of Types: A Compositional Approach, Technical report, University of Essex.
- Dávila-Pérez, R. (1995). Semantics and Parsing in Intuitionistic Categorical Grammar", Ph.d. thesis, University of Essex.
- Dummett, M. (1975). What is a theory of meaning? In S. Guttenplan (Ed.), *Mind and Language* (pp. 97–138). Oxford: Oxford University Press.
- Dummett, M. (1976). What is a theory of meaning? (II). In Evans & McDowell (Eds.), *Truth and Meaning* (pp. 67–137). Oxford: Oxford University Press.
- Evans, G. (1980). Pronouns. *Linguistic Inquiry*, 11, 337–362.
- Fox, C. (1994a). Discourse representation, type theory and property theory. In H. Bunt, R. Muskens & G. Rentier (Eds.), *The Proceedings of the International Workshop on Computational Semantics* (pp. 71–80). Tilburg: Institute for Language Technology and Artificial Intelligence (ITK).
- Fox, C. (1994b). Existence presuppositions and category mistakes. *Acta Linguistica Hungarica*, 42(3/4), 325–339. (Published 1996).
- Francez, N., & Dyckhoff, R. (2010). Proof-theoretic semantics for a natural language fragment. *Linguistics and Philosophy*, 33(6), 447–477.
- Francez, N., Dyckhoff, R., & Ben-Avi, G. (2010). Proof-theoretic semantics for subsentential phrases. *Studia Logica*, 94(3), 381–401.
- Geach, P. (1962). *Reference and Generality: An Examination of Some Medieval and Modern Theories*. Ithaca, New York: Cornell University Press.
- Gentzen, G. (1935). Untersuchungen über das logische Schliessen I,II. *Mathematische Zeitschrift* 39, pp. 176–210, 405–431. (Translated as 'Investigations into Logical Deduction', and printed in M.E. Szabo, *The Collected Works of Gerhard Gentzen*, Amsterdam: North-Holland, 1969, pp. 68–131).
- Geurts, B. (1999). *Presuppositions and Pronouns*. Oxford: Elsevier.
- Girard, J.-Y. (1972). *Interprétation fonctionnelle et élimination des coupures de l'arithmétique d'ordre supérieur*. Thèse de doctorat d'état: Université Paris VII.
- Groenendijk, J., & Stokhof, M. (1991). Dynamic predicate logic. *Linguistics and Philosophy*, 14, 39–100.
- Heim, I. (1982). The Semantics of Definite and Indefinite Noun Phrases, Ph.d dissertation, University of Massachusetts. Published 1989 by Garland Press, New York.
- Heim, I., & Kratzer, A. (1998). *Semantics in Generative Grammar*. Malden: Blackwell Publishers.
- Hook, J. G., & Howe, D. J. (1986). *Impredicative Strong Existential Equivalent to Type:Type*, Technical Report TR 86–760. Department of Computer Science, Cornell University.
- Kamp, H. (1981). A theory of truth and semantic representation. In J. Groenendijk, T. M. Janssen & M. Stokhof (eds.), *Formal Methods in the Study of Language*. Amsterdam: Mathematical Centre Tract 135.
- Kamp, H., J. van Genabith, & U. Reyle. (2011). Discourse representation theory. In D. M. Gabbay & F. Gunthner (Eds.), *Handbook of Philosophical Logic* (Vol. 15, pp.125–394). Dordrecht, Springer.
- Karttunen, L. (1976). Discourse referents. In J. D. McCawley (Ed.), *Syntax and Semantics 7: Notes from the Linguistic Underground* (Vol. 7, pp. 363–385). New York: Academic Press.
- Krahmer, E., & Piwek, P. (1999). Presupposition projection as proof construction. In H. Bunt & R. Muskens (Eds.), *Computing Meanings: Current Issues in Computational Semantics*, *Studies in Linguistics Philosophy Series*. Dordrecht: Kluwer Academic Publishers.
- Kripke, S. (2009). Presupposition and anaphora: remarks on the formulation of the projection problem. *Linguistic Inquiry*, 40(3), 367–386.

- Levesque, H. J. (2011). The winograd schema challenge. In *The Proceedings of AAAI Spring Symposium: Logical Formalization of Commonsense Reasoning*.
- Luo, Z. (2012a). Common nouns as types. In D. Béchet & A. Dikovsky (Eds.), *Proceedings of the Logical Aspects of Computational Linguistics, 7th International Conference, LACL2012, Nantes, France, July 2012* (pp. 173–185). Heidelberg: Springer.
- Luo, Z. (2012b). Formal semantics in modern type theories with coercive subtyping. *Linguistics and Philosophy*, 35(6), 491–513.
- Luo, Z. (2014). Formal semantics in modern type theories: is it model-theoretic, proof-theoretic, or both? In N. Asher & S. V. Soloviev (Eds.), *Logical Aspects of Computational Linguistics (8th International Conference, LACL2014, Toulouse, France, June 2014 Proceedings)*, LNCS 8535 (pp. 177–188). Toulouse: Springer.
- Magidor, O. (2013). *Category Mistakes*. Oxford: Oxford University Press.
- Martin-Löf, P. (1984). *Intuitionistic Type Theory*, G. Sambin (Ed.). Naples, Italy: Bibliopolis.
- Mikkelsen, L. (2011). Copular clauses. In *Semantics: An International Handbook of Natural Language Meaning*, HSK 33.2 (pp. 1805–1829). Berlin: de Gruyter.
- Mineshima, K. (2008). A presuppositional analysis of definite descriptions in proof theory. In: K. Satoh, A. Inokuchi, K. Nagao & T. Kawamura (Eds.), *New Frontiers in Artificial Intelligence: JSAI 2007 Conference and Workshops, Revised Selected Papers, Lecture Notes in Computer Science* (Vol. 4914, pp. 214–227). Heidelberg: Springer.
- Mineshima, K. (2013). Aspects of Inference in Natural Language, Ph.d. thesis, Keio University.
- Montague, R. (1974). *Formal Philosophy*. New Haven: Yale University Press.
- Nordström, B., Petersson, K., & Smith, J. (1990). *Programming in Martin-Löf's Type Theory*. Oxford: Oxford University Press.
- Piwek, P., & Krahmer, E. (2000). Presuppositions in context: constructing bridges. In P. Bonzon, M. Cavalcanti, & R. Nossium (Eds.), *Formal Aspects of Context, Applied Logic Series*. Dordrecht: Kluwer Academic Publishers.
- Prawitz, D. (1980). Intuitionistic Logic: A Philosophical Challenge. In G. von Wright (Ed.), *Logics and Philosophy*. The Hague: Martinus Nijhoff.
- Ranta, A. (1994). *Type-Theoretical Grammar*. Oxford: Oxford University Press.
- Russell, B. (1919). *Introduction to Mathematical Philosophy*. Crows Nest: George Allen & Unwin.
- Soames, S. (1989). Presupposition. In D. Gabbay & F. Guenther (Eds.), *Handbook of Philosophical Logic* (Vol. 4, pp. 553–616). Dordrecht: Reidel.
- Steedman, M. J. (1996). *Surface Structure and Interpretation*. Cambridge: The MIT Press.
- Sudo, Y. (2012). On the semantics of Phi features on pronouns, Doctoral dissertation, MIT.
- Sundholm, G. (1986). Proof theory and meaning. In D. Gabbay & F. Guenther (Eds.), *Handbook of Philosophical Logic* (Vol. III, pp. 471–506). Reidel: Kluwer.
- Sundholm, G. (1989). Constructive generalized quantifiers. *Synthese*, 79, 1–12.
- Tanaka, R. (2014). A proof-theoretic approach to generalized quantifiers in dependent type semantics. In R. de Haan (Ed.), *The Proceedings of the ESSLLI 2014 Student Session, 26th European Summer School in Logic, Language and Information* (pp. 140–151). Tübingen, Germany.
- Tanaka, R., Mineshima, K., & Bekki, D. (2014). Resolving modal anaphora in dependent type semantics. In *The Proceedings of the Eleventh International Workshop on Logic and Engineering of Natural Language Semantics (LENLS11), JSAI International Symposium on AI 2014* (pp. 43–56). Tokyo.
- Tanaka, R., Mineshima, K., Bekki, D. (2015). Factivity and presupposition in dependent type semantics. In *The Proceedings of Type Theory and Lexical Semantics (TYTTLES), ESSLLI2015 Workshop*.
- Tanaka, R., Nakano, Y., & Bekki, D. (2013). Constructive generalized quantifiers revisited. In *The Proceedings of Logic and Engineering of Natural Language Semantics 10 (LENLS 10)* (pp. 69–78). Tokyo.
- van der Sandt, R. (1992). Presupposition projection as anaphora resolution. *Journal of Semantics*, 9, 333–377.

- van der Sandt, R., & Geurts, B. (1991). Presupposition, anaphora, and lexical content. In O. Herzog & C.-R. Rollinger (Eds.), *Text Understanding in LILOG* (pp. 259–296). Berlin: Springer.
- Watanabe, N., McCready, E., & Bekki, D. (2014). Japanese honorification: compositionality and expressivity. In S. Kawahara & M. Igarashi (Eds.), *The Proceedings of FAJL 7: Formal Approaches to Japanese Linguistics, the MIT Working Papers in Linguistics 73* (pp. 265–276). International Christian University, Japan.

Modern Perspectives in Type-Theoretical Semantics

Chatzikyriakidis, S.; Luo, Z. (Eds.)

2017, VIII, 296 p. 49 illus., Hardcover

ISBN: 978-3-319-50420-9