A propositions-as-types approach to the generalized crossover effect[1]

Daiki MATSUOKA — *The University of Tokyo*
Daisuke BEKKI — *Ochanomizu University*
Hitomi YANAKA — *The University of Tokyo*

Abstract. It has recently been suggested that the crossover effect, standardly understood as a constraint on anaphoric dependencies between quantifiers and pronouns, extends to presupposition projection. Although this observation calls for a unified account of how presupposition can interact with quantifier scope and anaphora, existing theories fail to provide one. We address this challenge by adopting Dependent Type Semantics, a type-theoretical framework that represents propositions as types. In this approach, the interactions among quantifier scope, anaphora, and presupposition are analyzed through a process called type checking, based on which the generalized version of the crossover effect can be uniformly derived.

Keywords: crossover, anaphora, presupposition, Dependent Type Semantics, continuation-based grammar.

1. Introduction

Anaphoric dependencies between quantifiers and pronouns are known to be disallowed in certain syntactic configurations. This effect, called *crossover* (CO) (Postal, 1971), can be illustrated by the contrast between (1a) and (1b). In (1b), *her* cannot co-vary with *every girl* as in (1a), even though the universal quantifier in the object position can, in principle, take inverse scope over the subject.[2]

(1) a. Every girl$_i$ praised her$_i$ mother.
b. *Her$_i$ mother praised every girl$_i$.

Traditionally, theories of CO have relied on the structural relation of c-command (Reinhart, 1983), which can naturally account for the asymmetry in (1): the universal quantifier (or its trace, left behind after a scope-shifting operation like Quantifier Raising (QR) (May, 1977)) does not c-command the pronoun in (1b), whereas it does in (1a).

However, subsequent research has revealed diverse syntactic configurations that exhibit constraints reminiscent of classic CO cases despite the lack of c-command between the quantifier and the pronoun (Ruys, 2000; Büring, 2004; Barker, 2012). For example, the quantificational possessor in (2a) can bind *her* even though it fails to c-command the pronoun, and yet (2b) indicates that this type of pronominal binding shows the same asymmetry as (1). A similar observation applies to donkey anaphora, as demonstrated in (3).

(2) a. Every girl$_i$'s friend praised her$_i$ mother.
b. *Her$_i$ mother praised every girl$_i$'s friend.

[1]We would like to thank the anonymous reviewers and the participants of Sinn und Bedeutung 29 for their helpful feedback. This work was supported by JSPS KAKENHI Grant Number JP24H00809 and JST PRESTO Grant Number JPMJPR21C8.

[2]The configuration in (1b) is often referred to as *weak* crossover, distinguished from *strong* crossover. For reasons of space, we do not discuss the difference between them here.

(3) a. Every farmer who owns a donkey$_i$ admires its$_i$ strength.
b. *Its$_i$ strength helps every farmer who owns a donkey$_i$.

While these considerations have motivated more sophisticated accounts of CO (see, e.g., Safir (2017) for a review), Elliott and Sudo (2021) challenged them by pointing out that the empirical domain of CO can be further extended to include *presupposition projection*. They argued for a unified analysis of this general version of CO (*generalized crossover*; GCO), but only showed its intractability for existing theories, leaving the unification itself as an open problem.

We address this gap by using a framework called *Dependent Type Semantics* (DTS) (Bekki and Mineshima, 2017; Bekki, 2023). DTS uses types to represent propositions, allowing them to scopally interact with other elements. It also introduces underspecified representations for anaphora and presupposition, which are resolved with contextual information through a process called type checking. As we will show, this framework enables us to unify the interplay of pronouns, quantifiers, and presuppositions, thereby providing a unified account of GCO.

The rest of this paper is organized as follows. First, we review the evidence for GCO and its theoretical implications in Section 2. Then, we describe the framework of DTS in Section 3, based on which we present our proposal in Section 4. After discussing some possible extensions in Section 5, we conclude with some open issues in Section 6.

2. Generalized crossover

2.1. Empirical observations

As a starting point, we define the CO effect as in (4).

(4) Quantifier scope cannot feed an anaphoric dependency if the pronoun precedes the quantifier at a certain abstract level.[3]

GCO broadens the range of this statement in two respects: (i) generalizing "quantifier scope" to include projective content[4] and (ii) generalizing "anaphoric dependency" to include presupposition satisfaction.

We begin with the discussion on the second point, which is more straightforward. Considering its similarities with anaphoric binding (van der Sandt, 1992), it would be natural to expect that presupposition satisfaction is sensitive to the CO configuration, which is indeed the case. For example, Ruys (2000) pointed out a CO-like constraint with *another*, which triggers an additive presupposition. In (5a), the presupposition can be filtered by *every student*, which leads to the reading that each student talked to a student other than themselves. In contrast, (5b) lacks such a reading, indicating that the satisfaction of the additive presupposition is subject to the same constraint as regular CO cases such as (1).

[3]The "abstract level" can vary depending on the choice of theory. For example, Barker and Shan (2014) employed the notion of evaluation order, which is distinguished from the surface linear order. Chierchia (2020) basically utilized the linear order at the level of Logical Form.

[4]In addition to presuppositions, Elliott and Sudo (2021) also discussed the GCO effect with conventional implicatures (Potts, 2005), which constitute another class of projective content. Although we do not delve deeper into this point, we note that our account presented in Section 4 is compatible with a previous DTS-based analysis of conventional implicatures (Matsuoka et al., 2024).

(5) a. Every student talked to **another** student.
b. **Another** student talked to every student.

Elliott and Sudo (2021) observed the same pattern of quantifier-presupposition interaction for other kinds of triggers (e.g., factive predicates), concluding that CO applies to presupposition satisfaction as well as to anaphoric dependencies.

Next, we turn to cases where presupposition feeds anaphora. In (6a), for instance, the factive presupposition associated with *know*, projecting out of the scope of negation, provides an antecedent for the pronoun *it* in the second conjunct. We can confirm that presupposition projection is crucial here by replacing *know* with a non-factive predicate like *believe*, as in (6b), in which case anaphora is indeed disallowed.

(6) a. Alex did not **know** that Kim wrote a paper$_i$, and reviewed it$_i$.
b. *Alex did not **believe** that Kim wrote a paper$_i$, and reviewed it$_i$.

This presupposition-pronoun interaction is also subject to a constraint reminiscent of CO. We illustrate the point with the pair in (7). Crucially, pronominal anaphora is blocked in (7b), where the pronoun precedes the trigger.

(7) a. A researcher who did not **know** that Kim wrote a paper$_i$ reviewed it$_i$.
b. *Its$_i$ reviewer did not **know** that Kim wrote a paper$_i$.

The unavailability of anaphora in cases like (7b) is surprising given the common assumption that presuppositions impose a *precondition* on the utterance context (Heim, 1983; Beaver, 2001). In other words, the factive presupposition in (7b) is standardly regarded as part of the common ground, as is intuitively depicted in (8). This means that the presupposed content is supposed to be introduced *before* the at-issue content, which should allow *its* to refer to *a paper* in (7b). Thus, we face an issue similar to the one with CO: we need to account for why such configurations block an anaphoric dependency that should be possible in principle.

(8) [Context: Kim wrote a paper$_i$. (let this proposition be p)]
Its$_i$ reviewer did not know p.

At this point, one might suspect that the unacceptability of (7b) is due to the general dispreference for cataphora with indefinites. Elliott and Sudo (2021) argued against this idea with (9), reporting that the degradation effect with cataphora is much milder than in (7b).

(9) Her$_i$ boss needs to do her$_i$ work, if an independent contractor$_i$ is on sick leave.

Based on these observations, the GCO effect can be defined as follows.

(10) Quantifier scope (resp. projective content) cannot feed a semantic dependency if the semantically dependent expression precedes the quantifier (resp. the trigger of the projective content) at a certain abstract level.

2.2. Theoretical challenges

We next discuss the implications of the above generalization (see Elliott and Sudo (2021) for further details). As to interactions between quantifiers and presuppositions (e.g., (5)), they are challenging for theories that depend on indices for pronominal binding, since it is unclear how referential indices are involved in presupposition satisfaction. Although this issue may be circumvented by expanding a theory of presupposition that employs co-indexation with context variables (Schlenker, 2011), accounts of this kind have not been developed yet.

A more pressing problem arises with presupposition-pronoun interactions (e.g., (7)). The difficulty stems from the discrepancy between syntactic movement and presupposition projection. More concretely, the mainstream accounts of CO hinge on the assumption that pronominal binding is only possible from a c-commanding A-position (Reinhart, 1983). As for (1b), QR moves *every girl* to an A'-position (see (11)), so this assumption correctly blocks binding.

(11) [every girl$_i$] [her mother loves t_i]

This type of movement-based analysis of CO does not seem extendable to GCO, as presupposition projection is not conventionally considered to be driven by a syntactic movement. One crucial reason for this is that, whereas QR generally obeys the island constraints like other kinds of A'-movement, presupposition projection does not. For instance, quantifiers basically cannot take scope over the conditional antecedent (12a), but presuppositions can project out of it (12b).[5]

(12) a. If [every student comes], the professor will be surprised. (*$\forall$ > if)
b. If [Alex **knows** that Kim wrote a paper], she will have read it by now.

Therefore, existing theories of CO would have to posit two separate constraints to account for the GCO effect: one for quantificational binding, and another for binding with presuppositions. This situation, although not inconsistent, is undesirable given the parallels that motivated the generalization in Section 2.1. In other words, we need a structural relation that governs quantifiers and presuppositions in a unified way and realizes GCO as a single constraint. In the following, we pursue this desideratum with the propositions-as-types approach.

3. Dependent Type Semantics

3.1. The propositions-as-types principle

DTS follows the tradition of the type-theoretical approach to natural language semantics (Sundholm, 1986; Ranta, 1995; Luo, 2012), which uses *types* as semantic representations. Since its core ideas are crucial to our proposal, we spell out the details here.

First, we explain why type theory can be used to represent the meaning of a sentence. The motivation derives from the systematic conformity between logic and type theory called the *Curry-Howard correspondence* (Howard, 1980). As a guiding intuition, it is helpful to view a

[5]Even if one chooses to adopt a non-standard scope-shifting mechanism that can ignore some scope islands (e.g., Barker (2022)), one will then need to give a systematic account of why projected presuppositions are not at-issue (Simons et al., 2010), whereas shifted quantifier scope contributes to the at-issue content.

proof of a proposition as a *program* (i.e., a *term* in lambda calculus). Suppose, for instance, we have constructed a proof of $A \supset B$. Then, the proof gives us a method to turn any proof of A into a proof of B. Namely, a proof of $A \supset B$ amounts to a *function* from a proof of A to a proof of B. Likewise, a proof of the conjunction $A \wedge B$ can be regarded as a *pair* of proofs of A and B.

For a more precise discussion, we introduce some auxiliary notions. In type theory, a *typing judgment* $\Gamma \vdash M : A$ indicates that the term M has type A given the list of variable declarations $\Gamma \equiv x_1 : A_1, \ldots, x_n : A_n$ (called a *typing environment*). As with proof theory, there are some rules prescribing what typing judgments are derivable. To illustrate, we show two such rules below ($A \to B$ is a *function type*, and $A \times B$ a *product type*).

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}\,(\to I) \qquad \frac{\Gamma \vdash t : A \times B}{\Gamma \vdash \pi_1 t : A}\,(\times E_1)$$

$(\to I)$ forms a function via lambda abstraction. Recalling that a proof of $A \supset B$ is essentially a function, we can see that this rule is equivalent to "discharging" the premise (i.e., inferring $\Gamma \vdash A \supset B$ from $\Gamma, A \vdash B$). $(\times E_1)$ takes out the first element a of a pair $\langle a, b \rangle$ with π_1, and it is parallel to the inference rule drawing A from $A \wedge B$. Given this correspondence between type-theoretic and proof-theoretic rules, a typing judgment $\Gamma \vdash M : A$ can be interpreted as "M is a proof of the proposition A under the premises Γ." We can summarize this point with the following principle.

(13) The propositions-as-types principle
A proposition and its proofs can be identified with a type and its terms, respectively.

So far, we have considered only propositional connectives. To handle quantification, we need to enrich the type theory with *dependent types* (Martin-Löf, 1984). A dependent type is a type that depends on the value of another type, which can represent a proposition with a free variable (e.g., $\mathtt{girl}(x)$ for "x is a girl" with x being an entity). Those variables can be universally bound with the Π-*type* $(x : A) \to B$, a generalized version of the function type. For example, *Kim praised every girl* can be translated into the following type.

(14) $(x : \mathtt{e}) \to ((u : \mathtt{girl}(x)) \to \mathtt{praise}(\mathtt{k}, x))$

(14) is similar to the first-order logic formula $\forall x.(\mathtt{girl}(x) \supset \mathtt{praise}(\mathtt{k}, x))$. Note, however, that the Π-type plays a double role here. On the one hand, $(x : \mathtt{e}) \to \cdots$ amounts to the universal quantification over the domain of entities. On the other hand, and more importantly, the conditional part is also represented by a Π-type $(u : \mathtt{girl}(x)) \to \cdots$. This parallel treatment is possible because the conditional $A \supset B$ can be viewed as a universal quantification over the proofs of A ("for any proof x of A, ..."). The same relationship holds between $\exists$ and $\wedge$. Corresponding to the existential quantification, we have the Σ-*type* $(x : A) \times B$, a generalized version of the product type. For example, (15) serves as a representation for *Kim wrote a paper*, where the Σ-type is used for both $\exists$ and $\wedge$.

(15) $(x : \mathtt{e}) \times ((u : \mathtt{paper}(x)) \times \mathtt{write}(\mathtt{k}, x))$

This mechanism of "quantification over proofs" can capture some kinds of anaphoric dependencies that go beyond the expressive power of the standard predicate logic (Sundholm, 1986;

Ranta, 1995). Consider, for example, the inter-sentential anaphora in (16a). With dependent types, we can represent the meaning as (16b) (for readability, we will often use the square-bracket notation $[\cdots]$ for types of the form $(\cdots) \times \cdots$).

(16) a. Kim wrote a paper$_i$, and Alex reviewed it$_i$.

b. $$\begin{bmatrix} v : \begin{bmatrix} x : \mathtt{e} \\ \begin{bmatrix} u : \mathtt{paper}(x) \\ \mathtt{write}(\mathtt{k}, x) \end{bmatrix} \end{bmatrix} \\ \mathtt{review}(\mathtt{a}, \pi_1 v) \end{bmatrix}$$

Here, the representations for the two conjuncts are combined with a Σ-type. The variable v is existentially quantified over the proofs of $(x : \mathtt{e}) \times \cdots$ (= (15)). Those proofs have the form $\langle x, \langle p_1, p_2 \rangle \rangle$, where x is an entity and p_1 (resp. p_2) is a proof of "x is a paper" (resp. "Kim wrote x"). Hence, the term $\pi_1 v$ in the second conjunct refers to the (existentially quantified) entity x. As a result, we can correctly describe the co-construal between *a paper* and *it*, even though the scope of $x : \mathtt{e}$ is confined to the first conjunct. Note well that this prediction hinges on the assumption that the variable v ranges over the proofs of the first conjunct (*Kim wrote a paper*). Crucially, this treatment is possible because we represent propositions as types.

3.2. Underspecification and type checking

Although dependent type theory thus provides us with a powerful tool for representing meanings, we still need to specify how semantic representations are systematically derived: for instance, it is far from plausible to stipulate that *it* is lexically encoded as $\pi_1 v$, considering that its referent is context-dependent. To address this, DTS introduces a special type named the *underspecified type* $(x @ A) \times B$ (@-type, for short). It works as an intermediate representation, with x being a placeholder to be filled with a concrete term of A. We illustrate its usage in (17).

(17) Alex reviewed it. $\rightsquigarrow$ $(y @ \mathtt{e}) \times \mathtt{review}(\mathtt{a}, y)$

DTS handles presupposition triggers in the same way, in line with the *presupposition-as-anaphora* paradigm of van der Sandt (1992). (18) is an example of the factive presupposition with *know*, where we abbreviate (15) as A_k. Here, the @-type can be understood as anaphoric to a proof of A_k, thereby requiring the presupposition to be true in the context.

(18) Alex did not know that Kim wrote a paper. $\rightsquigarrow$ $\neg((v @ A_k) \times \mathtt{know}(\mathtt{a}, A_k))$

Those underspecified parts are resolved via *type checking*, a process that checks whether a given typing judgment is derivable. Formally, it can be defined as a function TC from a typing judgment (which may or may not be derivable) to a set of derivable ones. If no @-type is involved, TC is deterministic, with the result being a singleton (success) or an empty set (failure).

$$\mathsf{TC}(\Gamma \vdash M : A) = \begin{cases} \{\Gamma \vdash M : A\} & \text{(if } \Gamma \vdash M : A \text{ is derivable)} \\ \varnothing & \text{(otherwise)} \end{cases}$$

The @-type comes into play when we check a judgment of the form $\Gamma \vdash (x @ A) \times B : \mathsf{type}$, which means that $(x @ A) \times B$ is well-formed as a type in Γ. This computation requires a process called *proof search*, which looks for a term for the variable of the @-type. As a toy example, consider the type checking of $(z \,@\, \mathtt{e}) \times Pz$ in $\Gamma \equiv x : \mathtt{e},\ y : \mathtt{e}$ (i.e., a situation where a pronoun has two possible referents). Here, the proof search returns the terms that can fill the blank in $\Gamma \vdash \_\_ : \mathtt{e}$ (in this case, x and y). Substituting them for z, we obtain the following result.

$$\mathsf{TC}(\Gamma \vdash (z \,@\, \mathtt{e}) \times Pz : \mathsf{type}) = \{\Gamma \vdash Px : \mathsf{type},\ \Gamma \vdash Py : \mathsf{type}\}$$

Now, we are ready to describe how the inter-sentential anaphora in (16a) is predicted. Its underspecified representation is (19).

(19) $(v : A_k) \times ((y \,@\, \mathtt{e}) \times \mathtt{review}(\mathtt{a}, y))$

To resolve the anaphoric dependency, we check the well-formedness of the whole representation, inspecting each part one by one. The process goes from higher scope to lower. We first compute $\mathsf{TC}(\vdash A_k : \mathsf{type})$, which is straightforward. Then, we check the well-formedness of the second conjunct. At this point, we temporarily add the first conjunct to the typing environment, so the required type checking is as follows.

$$\mathsf{TC}(v : A_k \vdash (y \,@\, \mathtt{e}) \times \mathtt{review}(\mathtt{a}, y) : \mathsf{type})$$

Hence, we can use the information of the first conjunct in the proof search for $y \,@\, \mathtt{e}$, obtaining $\pi_1 v$ as a result. In this way, we can obtain the intended representation (16b) (this process can be visually described as in Figure 1).

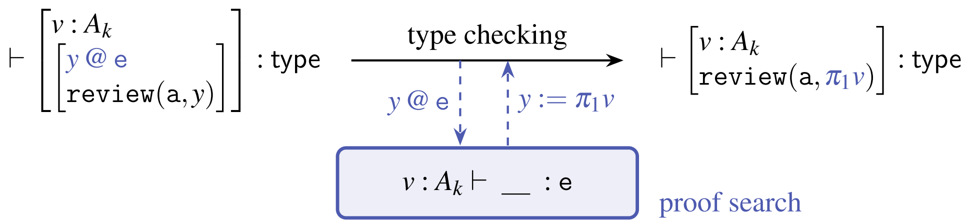

Figure 1. Visualization of the type checking of (19).

3.3. Deriving CO with the type checking order

At this point, we emphasize that the process of type checking is, in a sense, "dynamic." As we have just explained, the type checking of $(x : A) \times B$ and $(x : A) \to B$ proceeds from A to B, from higher scope to lower, so that the well-formedness of B may depend on $x : A$. As a result, type checking reflects the order-sensitive nature of the interpretation process. To elaborate on the idea, we demonstrate that anaphora resolution in DTS shows asymmetry with respect to the order of type checking. Take (20) as an example (note that indices are used only for descriptive purposes, without doing any work in the theory).

(20) Kim wrote it$_{\{*i/j\}}$, and Alex reviewed a paper$_i$. $\rightsquigarrow \begin{bmatrix} u : \begin{bmatrix} y \,@\, \mathtt{e} \\ \mathtt{write}(\mathtt{k}, y) \end{bmatrix} \\ \begin{bmatrix} x : \mathtt{e} \\ [\cdots] \end{bmatrix} \end{bmatrix}$

Crucially, when the proof search for $y\,@\,\mathtt{e}$ is triggered, the latter half $(x:\mathtt{e})\times\cdots$ is not available in the typing environment. This means that there is no way for $y\,@\,\mathtt{e}$ to be resolved with a term of the Σ-type for *a paper*, unlike in (19). Thus, we can correctly predict that *she* cannot co-vary with *a girl* in (20).[6]

Bekki (2023) suggested that this constraint on anaphora resolution in DTS can capture the CO effect. In the proposed account, the subject-object asymmetry observed in the classic CO case (1) is analyzed in terms of the scopal asymmetry shown in (21): $y\,@\,\mathtt{e}$ can be resolved by $x:\mathtt{e}$ in (21a), but not in (21b), where the @-type is type checked before the Π-type.

(21) a. Every girl praised her mother. $\rightsquigarrow (x:\mathtt{e})\to\left((u:\mathtt{girl}(x))\to\begin{bmatrix}y\,@\,\mathtt{e}\\ \mathtt{praise}(x,\mathtt{mother}(y))\end{bmatrix}\right)$

b. Her mother praised every girl. $\rightsquigarrow \begin{bmatrix}y\,@\,\mathtt{e}\\ (x:\mathtt{e})\to((u:\mathtt{girl}(x))\to\mathtt{praise}(\mathtt{mother}(y),x))\end{bmatrix}$

This DTS-based account is promising for providing a uniform analysis of GCO, considering that the @-type enables uniform handling of anaphora resolution and presupposition satisfaction. More specifically, since the @-type can represent the meaning of presupposition triggers as well as pronouns, the quantifier-presupposition interactions (e.g., (5)) can be derived in a way parallel with (21).

However, two issues remain to be addressed. The first problem concerns the scope of @-types. Bekki (2023) treated possessives as a generalized quantifier, as shown in (22). This means that, without any restriction on the syntax-semantics interface, we would predict that (21b) may have the "inverse scope" reading ($\Sigma > @$), as a result of which the semantic representation would have the same scopal relation as (21a), wrongly allowing pronominal anaphora.

(22) her mother $\rightsquigarrow \lambda p.(y\,@\,\mathtt{e})\times p(\mathtt{mother}(y))$

Even if this problem is set aside, we face a more serious issue related to presupposition accommodation (Lewis, 1979). In Bekki (2023), accommodation is defined as an operation that (i) extends the context with the presupposed content and (ii) re-runs the whole type checking process. Critically, the second point prevents us from predicting the GCO effect. To see why, we repeat the GCO-violating case below.

(7b) *Its reviewer$_i$ did not know that Kim wrote a paper$_i$.

Under the current definition of accommodation, the type checking is re-run with the (accommodated) factive presupposition in the context. Then, the proof search for *its* picks out *a paper* as its antecedent, which wrongly predicts that anaphora is possible. This means that we need to make the operation of accommodation sensitive to the type checking order. In the next section, we present a solution to each of these two issues.

[6]The type checking order is a necessary but not sufficient condition for DTS to predict anaphoric dependencies. Like standard dynamic semantic frameworks (e.g., Groenendijk and Stokhof (1991)), DTS is equipped with certain conditions on anaphora accessibility, as will be explained in Section 5.1.

4. Proposal

Our proposal is twofold. First, we build a syntax-semantics interface that properly handles the scope of @-types. For this purpose, we adopt *continuation-based grammar* (Shan and Barker, 2006; Barker and Shan, 2014), which provides a principled way to handle scope ambiguity without relying on the notion of c-command. Second, we revise the definition of accommodation in DTS. Our core idea is to trigger the process of accommodation *during* type checking, thereby making it sensitive to the type checking order.

4.1. Combining DTS with continuation-based grammar

We first explain how continuations can be used to handle quantifier scope. Intuitively, the continuation of an expression refers to the surrounding expression that it takes scope over. In (23), for example, the quantifier *every book* has *Kim read* [ ] *last week* as its continuation.

(23) Kim read every book last week.

Barker and Shan (2014) developed a categorial grammatical framework that treats a scope taker as a function that takes its continuation as an argument. In general, a scope-taking expression has category $C \mathbin{/\!\!/} (A \mathbin{\backslash\!\!\backslash} B)$, which is often written alternatively as $\frac{C \mid B}{A}$ for readability (*tower notation*). The category tells us that the expression locally works as an A, takes scope over a B, and forms a C after its continuation is given. Correspondingly, the semantic representation of a scope taker has the form $\lambda\kappa.f[\kappa x]$, with the continuation variable κ. In the tower notation, we simplify it by writing $\frac{f[\,]}{x}$. To illustrate, (24) shows the category and the semantic representation of *every book*.

(24) $$\text{every book} := \frac{\mathrm{S} \mid \mathrm{S}}{\mathrm{DP}} : \frac{\forall x.(\mathtt{book}(x) \supset [\,])}{x}$$

Two scope-taking expressions are composed via the *combination schema*. We show the function application rules below. In both cases, the bottom level is the same as an ordinary formulation of categorial grammar, with the slashes specifying the direction in which the argument B must appear. In contrast, the second level simply composes the two scopes g and h from left to right, independently from the direction of the function application. Notably, all the combination schema handles scopes according to the linear order, which will be a key to the account of the (G)CO effect.

$$\left(\begin{array}{cc} \frac{E \mid D}{A/B} & \frac{D \mid C}{B} \\ \text{exp1} & \text{exp2} \\ \frac{g[\,]}{f} & \frac{h[\,]}{a} \end{array}\right) = \begin{array}{c} \frac{E \mid C}{A} \\ \text{exp1 exp2} \\ \frac{g[h[\,]]}{fa} \end{array} \qquad \left(\begin{array}{cc} \frac{E \mid D}{B} & \frac{D \mid C}{B\backslash A} \\ \text{exp1} & \text{exp2} \\ \frac{g[\,]}{a} & \frac{h[\,]}{f} \end{array}\right) = \begin{array}{c} \frac{E \mid C}{A} \\ \text{exp1 exp2} \\ \frac{g[h[\,]]}{fa} \end{array}$$

In addition, we have a unary rule that "collapses" a tower, called the *lowering* rule. Its definition is shown below.[7] This rule is typically used when a whole clause is constructed and the scope

[7]Here are some remarks on the difference between our formulation and the one presented by Barker and Shan

taking effect needs to be closed off.

$$\begin{array}{c} \dfrac{B \mid A}{A} \\ \text{exp} \\ \dfrac{f[\,]}{a} \end{array} \xrightarrow{\text{LOWER}} \begin{array}{c} B \\ \text{exp} \\ f[a] \end{array}$$

To see how the system works, consider the sentence *a student read every book*. Below, we derive its surface scope reading (we have omitted the sub-derivations for the quantifiers). Note that the verb *read*, which itself is not a scope taker, is given a vacuous scope [ ] so that it can be combined with *every book*.[8]

$$\left(\begin{array}{c} \dfrac{\mathrm{S} \mid \mathrm{S}}{\mathrm{DP}} \\ \text{a student} \\ \dfrac{\exists x.(\mathtt{std}(x) \wedge [\,])}{x} \end{array} \left(\begin{array}{c} \dfrac{\mathrm{S} \mid \mathrm{S}}{(\mathrm{DP}\backslash\mathrm{S})/\mathrm{DP}} \\ \text{read} \\ \dfrac{[\,]}{\lambda w.\lambda z.\mathtt{read}(z,w)} \end{array} \begin{array}{c} \dfrac{\mathrm{S} \mid \mathrm{S}}{\mathrm{DP}} \\ \text{every book} \\ \dfrac{\forall y.(\mathtt{book}(y) \supset [\,])}{y} \end{array} \right) \right) = \begin{array}{c} \dfrac{\mathrm{S} \mid \mathrm{S}}{\mathrm{S}} \\ \text{a ... every ...} \\ \dfrac{\exists x.(\mathtt{std}(x) \wedge \forall y.(\mathtt{book}(y) \supset [\,]))}{\mathtt{read}(x,y)} \end{array}$$

$$\xrightarrow{\text{LOWER}} \begin{array}{c} \mathrm{S} \\ \text{a ... every ...} \\ \exists x.(\mathtt{std}(x) \wedge \forall y.(\mathtt{book}(y) \supset \mathtt{read}(x,y))) \end{array}$$

Due to the left-to-right nature of the combination schema, we cannot derive the inverse scope only with two-level towers. This motivates us to introduce a tower with more continuation levels, as exemplified in (25). As indicated, the scope of an expression can in principle belong to any continuation level.

(25) every book $:= \begin{array}{c} \mathrm{S} \mid \mathrm{S} \\ \hline \mathrm{S} \mid \mathrm{S} \\ \hline \mathrm{DP} \end{array} : \begin{array}{c} \forall x.(\mathtt{book}(x) \supset [\,]) \\ \hline [\,] \\ \hline x \end{array}$ or $\begin{array}{c} [\,] \\ \hline \forall x.(\mathtt{book}(x) \supset [\,]) \\ \hline x \end{array}$

Correspondingly, we generalize the syntactic rules to multiple-level towers so that the scope is composed left to right and a higher continuation level takes scope over a lower one (see Barker and Shan (2014) for details). Then, we can derive the inverse scope reading of *a student read every book* as follows. Here, $\exists y$ belongs to the first continuation level and $\forall x$, which is in a higher level, takes scope over $\exists y$.

(2014). In their definition, the category A suppressed by the lowering rule is restricted to S. This restriction is essential to their account of the CO effect, in which pronominal binding is resolved in the syntactic derivation with what they call the *binding* rule. Since DTS resolves semantic dependencies post-syntactically (i.e., at the level of semantic representations), we do not adopt the binding rule or introduce the restriction on the lowering rule.

[8]The type-shifting mechanism introducing a null continuation level can be formalized either as a lexical operation or as a syntactic rule (the difference does not bear on the current discussion).

$$\left(\begin{array}{c} \frac{\frac{S \mid S}{S \mid S}}{\mathrm{DP}} \\ \text{a student} \\ \frac{\frac{[\,]}{\exists y.(\mathtt{std}(y) \wedge [\,])}}{y} \end{array} \left(\begin{array}{c} \frac{\frac{S \mid S}{S \mid S}}{\mathrm{DP}\backslash(\mathrm{S/DP})} \\ \text{read} \\ \frac{\frac{[\,]}{[\,]}}{\lambda w.\lambda z.\mathtt{read}(z,w)} \end{array} \begin{array}{c} \frac{\frac{S \mid S}{S \mid S}}{\mathrm{DP}\backslash \mathrm{S}} \\ \text{every book} \\ \frac{\frac{\forall x.(\mathtt{book}(x) \supset [\,])}{[\,]}}{x} \end{array} \right) \right) = \begin{array}{c} \frac{\frac{S \mid S}{S \mid S}}{\mathrm{S}} \\ \text{a ... every ...} \\ \frac{\frac{\forall x.(\mathtt{book}(x) \supset [\,])}{\exists y.(\mathtt{std}(y) \wedge [\,])}}{\mathtt{read}(y,x)} \end{array}$$

$$\xrightarrow{\text{LOWER}} \begin{array}{c} \frac{S \mid S}{\mathrm{S}} \\ \text{a ... every ...} \\ \frac{\forall x.(\mathtt{book}(x) \supset \exists y.(\mathtt{std}(y) \wedge [\,]))}{\mathtt{read}(y,x)} \end{array} \xrightarrow{\text{LOWER}} \begin{array}{c} \mathrm{S} \\ \text{a ... every ...} \\ \forall x.(\mathtt{book}(x) \supset \exists y.(\mathtt{std}(y) \wedge \mathtt{read}(y,x))) \end{array}$$

In this way, continuation-based grammar derives scopal interactions without employing the c-command relation. One advantage of this formalism is that it can straightforwardly handle some instances of quantificational binding where the quantifier does not c-command the pronoun (e.g., possessor binding (2a)), which are challenging to the traditional c-command-based theories of CO, as mentioned in Section 1.

Having described the basic workings of continuation-based grammar, we turn to combining it with DTS. Concretely, we make one assumption about the @-type.

(26) Tower restriction
The scope of an @-type must be in the highest continuation level.

Essentially, (26) prohibits a pronoun/presupposition trigger from being subject to inverse scope. Consider the sentence *her mother praised every girl* again. As demonstrated by the following (failed) derivation, even if we place the universal quantifier in the third level (so that it can take inverse scope), we cannot complete the derivation due to the level mismatch. As a result, (26) forces *her* to take scope over *every girl*, thereby correctly blocking the co-construal.[9]

$$\left(\begin{array}{c} \frac{S \mid S}{\mathrm{DP}} \\ \text{her mother} \\ \frac{(y \,@\, \mathtt{e}) \times [\,]}{\mathtt{mother}(y)} \end{array} \begin{array}{c} \frac{\frac{S \mid S}{S \mid S}}{\mathrm{DP}\backslash \mathrm{S}} \\ \text{praised every girl} \\ \frac{\frac{(x : \mathtt{e}) \to ((u : \mathtt{girl}(x)) \to [\,])}{[\,]}}{\lambda z.\mathtt{praise}(z,x)} \end{array} \right) = \times$$

We remark that this kind of restriction on tower levels is (at least) justified in continuation-based grammar. It has been noted that some quantifiers (e.g., negative ones (27)) do not allow inverse scope (Mayr and Spector, 2010). To handle such cases, we anyway need to introduce stipulations about the possibility of inverse scope for certain expressions, of which (26) can be regarded as one case.

[9]In our approach, an individual who does accept a CO-violating anaphoric dependency can be considered as having access to a tower with an @-type situated in an intermediate continuation level.

(27) No student read every book. (no $>\forall$, $^*\forall >$ no)

4.2. Dynamic accommodation in DTS

We turn to our second task, the revision of the formalization of presupposition accommodation. As we pointed out at the end of Section 3.3, the previous DTS studies did not consider how accommodation interacts with the flow of anaphora resolution. To address this issue, we "dynamicize" the operation of accommodation in DTS, by positing that it is performed in the course of type checking. We present a formal definition in (28).

(28) Dynamic accommodation
If the proof search fails in computing $\mathsf{TC}(\Gamma \vdash (x\,@^{-}A)\times B : \mathsf{type})$, then we can replace the result of this type checking with that of $\mathsf{TC}(\Gamma, x : A \vdash B : \mathsf{type})$.

Here, the feature $\pm$ on the @-type indicates the *strong contextual felicity* constraint (Tonhauser et al., 2013): the $+$ value means that the projective content cannot be accommodated (in which case (28) does not apply). For example, the existence implication associated with a pronoun is subject to this constraint.

Let us check that (28) correctly predicts the projection behavior of presuppositions. Consider the factive presupposition with *know*, whose lexical entry is given in (29).

(29) $$\text{know} := \frac{\mathrm{S} \mid \mathrm{S}}{\mathrm{DP}\backslash\mathrm{S}} \Big/ \mathrm{S} \; : \; \lambda A. \frac{(u\,@^{-}A)\times[\,]}{\lambda x.\mathtt{know}(x, A)}$$

Then, the representation for (18) (*Alex did not know that* A_k) is type checked as shown in Figure 2. We can see that the factive presupposition A_k is added to the typing environment (i.e., extends the common ground), thereby projecting out of the scope of negation.

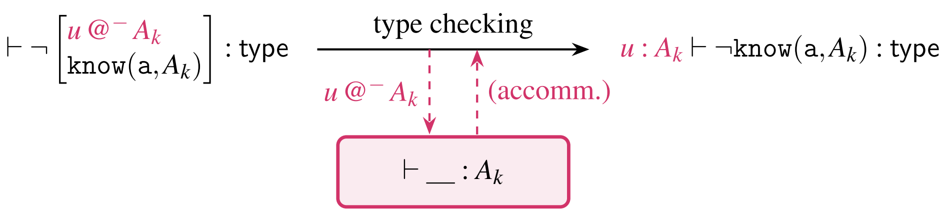

Figure 2. Visualization of the type checking of the semantic representation of (18).

We are now ready to analyze presupposition-pronoun interactions. We begin with (6a), where the presupposition can feed anaphora (a similar treatment applies to (7a)).

(6a) Alex did not know that Kim wrote a paper, and reviewed it.

The type checking process is shown in Figure 3. We first check the upper half of the representation (this process is identical to the one in Figure 2), resulting in a typing environment extended with $v : A_k$ via dynamic accommodation. Then, we check the $@^{+}$-type in the typing environment containing A_k (see the bottom-right box). This means that y can be resolved with $\pi_1 v$ (as in Figure 1), which correctly predicts the availability of pronominal anaphora in (6a).

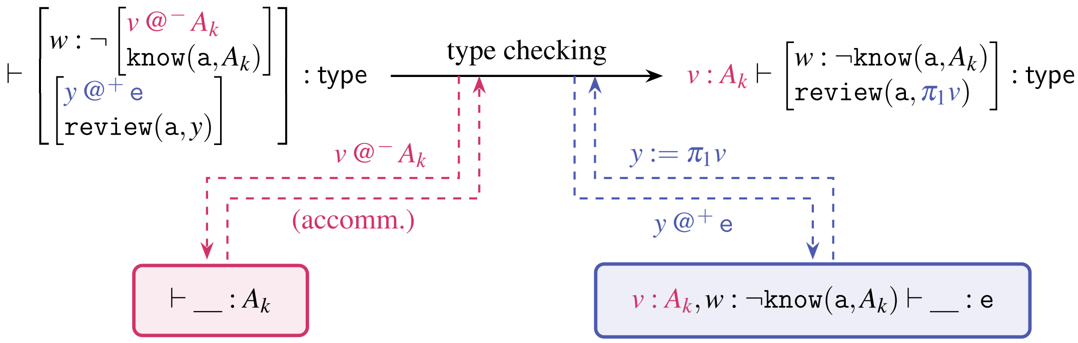

Figure 3. Visualization of the type checking of the semantic representation of (6a).

Turning to the GCO-violating case (7b), the tower restriction (26) guarantees that the possessive takes scope over the factive presupposition, leading to the representation shown below. Since the $@^{+}$-type must be resolved before the $@^{-}$-type is accommodated, we predict that there cannot be co-construal between *its* and *a paper*, as desired.

(7b) Its reviewer did not know that Kim wrote a paper. $\rightsquigarrow \begin{bmatrix} y @^{+} \mathtt{e} \\ \neg \begin{bmatrix} v @^{-} A_k \\ \mathtt{know}(\mathtt{reviewer}(y), A_k) \end{bmatrix} \end{bmatrix}$

Let us give an interim summary. With the continuation-based syntax-semantics interface, the GCO cases have parallel semantic representations, as shown below, where the @-type takes higher scope. Then, by ensuring that accommodation is sensitive to the type checking order, we can uniformly predict the unavailability of the semantic dependencies in the two cases.

semantically dependent expression ⋯ quantifier $\rightsquigarrow$ $y @^{\pm} B \cdots x : A$

semantically dependent expression ⋯ presup. trigger $\rightsquigarrow$ $y @^{\pm} B \cdots x @^{-} A$

5. Extensions

5.1. Donkey crossover

As mentioned in Section 1, donkey anaphora is also subject to CO. To analyze this effect, we start by looking at how DTS handles donkey anaphora in (3a), which is repeated below with the underspecified representation.

(3a) Every farmer who owns a donkey admires its strength.

$$\rightsquigarrow (x : \mathtt{e}) \to \left(\left(w : \begin{bmatrix} u : \mathtt{farmer}(x) \\ \begin{bmatrix} y : \mathtt{e} \\ \begin{bmatrix} v : \mathtt{donkey}(y) \\ \mathtt{own}(x, y) \end{bmatrix} \end{bmatrix} \end{bmatrix} \right) \to \begin{bmatrix} z @^{+} \mathtt{e} \\ \mathtt{admire}(x, \mathtt{strength}(z)) \end{bmatrix} \right)$$

Here, the $@^{+}$-type can be resolved with $\pi_1(\pi_2 w)$, which refers to $y : \mathtt{e}$ (i.e., the donkey owned by each farmer x). This result correctly predicts the co-variation between *a donkey* and *its*.

The upshot here is that the anaphoric dependency in the above example is only indirectly established via the universally quantified variable w. Namely, our prediction results from the scopal interaction between the pronoun *its* and the entire universal quantifier *every farmer who owns a donkey*, not with the indefinite *a donkey* itself, whose scope does not go out of the restrictor $w : [\cdots]$. As a result, the donkey crossover can be analyzed in exactly the same way as the classic CO cases. Namely, in cases like (3b), the pronoun must take scope over the quantifier due to the tower restriction, consequently making the anaphoric dependency impossible.

(3b) Its strength helps every farmer who owns a donkey.

$$\rightsquigarrow \begin{bmatrix} z \mathbin{@^+} \mathtt{e} \\ (x : \mathtt{e}) \to ((w : [\cdots]) \to \mathtt{help}(\cdots)) \end{bmatrix}$$

At this point, we emphasize that the precedence with respect to the type checking order is only a necessary condition for pronominal binding. To see how anaphora resolution is constrained, consider the anaphoric island with universal quantification, which is exemplified in (30).

(30) *A farmer who owns every donkey$_i$ admires its$_i$ strength.

$$\rightsquigarrow \begin{bmatrix} x : \mathtt{e} \\ \begin{bmatrix} w : \begin{bmatrix} u : \mathtt{farmer}(x) \\ (y : \mathtt{e}) \to ((v : \mathtt{donkey}(y)) \to \mathtt{own}(x,y)) \end{bmatrix} \\ (z \mathbin{@^+} \mathtt{e}) \times \cdots \end{bmatrix} \end{bmatrix}$$

The representation here involves the same type checking order as (3a). Although the proof search for $z \mathbin{@^+} \mathtt{e}$ can access $\pi_2 w : (y : \mathtt{e}) \to \cdots$, it does not provide a term of type $\mathtt{e}$, because there is no inference rule that allows deriving A from $(x : A) \to B$. In this way, DTS blocks anaphora with the condition of *proof constructability* besides the type checking order.

5.2. *Wh*-crossover

We move on to the CO effect exhibited by *wh*-elements, which is exemplified in (31).

(31) a. Which paper$_i$ __ surprised its$_i$ reviewer?
b. *Which paper$_i$ did its$_i$ reviewer praise __?

We can basically follow the idea of Barker and Shan (2014), who assumed that the trace of a *wh*-element is a scope taker. Concretely, we introduce an unpronounced operator (called a *gap*), defined as in (32). Here, GQ stands for S $/\!\!/$ (DP $\backslash\!\!\backslash$ S), the category of generalized quantifiers.

(32) $$\_\_ := \frac{\mathrm{GQ} \mathbin{\backslash\!\!\backslash} \mathrm{S} \mid \mathrm{S}}{\mathrm{DP}} : \frac{\lambda\sigma.\sigma(\lambda x.[\,])}{x}$$

Since $A \mathbin{\backslash\!\!\backslash} B$ indicates that B lacks A inside, GQ $\backslash\!\!\backslash$ S is the category of sentences waiting for a quantifier to be reconstructed at some position inside (e.g., __ *surprised its reviewer*). In the semantic representation, σ works on behalf of the scope taker that is to be reconstructed at the gap position. Below we show how a gap is combined with *surprised its reviewer*.[10]

$$\left(\begin{array}{c}\dfrac{\mathrm{GQ}\mathbin{\backslash\!\!\backslash}\mathrm{S}\mid\mathrm{S}}{\mathrm{DP}}\\ \_\_\\ \dfrac{\lambda\sigma.\sigma(\lambda y.[\,])}{y}\end{array}\quad\begin{array}{c}\dfrac{\mathrm{S}\mid\mathrm{S}}{\mathrm{DP}\backslash\mathrm{S}}\\ \text{surprised its rev.}\\ \dfrac{(x\,@^{+}\,\mathtt{e})\times[\,]}{\lambda z.\mathtt{spr}(z,\mathtt{rev}(x))}\end{array}\right)=\begin{array}{c}\dfrac{\mathrm{GQ}\mathbin{\backslash\!\!\backslash}\mathrm{S}\mid\mathrm{S}}{\mathrm{S}}\\ \_\_\text{ ... its ...}\\ \dfrac{\lambda\sigma.\sigma(\lambda y.(x\,@^{+}\,\mathtt{e})\times[\,])}{\mathtt{spr}(y,\mathtt{rev}(x))}\end{array}\xrightarrow{\text{LOWER}}\begin{array}{c}\mathrm{GQ}\mathbin{\backslash\!\!\backslash}\mathrm{S}\\ \_\_\text{ ... its ...}\\ \lambda\sigma.\sigma(\lambda y.(x\,@^{+}\,\mathtt{e})\times\mathtt{spr}(\cdots))\end{array}$$

Then, those phrases of category $\mathrm{GQ}\mathbin{\backslash\!\!\backslash}\mathrm{S}$ are passed to a *wh*-element as its argument. For instance, (33) shows the lexical entry for *which* (n^{*} stands for $(x:\mathtt{e})\times nx$, and $(u:_{Q}A)\times\cdots$ is a placeholder for the question operator, the specifics of which are irrelevant here).

(33) which $:=\ (\mathrm{S}_{whq}/(\mathrm{GQ}\mathbin{\backslash\!\!\backslash}\mathrm{S}))/\mathrm{N}\ :\ \lambda n.\lambda\kappa.\kappa(\lambda p.(u:_{Q}n^{*})\times p(\pi_{1}u))$

Below, we describe the β-reduction sequence after *which paper* is combined with __ *surprised its reviewer*. In the normal form, the scope of the question operator $(u:_{Q}\mathtt{paper}^{*})\times\cdots$ has come down to where σ was originally located, as desired.

$$\begin{aligned}&(\lambda\kappa.\kappa(\lambda p.(u:_{Q}\mathtt{paper}^{*})\times p(\pi_{1}u)))\ (\lambda\sigma.\sigma(\lambda y.(x\,@^{+}\,\mathtt{e})\times\mathtt{spr}(\cdots)))\\ \to_{\beta}\ &(\lambda\sigma.\sigma(\lambda y.(x\,@^{+}\,\mathtt{e})\times\mathtt{spr}(\cdots)))\ (\lambda p.(u:_{Q}\mathtt{paper}^{*})\times p(\pi_{1}u))\\ \to_{\beta}\ &(\lambda p.(u:_{Q}\mathtt{paper}^{*})\times p(\pi_{1}u))\ (\lambda y.(x\,@^{+}\,\mathtt{e})\times\mathtt{spr}(\cdots))\\ \twoheadrightarrow_{\beta}\ &(u:_{Q}\mathtt{paper}^{*})\times((x\,@^{+}\,\mathtt{e})\times\mathtt{spr}(\pi_{1}u,\mathtt{rev}(x)))\end{aligned}$$

Consequently, the interaction between a *wh*-element and a pronoun is reduced to that between the trace and the pronoun. This means that *wh*-crossover can be handled similarly to other CO cases. In (31b), for example, the gap cannot take scope over the preceding pronoun *its*, so $x\,@^{+}\,\mathtt{e}$ is type checked before $u:\mathtt{paper}^{*}$; hence no anaphoric dependency between them.

Furthermore, we can apply the present analysis to the *reconstruction effect*, which is exemplified in (34), under the assumption that gaps obey the tower restriction (26) as well as pronouns.[11] More concretely, if the gap is in the object position (34a), the universal quantifier $(x:\mathtt{e})\to\cdots$ takes scope over the $@^{+}$-type (contained in the restrictor of the question operator), so anaphora is predicted to be possible. In contrast, with the gap in the subject position (34b), $y\,@^{+}\,\mathtt{e}$ must be resolved earlier than the type checking of the Π-type, correctly prohibiting the anaphoric dependency between *theirs* and *every student*.

[10] To derive a phrase where a quantifier/pronoun precedes a gap (e.g., *every reviewer praise* __), we need to generalize its lexical entry. We illustrate this point with *every* below. Here, T is a meta-variable for categories resulting in S (e.g., S and $\mathrm{GQ}\mathbin{\backslash\!\!\backslash}\mathrm{S}$). $\lambda\vec{y}$ indicates the corresponding (possibly empty) sequence of lambda abstractions.

(i) every $:=\ \dfrac{T\mid T}{\mathrm{DP}}\Big/N\ :\ \lambda n.\dfrac{\lambda\vec{y}.(x:\mathtt{e})\to((u:nx)\to[\,]\vec{y})}{x}$

[11] This stipulation is independently motivated by the observation that the *functional answer* to a *wh*-question is subject to CO (Chierchia, 1993). For instance, we can attribute the unavailability of the functional answer in the example below to the assumption that *every student* cannot take scope over the gap.

(i) Which paper __ impressed every student? - *Their supervisor's.

(34) a. Which relative of theirs$_i$ does every student$_i$ love __?
$\rightsquigarrow$ $(x : \mathtt{e}) \to ((u : \mathtt{std}(x)) \to ((v :_Q ((y\,@^+\,\mathtt{e}) \times \cdots)) \times \cdots))$

b. *Which relative of theirs$_i$ __ loves every student$_i$?
$\rightsquigarrow$ $(u :_Q ((y\,@^+\,\mathtt{e}) \times \cdots)) \times ((x : \mathtt{e}) \to ((u : \mathtt{std}(x)) \to \cdots))$

6. Conclusion and open issues

This paper presented a DTS-based account of the GCO effect, employing the following two fundamental notions: (i) the propositions-as-types principle, which enables propositions to serve as a range of quantification, and (ii) the type checking order, defined in terms of the variable scope, which constrains how we can resolve semantic dependencies (formalized with @-types). On top of them, we formulated a syntax-semantics interface with continuation-based grammar, limiting the scope-taking potential of @-types with the tower restriction. Then, we introduced a dynamic mechanism of accommodation, thus ensuring that the context extension with a presupposition is sensitive to the type checking order. As a result, the GCO effect can be derived as a consequence of a single constraint on type checking in DTS, namely that @-types cannot be resolved with any elements that have not been type checked yet.

One potential shortcoming of our account is that the tower restriction may not be enough to prevent all CO-violating cases. For instance, consider a situation where a pronoun is part of a quantifier in the inverse scope of another, as in (35a). Here, due to the transitivity of the scopal relation, the $@^+$-type for the pronoun falls in the scope of the universal quantifier, so we would predict anaphora to be possible here, despite the inverse-scope status of the object quantifier. The same applies to dative constructions like (35b), where the pronoun is in the scope of the subject, which in turn is in the scope of the prepositional phrase.

(35) a. A student of theirs praised every professor. ($\forall > \exists$)
b. A girl sent her picture to every woman. ($\forall > \exists$)

However, we must be cautious about the empirical status of these cases. For instance, Bruening (2001: fn. 9) reported that the CO effect is absent in cases like (35a), where the pronoun occurs postnominally. Even for (35b), we might suspect that the anaphora between *her* and *every woman* is merely dispreferred due to the salience of the reading where the pronoun is bound by *a girl*. We believe that more fine-grained empirical evidence is necessary to settle this issue.

Another question to be addressed is whether the GCO generalization applies to proper names. Given a widespread view that a proper name presupposes the existence of its referent (see, e.g., Elbourne (2005)), we find the cataphora in (36) to be a GCO violation.

(36) Her friends praised Alex.

While the Binding Conditions (Chomsky, 1981) do not block coreference in (36), Ross et al. (2023) presented experimental evidence showing that the acceptance rate distribution of the proper name cataphora is similar to that of *wh*-crossover, potentially supporting our conclusion here. This observation invokes a broader question as to how the Binding Conditions should be incorporated into our type-theoretical approach, which we leave for future work.

References

Barker, C. (2012). Quantificational binding does not require c-command. *Linguistic Inquiry 43*(4), 614–633.

Barker, C. (2022). Rethinking scope islands. *Linguistic Inquiry 53*(4), 633–661.

Barker, C. and C.-c. Shan (2014). *Continuations and Natural Language*. Oxford Studies in Theoretical Linguistics. Oxford University Press.

Beaver, D. (2001). *Presupposition and Assertion in Dynamic Semantics*. CSLI Publications.

Bekki, D. (2023). A proof-theoretic analysis of weak crossover. In *New Frontiers in Artificial Intelligence: JSAI-isAI 2021 Workshops, JURISIN, LENLS18, SCIDOCA, Kansei-AI, AI-BIZ, Yokohama, Japan, November 13–15, 2021, Revised Selected Papers*, pp. 228–241. Springer Nature Switzerland.

Bekki, D. and K. Mineshima (2017). Context-passing and underspecification in dependent type semantics. In S. Chatzikyriakidis and Z. Luo (Eds.), *Modern Perspectives in Type-Theoretical Semantics*, pp. 11–41. Springer International Publishing.

Bruening, B. (2001). QR obeys superiority: Frozen scope and ACD. *Linguistic Inquiry 32*(2), 233–273.

Büring, D. (2004). Crossover situations. *Natural Language Semantics 12*(1), 23–62.

Chierchia, G. (1993). Questions with quantifiers. *Natural Language Semantics 1*(2), 181–234.

Chierchia, G. (2020). Origins of weak crossover: when dynamic semantics meets event semantics. *Natural Language Semantics 28*(1), 23–76.

Chomsky, N. (1981). *Lectures on Government and Binding: The Pisa Lectures*. Foris.

Elbourne, P. D. (2005). *Situations and Individuals*. MIT Press.

Elliott, P. D. and Y. Sudo (2021). Generalised crossover. In *Semantics and Linguistic Theory*, Volume 30, pp. 396–408.

Groenendijk, J. and M. Stokhof (1991). Dynamic predicate logic. *Linguistics and Philosophy 14*(1), 39–100.

Heim, I. (1983). On the projection problem for presuppositions. In *Proceedings of the Second West Coast Conference on Formal Linguistics*, pp. 114–125.

Howard, W. A. (1980). The formulae-as-types notion of construction. In H. Curry, H. B., S. J. Roger, and P. Jonathan (Eds.), *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*. Academic Press.

Lewis, D. (1979). Scorekeeping in a language game. *Journal of Philosophical Logic 8*, 339–359.

Luo, Z. (2012). Formal semantics in modern type theories with coercive subtyping. *Linguistics and Philosophy 35*(6), 491–513.

Martin-Löf, P. (1984). *Intuitionistic type theory*. Bibliopolis. Notes by Giovanni Sambin of a series of lectures given in Padua, June 1980.

Matsuoka, D., D. Bekki, and H. Yanaka (2024). Appositive projection as implicit context extension in Dependent Type Semantics. In *Logic and Engineering of Natural Language Semantics: 20th International Conference, LENLS20, Osaka, Japan, November 18–20, 2023, Revised Selected Papers*, pp. 224–243. Springer-Verlag.

May, R. C. (1977). *The Grammar of Quantification*. Ph. D. thesis, Massachusetts Institute of Technology.

Mayr, C. and B. Spector (2010). Not too strong! Generalizing the scope economy condition. In *Proceedings of Sinn und Bedeutung*, Volume 14, pp. 305–321.

Postal, P. (1971). *Cross-Over Phenomena*. Holt, Rinehart and Winston.

Potts, C. (2005). *The Logic of Conventional Implicatures*. Oxford Studies in Theoretical Linguistics. Oxford University Press.

Ranta, A. (1995). *Type-Theoretical Grammar*. Oxford University Press.

Reinhart, T. (1983). *Anaphora and Semantic Interpretation*. Routledge.

Ross, H., G. Chierchia, and K. Davidson (2023). Quantifying weak and strong crossover for wh-crossover and proper names. In *Proceedings of Sinn und Bedeutung*, Volume 27, pp. 535–553.

Ruys, E. G. (2000). Weak crossover as a scope phenomenon. *Linguistic Inquiry 31*(3), 513–539.

Safir, K. (2017). Weak crossover. In *The Wiley Blackwell Companion to Syntax, Second Edition*, pp. 1–40. John Wiley & Sons, Ltd.

Schlenker, P. (2011). DRT with local contexts. *Natural Language Semantics 19*(4), 373–392.

Shan, C.-c. and C. Barker (2006). Explaining crossover and superiority as left-to-right evaluation. *Linguistics and Philosophy 29*(1), 91–134.

Simons, M., J. Tonhauser, D. Beaver, and C. Roberts (2010). What projects and why. In *Semantics and Linguistic Theory*, Volume 20, pp. 309–327.

Sundholm, G. (1986). Proof theory and meaning. In D. Gabbay and F. Guenthner (Eds.), *Handbook of Philosophical Logic: Volume III: Alternatives in Classical Logic*, pp. 471–506. Springer Netherlands.

Tonhauser, J., D. Beaver, C. Roberts, and M. Simons (2013). Toward a taxonomy of projective content. *Language 89*(1), 66–109.

van der Sandt, R. (1992). Presupposition projection as anaphora resolution. *Journal of Semantics 9*(4), 333–377.