Chapter 16

Quantification

In the preceding chapter, our discourse was confined to predicates whose arguments were exclusively proper names, assuming for these a uniform semantic representation of type entity. However, a crucial distinction must now be drawn: the semantic representations of a class of noun phrases, termed *quantificational expressions* (or *quantificational noun phrases*) – exemplified by the bolded constituents in the sentences that follow – are not of type entity.

(75) a. **Every student** is tall.
b. **A/Some student** is tall.
c. **No student** is tall.
d. **Not every student** is tall.

The term "quantificational expressions" serves as a broad designation, encompassing not only those forms enumerated in (75) but also a diverse array of other nominal elements. These include, but are not limited to, numeral expressions such as *two students*, *more than two students*, *less than two students*, *exactly two students*, and *two or more students*. Further, the category subsumes noun phrases exhibiting internal quantification, exemplified by *only John*, *even John*, *students exept John*. Finally, it extends to other inherently quantifying nominals, such as *many students* and *large number of students*. [*1]

This chapter embarks upon an examination of the semantic representations of quantificational expressions, commencing with a foundational argument advanced by Heim and Kratzer (1998). Their seminal work compellingly demonstrates that the semantic type of such expressions does not reduce to the type entity[*2], which is to say, they

[*1] In contrast, a distinct class of noun phrases exists which does not fall under the rubric of quantificational expressions. This category comprises proper names, pronouns (e.g., *she/her*, *he/his/him*, *it/its*, *they/their/them*), demonstratives (e.g., *this student*, *that student*, *those students*), definite descriptions (e.g., *the student*, *the students*, *the two students*, *his students*), and coordination structures of noun phrases. It is pertinent to acknowledge the inherent difficulty in formulating a precise qualitative definition of "quantificational expression" based solely on the presence of quantification, given that definite descriptions themselves may incorporate quantificational elements. Nevertheless, each of the aforementioned categories of noun phrases forms a natural and cohesive group. Consequently, a more perspicuous definition of quantificational expressions can be established by defining them as a superset encompassing these naturally delineated categories.

[*2] The specific arguments presented in Section 16.1 derive directly from Chapter 9 of Heim and Kratzer (1998), a cornerstone text in formal semantics. While the core argument remain

do not denote simple entities. Following this introduction, we shall delve into the intrinsic nature of these semantic representations.

16.1 Quantificatinal Expressions are not of type Entity

Let us consider, for a moment, the analytical premise that quantificational expressions might be semantically treated akin to proper names. Under such a supposition, one could directly assign lexical item (76b) and (76c) to the expressions such as *some student* and *no student*, with denotations as the entities *someStudent* and *noStudent*. The crucial postulation here is that these expressions, despite their quantificational force, would denote terms of type entity, much like a proper name denotes a specific individual. Thus, we would posit, for example, $\vdash \mathit{someStudent} : \mathsf{entity}$, acknowledging, of course, that the precise identity of such an entity remains uninstantiated. [*3]

(76) a. $[\![\text{every student} \vdash NP]\!] \stackrel{def}{\equiv} \mathit{everyStudent}$

b. $[\![\text{a student, some student} \vdash NP]\!] \stackrel{def}{\equiv} \mathit{someStudent}$

c. $[\![\text{no student} \vdash NP]\!] \stackrel{def}{\equiv} \mathit{noStudent}$

d. $[\![\text{not every student} \vdash NP]\!] \stackrel{def}{\equiv} \mathit{notEveryStudent}$

In contrast to the above discussion, Heim and Kratzer (1998) articulate compelling counter-arguments regarding the semantic type of quantificational expressions. We shall explore two such arguments in the ensuing subsections.[*4]

16.1.1 Argument based on Modifiers

An empirical entailment holds between the following two sentences: (77a) entails (77b). That is to say, no situation exists wherein (77a) is true and (77b) is false. This deductive relation, as discussed in Section Section **??**, constitutes an observed datum, explicitly formalized as (77)

(77) Mary is a good pianist. $\Longrightarrow$ Mary is a pianist.

a. Mary is a good pianist.

b. Mary is a pianist.

---

faithful to the original exposition, the examples have been carefully replaced with those from the primary literature to best serve our present purposes.

[*3] For the ensuing discourse, and for the sake of conciseness, we shall henceforth abbreviate entity as e.

[*4] It is crucial to note that Heim and Kratzer's original critique is framed within the context of Montagovian model-theoretic semantics, where quantificational expressions are argued not to be of type e (entities). While their underlying formal theory thus diverges from DTS, the essence of their argument retains its force. Herein, we re-present these arguments, adapting them to a framework grounded in DTT, asserting that the semantic representations of quantificational expressions are similarly not of the basic type entity within DTT. The core logical structure of their challenge remains invariant under this theoretical transposition.

DTS offers a precise account of this entailment. Suppose that the semantic representations of (77a) and (77b) are calculated as (78a) and (78b).

(78) a. $\mathbf{good}(mary) \times \mathbf{pianist}(mary)$
b. $\mathbf{pianist}(mary)$

The inference presented in (77) is correctly predicted to hold, a consequence directly attributable to the valid deduction between its corresponding semantic representations.

(79) Proof diagram of (77)

$$\frac{u : \mathbf{good}(mary) \times \mathbf{pianist}(mary)}{\pi_2 u : \mathbf{pianist}(mary)}\,{\scriptstyle (\Sigma E)}$$

This deductive validity, however, is not confined to proper names; it extends to any repreesntation instantiating the semantic type entity. Consequently, were quantificational expressions to share this same entity-type semantic representation as proper names, the inference in (80) would be similarly predicted as valid.

(80) No student is a good pianist. $\not\Longrightarrow$ No student is a pianist.
a. No student is a good pianist.
b. No student is a pianist.

This prediction, however, is demonstrably incorrect. Consider a scenario where several student-pianists exist, yet none possess considerable skill. In such a context, (80a) is veridical, while (80b) is false.

Let us trace the derivation that yields this erroneous prediction. Suppose the quantificational expression *no student* is assigned the semantic representation in (76c). Through an identical process of syntactic structuring and semantic composition, as delineated in (227a) and (220), the following problematic semantic representations are derived.

(81) a. $\mathbf{good}(noStudent) \times \mathbf{pianist}(noStudent)$
b. $\mathbf{pianist}(noStudent)$

Thus, the following inference is expected to hold.

(82) Proof diagram of (80)

$$\frac{u : \mathbf{good}(noStudent) \times \mathbf{pianist}(noStudent)}{\pi_2 u : \mathbf{pianist}(noStudent)}\,{\scriptstyle (\Sigma E)}$$

Consequently, the hypothesis that a quantificational expression such as *no student* functions as a noun phrase with a semantic representation of type entity must be re-evaluated. This constitutes a primary argument against assigning an entity type to the semantic representation of quantificational expressions of the form *no N*.

Exercise 337. Each of the following inferences is demonstrably valid. Consider whether this validity constitutes a counter-argument to the preceding discussion.

(83) a. Every student is a good pianist. $\Longrightarrow$ Every student is a pianist.
b. Some student is a good pianist. $\Longrightarrow$ Some student is a pianist.

16.1.2 Argument based on LNC

Heim and Kratzer (1998)'s second argument centers on the Law of Non-Contradiction (LNC). Consider the contrasting pair (84a) and (84b). Crucially, sentence (84a) is inherently contradictory, assuming a unique individual named Chomsky. Sentence (84b), however, is not.

(84) a. # Chomsky wrote *Syntactic Structures*, and/but Chomsky didn't write *Syntactic Strucutures*.

b. Some linguist wrote *Syntactic Structures*, and/but some linguist didn't write *Syntactic Structures*.

The contradictory nature of (84a) is predicted as follows.

(85) Syntactic structure of (84a)

$$\dfrac{\dfrac{\text{Chomsky wrote Syntactic Structures}}{S} \quad \dfrac{\text{but}}{CONJ_S} \quad \dfrac{\text{Chomsky didn't write Syntactic Structures}}{S}}{S}\,\langle\Phi\rangle$$

(86) Semantic composition of (84a)

$$\dfrac{\dfrac{\text{Chomsky wrote Syntactic Structures}}{\mathbf{write}(chomsky, SS)} \quad \dfrac{\text{but}}{\mathbf{and}_S} \quad \dfrac{\text{Chomsky didn't write Syntactic Structures}}{\neg\mathbf{write}(chomsky, SS)}}{\mathbf{write}(chomsky, SS) \times \neg\mathbf{write}(chomsky, SS)}\,\langle\Phi\rangle$$

This immediate contradiction arises irrespective of the specific proper name and the intransitive verb employed here.

(87) Proof diagram of (84a)

$$\dfrac{\dfrac{m : \begin{bmatrix} \mathbf{write}(chomsky, SS) \\ \neg\mathbf{write}(chomsky, SS) \end{bmatrix}}{\pi_2 m : \neg\mathbf{write}(chomsky, SS)}\,(\Sigma E) \quad \dfrac{m : \begin{bmatrix} \mathbf{write}(chomsky, SS) \\ \neg\mathbf{write}(chomsky, SS) \end{bmatrix}}{\pi_1 m : \mathbf{write}(chomsky, SS)}\,(\Sigma E)}{(\pi_2 m)(\pi_1 m) : \bot}\,(\Pi E)$$

Should *some linguist* be analyzed as a proper name, sentence (84b) would similarly be predicted to be contradictory. This prediction, however, stands in stark empirical contrast to our intuitions. Therefore, *some linguist* cannot be treated as a proper name in our semantic framework.[*5]

[*5] Consider the following contradictory sentence.

(1) Every linguist wrote *Syntactic Structures*,
{and | but} every linguist didn't write *Syntactic Structures*.

The logical incompatibility inheres in the assertion of both universal affirmation and universal negation over the same predicate.

However, the empirical falsehood of the conjunct *Every linguist wrote Syntactic Structures* – a premise patently untrue given the vast majority of linguists did not author the work in question – complicates its utility as a diagnostic tool. When presented to an informant, this initial falsity can obscure the underlying logical inconsistency, potentially eliciting a judgment of mere untruth rather than inherent contradiction.

Exercise 338. The preceding analysis hinges on a crucial premise: that the two instantiations of *some linguist* in example (84b) are assigned an identical constant, *someLinguist*. Consider, however, an alternative assignment where each occurrence of *some linguist* is instead mapped to a distinct constant, $someLinguist_i$ (for $i \in \mathbb{N}$). Formulate a counter-argument that challenges the conclusions drawn under this revised semantic mapping.

The discourse in the preceding two subsections establishes a fundamental tenet: the semantic representations of quantificational expressions are demonstrably not of type entity.

Exercise 339. Consider whether the semantic representation of the phrase *not every* can be similarly shown to be incompatible with type entity, employing either of the two arguments previously delineated.

16.2 Semantic Representations for Quantification

The central inquiry concerning quantificational expressions, if their semantic representations do not resolve to the type entity, is their true typal assignment. To address this, we initiate our analysis by examining the semantic representation of sentences in which such expressions are embedded.

Consider, for instance, the standard treatment within FoL that the logical formulas corresponding to, for example, (75a)and (75b) are (88a) and (88b).

(75a) Every student is tall.

(75c) Some student is tall.

(88) a. $\forall x(\mathbf{student}(x) \rightarrow \mathbf{tall}(x))$

b. $\exists x(\mathbf{student}(x) \wedge \mathbf{tall}(x))$

Exercise 340. Explain why the FoL representations for (75a) and (75b) is NOT (**??**) and (**??**), respectively.

(89) a. $\forall x(\mathbf{student}(x) \wedge \mathbf{tall}(x))$

b. $\exists x(\mathbf{student}(x) \rightarrow \mathbf{tall}(x))$

While this exercise pertains to elementary symbolic logic, the underlying rationale – why a given logical formula fails to capture the semantic essence of a natural language sentence – is fundamentally a question for natural language semantics.

As established in Exercise **??**, the FoL formulae presented in (88a) and (88b) demonstrably possess a degree of validity as semantic representations for (75a) and (75b). Consequently, we propose to investigate, as alternative candidates for these semantic representations, the DTT formulae given in (90a) and (90b), respectively. These DTT formulae are specifically constructed to correspond to their FoL counterparts.

(90) a. $\left(u : \begin{bmatrix} x : \mathsf{entity} \\ \mathbf{student}(x) \end{bmatrix}\right) \rightarrow \mathbf{tall}(\pi_1 u)$

b. $\begin{bmatrix} u : \begin{bmatrix} x : \mathsf{entity} \\ \mathbf{student}(x) \end{bmatrix} \\ \mathbf{tall}(\pi_1 u) \end{bmatrix}$

It is crucial to note that within the framework of DTT, the semantic representations in (90a) and (90b) are proof-theoretically equivalent to the alternative forms presented in (91a) and (91b), respectively.

(91) a. $(x : \mathsf{entity}) \to \mathbf{student}(x) \to \mathbf{tall}(x)$

b. $\begin{bmatrix} x : \mathsf{entity} \\ \begin{bmatrix} \mathbf{student}(x) \\ \mathbf{tall}(x) \end{bmatrix} \end{bmatrix}$

Exercise 341. Prove the proof-theopretic equivalence mentioned above.

Exercise 342. Confirm that (90a), (90b), (91a), and (91b) are well-formed type.

16.2.1 every

We now turn to the hypothesis that the semantic representations for the expressions illustrated in (90a) and (90b) correspond directly to the structures of (75a) and (75b), respectively. Our initial focus will be the universally quantified sentence, (75a).

(75a) Every student is tall.

(90a) $\left(u : \begin{bmatrix} x : \mathsf{entity} \\ \mathbf{student}(x) \end{bmatrix}\right) \to \mathbf{tall}(\pi_1 u)$

Our analysis of (75a) proceeds through an examination of a collection of inferences in which it participates. While the set of such inferences is, in principle, unbounded, Dependent Type Semantics (DTS) posits the verification condition of (75a) as fundamentally central. This centrality arises from a key advantage of verificationist semantics over its truth-conditional counterpart: if the minimal conditions for demonstrating (75a) can be precisely articulated as a type, then the validity of all other pertinent inferences necessarily follows from that very type.

Expressed as an inference, the verification condition specifies the required context for a judgment where the semantic representation of the target sentence (here, (90a)) stands alone as the consequent. For (75a), this condition dictates that, to establish its truth, for every individual entity, one must possess either a proof that said individual is tall or a proof that they are not a student—($\star$). In what follows, we will demonstrate that the semantic representation of (75a) is indeed provable within a context that satisfies this verification condition.

1. For the sake of expositional clarity, let us constrain our entity type such that it comprises precisely two constructors, *john* and *mary*, namely, only John and Mary exist in the world: $\mathsf{entity} \stackrel{def}{\equiv} \{john, mary\}$. Within this circumscribed domain, the verification condition necessitates that the proposition ($\star$) holds for both *john* and *mary*. Here, let stipulate the truth of $\mathbf{tall}(john)$ and $\neg\mathbf{student}(mary)$.
2. The context corresponding to this stipulated scenario can be expressed as $\Gamma \equiv t_j : \mathbf{tall}(john),\ s_m : \neg\mathbf{student}(mary)$.
3. We shall adopt the semantic representation (91a) in lieu of (90a) for (75a), so that the proof becomes more concise.
4. The demonstration that $\Gamma \vdash (91a)$ suffices to establish the truth of (90a) under the given scenario.

Let P be $\lambda x.(\mathbf{student}(x) \to \mathbf{tall}(x))$ and the proof diagram of $P : \mathsf{entity} \to \mathsf{type}$ be $\mathcal{D}_P$, then the proof for (91a) can be constructed as follows.

(92) Proof diagram of (75a)

$$\dfrac{\dfrac{\dfrac{\overline{x : \mathsf{entity}}^{1} \quad \mathcal{D}_P \quad \dfrac{\dfrac{\dfrac{\overline{y : \mathbf{student}(john)}^{2} \quad t_j : \mathbf{tall}(john)}{t_j : \mathbf{tall}(john)}{\scriptstyle (WK)}}{\lambda y.t_j : \mathbf{student}(john) \to \mathbf{tall}(john)}{\scriptstyle (\Pi I),1}}{\lambda y.t_j : P(john)}{\scriptstyle (CONV)} \quad \dfrac{\dfrac{\dfrac{\dfrac{\overline{y : \mathbf{student}(mary)}^{2} \quad s_m : \neg\mathbf{student}(mary)}{s_m(y) : \bot}{\scriptstyle (\Pi E)}}{\mathsf{case}_{s_m(y)}() : \mathbf{tall}(mary)}{\scriptstyle (EFQ)}}{\lambda y.\mathsf{case}_{s_m(y)}() : \mathbf{student}(mary) \to \mathbf{tall}(mary)}{\scriptstyle (\Pi I),2}}{\lambda y.\mathsf{case}_{s_m(y)}() : P(mary)}{\scriptstyle (CONV)}}{\mathsf{case}_x^P(\lambda y.t_j, \lambda y.\mathsf{case}_{s_m(y)}()) : P(x)}{\scriptstyle (\{\}E)}}{\mathsf{case}_x^P(\lambda y.t_j, \lambda y.\mathsf{case}_{s_m(y)}()) : \mathbf{student}(x) \to \mathbf{tall}(x)}{\scriptstyle (CONV)}}{\lambda x.\mathsf{case}_x^P(\lambda y.t_j, \lambda y.\mathsf{case}_{s_m(y)}()) : (x : \mathsf{entity}) \to \mathbf{student}(x) \to \mathbf{tall}(x)}{\scriptstyle (\Sigma I),2}$$

The structure of the proof reveals its inherent generality: the validity of the argument holds irrespective of whether the canonical elements of type entity are restricted to specific individuals such as *john* and *mary*, or encompass an arbitrary set of canonical elements. Consequently, (91a) (and by the proof-theoretic equivalence, (90a)) precisely articulates the necessary and sufficient verification conditions for expression (75a).

The assertion that (90a) serves as the correct semantic representation for (75a) can be substantiated through the following inferential schema, among others.

(93) a. Every student is tall. John is a student. $\Longrightarrow$ John is tall.
b. Every student is a vegetarian. Every vegetarian loves avocados. $\Longrightarrow$ Every student loves avocados.

Exercise 343. Check if the inferences in (93) are predicted correctly.

The failure of the following inference provides corroborating evidence for the claim that (90a) is the correct semantic representation of (75a).

(94) Every student is tall. Mary is not a student. $\not\Longrightarrow$ Mary is tall.

A critical distinction between DTT and decidable logical systems, such as propositional logic, lies in the absence of a general procedure for demonstrating the invalidity of an inference. In the latter, one can, in a finite number of steps, employ truth tables or semantic tableaux to definitively prove that a particular inference does not hold. Such a universally applicable method is not present in DTT.

Consequently, data points like those exemplified in (94) assume a distinct epistemological role. If the inference suggested by such data were to be formally proven valid within DTT, it would serve as a disproof of the linguistic analysis currently under consideration. Conversely, the persistent lack of such a proof functions as corroborating evidence in support of the proposed analysis. This nuanced interpretive framework underscores a fundamental characteristic of formal inquiry within DTT.

16.2.2 some, a

We now turn our attention to the existential quantification expressed in sentence (75b). Our objective is to ascertain whether its semantic representation can be adequately captured by (90b).

(75b) A/Some student is tall.

The verification condition for this assertion is straightforward: there exists an individual – any individual – who simultaneously satisfies the predicates of being a student and possessing the property of tallness. This condition, for instance, finds satisfaction under the following set of premises:

(95) John is a student and is tall. $\Longrightarrow$ A student is tall.

This verification condition is formally represented by (**??**).

(96) Proof diagram of (95)

$$\cfrac{\cfrac{\cfrac{}{john : \mathsf{entity}}\,\{\}I \qquad j_s : \mathbf{student}(john)}{(john, j_s) : \left[\begin{array}{l} x : \mathsf{entity} \\ \mathbf{student}(x)\end{array}\right]}\,(\Sigma I) \qquad t_j : \mathbf{tall}(john)}{((john, j_s), t_j) : \left[\begin{array}{l} u : \left[\begin{array}{l} x : \mathsf{entity} \\ \mathbf{student}(x)\end{array}\right] \\ \mathbf{tall}(x)\end{array}\right]}\,(\Sigma I)$$

The hypothesis positing that the semantic representation of (75b) is precisely (90b) is rigorously evaluated by means of the following set of inferences.

(97) a. A student is tall. $\Longrightarrow$ There is a student.
b. A student is tall. $\Longrightarrow$ Somebody is tall.
c. A student is tall. John is a student. $\not\Longrightarrow$ John is tall.
d. Some reptiles are herbivores. $\Longrightarrow$ Some herbivores are reptiles.
e. $\Longrightarrow$ Some reptiles are reptiles.

The validity of these inferences is predicted by their respective proofs.

(98) Proof diagram of (97a)

$$x_1 : \left[\begin{array}{l} u : \left[\begin{array}{l} x : \mathsf{entity} \\ \mathbf{student}(x)\end{array}\right] \\ \mathbf{tall}(\pi_1 u)\end{array}\right] \vdash \left[\begin{array}{l} x : \mathsf{entity} \\ \mathbf{student}(x)\end{array}\right] \; true$$

(99) Proof diagram of (97a)

$$x_1 : \left[\begin{array}{l} u : \left[\begin{array}{l} x : \mathsf{entity} \\ \mathbf{student}(x)\end{array}\right] \\ \mathbf{tall}(\pi_1 u)\end{array}\right] \vdash \left[\begin{array}{l} u : \left[\begin{array}{l} x : \mathsf{entity} \\ \mathbf{human}(x)\end{array}\right] \\ \mathbf{tall}(\pi_1 u)\end{array}\right] \; true$$

(100) Proof diagram of (97a)

$$s_1 : \mathbf{tall}(john), e_1 : \mathbf{student}(john) \vdash \left[\begin{array}{l} u : \left[\begin{array}{l} x : \mathsf{entity} \\ \mathbf{student}(x)\end{array}\right] \\ \mathbf{tall}(\pi_1 u)\end{array}\right] \; true$$

Exercise 344. Check if DTS correctly predicts the patterns in (97).

Exercise 345. Examine the following entailment.

(101) A man entered. $\not\Longrightarrow$ Every man entered.

(102) Every man entered. $\Longrightarrow$ A man entered.

The empirical validity of example (102), though undeniable, is not a direct prediction of DTS. It bears noting that standard First-Order Logic (FoL) similarly fails to account for this phenomenon. The observed truth of (102) stems from what has been termed the existential presupposition inherent in certain quantifiers. This critical aspect of quantifier semantics, which posits the existence of entities satisfying the quantified property, will be revisited in greater detail in Chapter Chapter **??**.

16.2.3 no, not every

The respective meanings of expressions *no* and *not every* are formally rendered in FoL. Specifically, the semantic representation for an utterance exemplified by (75c) is given by (103a), while that for an utterance such as (75d) is correspondingly assigned (103b).

(75c) No student is tall.

(75d) Not every student is tall.

(103) a. $\forall x(\mathbf{student}(x) \to \neg\mathbf{tall}(x))$

b. $\neg\forall x(\mathbf{student}(x) \to \mathbf{tall}(x))$

The corresponding semantic representations in DTS are given below.

(104) a. $\left(u : \begin{bmatrix} x : \mathsf{entity} \\ \mathbf{student}(x) \end{bmatrix}\right) \to \neg\mathbf{tall}(\pi_1(u))$

b. $\neg\left(\left(u : \begin{bmatrix} x : \mathsf{entity} \\ \mathbf{student}(x) \end{bmatrix}\right) \to \mathbf{tall}(\pi_1(u))\right)$

The verification condition for (75c) represents the negative counterpart to that of (75a). Succinctly, to establish the truth of (75c), one must demonstrate, for every entity, either its non-membership in the set of students or its failure to posses the property of being tall.

Exercise 346. Following the discussion of Subsection 16.2.1, set up a context that satisfies the above verification conditions and prove that (104a) is true under that context.

Exercise 347. Consider the verification conditions for (75d) and examine (104b) by using the same procedure.

Exercise 348. Check if the inferences in (105) are correctly predicted.

(105) a. No student is tall. John is a student. $\Longrightarrow$ John didn't show up.

b. Every student is a female. No female jogs. $\Longrightarrow$ No student jogs.

c. All grizzly are bears. No bears are reptiles. $\Longrightarrow$ No reptiles are grizzly.

d. All bears are mammals. Some bears are grizzly. $\Longrightarrow$ Some mammals are grizzly.

e. All grizzly are bears. Some grey bears are grizzly. $\Longrightarrow$ Some bears are grey bears.

f. Some bears are grizzly. No bears are grizzly. $\Longrightarrow$

16.3 Lexicalizing Quantifiers

16.3.1 Lexical Mearning of Quantificational Expressions

What can we say about the lexical items of quantifiers? If we assume that the syntactic type of every student is X, and that the lexical items of *student* and *is tall* are the same as those discussed so far, then the syntactic structure should be as follows.

(106) Syntactic structure of (75a)

$$\cfrac{\cfrac{\cfrac{\text{Every}}{X/N} \quad \cfrac{\text{student}}{N}}{X}{>} \quad \cfrac{\text{is tall}}{S\backslash NP}}{S}{?}$$

Here, there are only two possibilities in which X and $S\backslash NP$ can be combined and becomes S under the CCG combinatorial rules (Exercise: think why there are no other rule possibilities).

1. $X \equiv NP$ and the rule is $<$
2. $X \equiv S/(S\backslash NP)$ and the rule is $>$

First, in the case of $X \equiv NP$, then the type of the semantic representation of *every student* is $\lceil NP \rceil = \mathsf{entity}$. However, this possibility is ruled out because we know from the discussion in the previous section that the type of the semantic representation of *every student* is not entity. Therefore, we can conclude that $X \equiv S/(S\backslash NP)$.

Next, if the semantic representation of *every* is Q, the semantic composition of (75a) is as follows.

(107) Semantic composition of (75a)

$$\cfrac{\cfrac{\cfrac{\text{Every}}{Q} \quad \cfrac{\text{student}}{\mathbf{student}}}{Q(\mathbf{student})}{>} \quad \cfrac{\text{is tall}}{\mathbf{tall}}}{\left(u : \begin{bmatrix} x : \mathsf{entity} \\ \mathbf{student}(x) \end{bmatrix}\right) \to \mathbf{tall}(\pi_1 u)}{?}$$

The type of the semantic representation of $Q(\mathbf{student})$ is $\lceil S/(S\backslash NP) \rceil = (\mathsf{entity} \to \mathsf{type}) \to \mathsf{type}$. According to the diagram above, since we are taking a semantic representation $\mathbf{tall}$ of type $\mathsf{entity} \to \mathsf{type}$ and returning

(108) $$\left(u : \begin{bmatrix} x : \mathsf{entity} \\ \mathbf{student}(x) \end{bmatrix}\right) \to \mathbf{tall}(\pi_1 u)$$

it must be

(109) $$Q(\mathbf{student}) \equiv \lambda p. \left(u : \begin{bmatrix} x : \mathsf{entity} \\ \mathbf{student}(x) \end{bmatrix}\right) \to p(\pi_1 u)$$

Then Q first has the syntactic type $X/N \equiv S/(S\backslash NP)/N$, and therefore the type of semantic representation of Q is $\lceil S/(S\backslash NP)/N \rceil = (\mathsf{entity} \to \mathsf{type}) \to (\mathsf{entity} \to \mathsf{type}) \to \mathsf{type}$. This is because it takes the semantic representation $\mathbf{student}$ of type $\mathsf{entity} \to \mathsf{type}$ and returns the (109),

(110) $Q \equiv \lambda n.\lambda p.\left(u : \begin{bmatrix} x : \mathsf{entity} \\ n(x) \end{bmatrix}\right) \to p(\pi_1 u)$

Therefore, the lexical item of *every* should be given as follows.

(111) $[\![\text{every} \vdash S/(S\backslash NP)/N]\!] \overset{def}{\equiv} \lambda n.\lambda p.\left(u : \begin{bmatrix} x : \mathsf{entity} \\ n(x) \end{bmatrix}\right) \to p(\pi_1 u)$

Theorem 349. (111) satifies the semantic felicity condition.

Proof. $\lceil S/(S\backslash NP)/N \rceil = \lceil N \rceil \to (\lceil NP \rceil \to \lceil S \rceil) \to \lceil S \rceil = (\mathsf{entity} \to \mathsf{type}) \to (\mathsf{entity} \to \mathsf{type}) \to \mathsf{type}$, so it safices to show the following.

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{$(\{\}F)$}
\UnaryInfC{$\mathsf{entity} : \mathsf{type}$}
\AxiomC{$\overline{n : \mathsf{entity} \to \mathsf{type}}^{4}$}
\AxiomC{$\overline{x : \mathsf{entity}}^{1}$}
\RightLabel{(ΠE)}
\BinaryInfC{$n(x) : \mathsf{type}$}
\RightLabel{$(\Sigma F),1$}
\BinaryInfC{$\begin{bmatrix} x : \mathsf{entity} \\ n(x) \end{bmatrix} : \mathsf{type}$}
\AxiomC{$\overline{p : \mathsf{entity} \to \mathsf{type}}^{3}$}
\AxiomC{$\overline{u : \begin{bmatrix} x : \mathsf{entity} \\ n(x) \end{bmatrix}}^{2}$}
\RightLabel{(ΣE)}
\UnaryInfC{$\pi_1 u : \mathsf{entity}$}
\BinaryInfC{$p(\pi_1 u) : \mathsf{type}$}
\RightLabel{$(\Pi F),2$}
\BinaryInfC{$\left(u : \begin{bmatrix} x : \mathsf{entity} \\ n(x) \end{bmatrix}\right) \to p(\pi_1 u) : \mathsf{type}$}
\RightLabel{$(\Pi I),3$}
\UnaryInfC{$\lambda p.\left(u : \begin{bmatrix} x : \mathsf{entity} \\ n(x) \end{bmatrix}\right) \to p(\pi_1 u) : (\mathsf{entity} \to \mathsf{type}) \to \mathsf{type}$}
\RightLabel{$(\Pi I),4$}
\UnaryInfC{$\lambda n.\lambda p.\left(u : \begin{bmatrix} x : \mathsf{entity} \\ n(x) \end{bmatrix}\right) \to p(\pi_1 u) : (\mathsf{entity} \to \mathsf{type}) \to (\mathsf{entity} \to \mathsf{type}) \to \mathsf{type}$}
\end{prooftree}
$$

□

In the same way, it is concluded that the semantic representations of *a/an*, *some*, *no*, *not every*.

(112) a. $[\![\text{a,an,some} \vdash S/(S\backslash NP)/N]\!] \overset{def}{\equiv} \lambda n.\lambda p.\begin{bmatrix} u : \begin{bmatrix} x : \mathsf{entity} \\ n(x) \end{bmatrix} \\ p(\pi_1 u) \end{bmatrix}$

b. $[\![\text{no} \vdash S/(S\backslash NP)/N]\!] \overset{def}{\equiv} \lambda n.\lambda p.\left(u : \begin{bmatrix} x : \mathsf{entity} \\ n(x) \end{bmatrix}\right) \to \neg p(\pi_1 u)$

c. $[\![\text{not every} \vdash S/(S\backslash NP)/N]\!] \overset{def}{\equiv} \lambda n.\lambda p.\neg\left(\left(u : \begin{bmatrix} x : \mathsf{entity} \\ n(x) \end{bmatrix}\right) \to p(\pi_1 u)\right)$

Exercise 350. Following Theorem 349, show that each of (112a), (112b), and (112c) satifies the semantic felicity condition.

16.3.2 On Modifier Puzzle

Under the semantic representation (112b), the modifier puzzle is resolved. First, the syntactic structure and semantic representations of (80a) and (80b) are as follows.

(113) Syntactic structure of (80a)

$$\dfrac{\dfrac{\dfrac{\text{No}}{S/(S\backslash NP)/N}\quad\dfrac{\text{student}}{N}}{S/(S\backslash NP)}{>}\quad \dfrac{\dfrac{\text{is}}{S\backslash NP/(S\backslash NP)}\quad \dfrac{\dfrac{\text{a}}{S\backslash NP/N}\quad\dfrac{\dfrac{\text{good}}{N/N}\quad\dfrac{\text{pianist}}{N}}{N}{>}}{S\backslash NP}{>}}{S\backslash NP}{>}}{S}{>}$$

(114) Semantic composition of (80a)

$$\dfrac{\dfrac{\dfrac{\text{No}}{\lambda n.\lambda p.\left(u:\begin{bmatrix}x:\mathsf{entity}\\ n(x)\end{bmatrix}\right)\to\neg p(\pi_1 u)}\quad\dfrac{\text{student}}{\mathbf{student}}}{\lambda p.\left(u:\begin{bmatrix}x:\mathsf{entity}\\ \mathbf{student}(x)\end{bmatrix}\right)\to\neg p(\pi_1 u)}{>}\quad\dfrac{\dfrac{\text{is}}{id}\quad\dfrac{\dfrac{\text{a}}{id}\quad\dfrac{\dfrac{\text{good}}{\lambda n.\lambda x.(\mathbf{good}(x)\times n(x))}\quad\dfrac{\text{pianist}}{\mathbf{pianist}}}{\lambda x.(\mathbf{good}(x)\times\mathbf{pianist}(x))}{>}}{\lambda x.(\mathbf{good}(x)\times\mathbf{pianist}(x))}{>}}{\lambda x.(\mathbf{good}(x)\times\mathbf{pianist}(x))}{>}}{\left(u:\begin{bmatrix}x:\mathsf{entity}\\ \mathbf{student}(x)\end{bmatrix}\right)\to\neg\begin{bmatrix}\mathbf{good}(\pi_1 u)\\ \mathbf{pianist}(\pi_1 u)\end{bmatrix}}{>}$$

(115) Syntactic structure of (80b)

$$\dfrac{\dfrac{\dfrac{\text{No}}{S/(S\backslash NP)/N}\quad\dfrac{\text{student}}{N}}{S/(S\backslash NP)}{>}\quad \dfrac{\dfrac{\text{is}}{S\backslash NP/(S\backslash NP)}\quad \dfrac{\dfrac{\text{a}}{S\backslash NP/N}\quad\dfrac{\text{pianist}}{N}}{S\backslash NP}{>}}{S\backslash NP}{>}}{S}{>}$$

(116) Semantic composition of (80b)

$$\dfrac{\dfrac{\dfrac{\text{No}}{\lambda n.\lambda p.\left(u:\begin{bmatrix}x:\mathsf{entity}\\ n(x)\end{bmatrix}\right)\to\neg p(\pi_1 u)}\quad\dfrac{\text{student}}{\mathbf{student}}}{\lambda p.\left(u:\begin{bmatrix}x:\mathsf{entity}\\ \mathbf{student}(x)\end{bmatrix}\right)\to\neg p(\pi_1 u)}{>}\quad\dfrac{\dfrac{\text{is}}{id}\quad\dfrac{\dfrac{\text{a}}{id}\quad\dfrac{\text{pianist}}{\mathbf{pianist}}}{\lambda x.\mathbf{pianist}(x)}{>}}{\lambda x.\mathbf{pianist}(x)}{>}}{\left(u:\begin{bmatrix}x:\mathsf{entity}\\ \mathbf{student}(x)\end{bmatrix}\right)\to\neg\mathbf{pianist}(\pi_1 u)}{>}$$

Corresponding to (80b) $\Longrightarrow$ (80a) is the following inference.

(117) $$\left(u:\begin{bmatrix}x:\mathsf{entity}\\ \mathbf{student}(x)\end{bmatrix}\right)\to\neg\begin{bmatrix}\mathbf{good}(\pi_1 u)\\ \mathbf{pianist}(\pi_1 u)\end{bmatrix}\vdash\left(u:\begin{bmatrix}x:\mathsf{entity}\\ \mathbf{student}(x)\end{bmatrix}\right)\to\neg\mathbf{pianist}(\pi_1 u)$$

However, this does not hold. Compare with the proof of Section 16.1.1.

16.3.3 On LNC Puzzle

Next, check that LNC puzzle does not occur under the semantic representation (112a).

(84b) Someone wrote Syntactic Structures, and someone didn't write Syntactic Structures.

The syntactic structure and semantic composition of (84b) is as follows.

(118) Syntactic structure of (84b)

$$\dfrac{\dfrac{\dfrac{\text{Someone}}{S/(S\backslash NP)}\quad\dfrac{\dfrac{\text{wrote}}{S\backslash NP/NP}\quad\dfrac{\text{Syntactic Structures}}{NP}}{S\backslash NP}>}{S}>\quad\dfrac{\text{but}}{CONJ_S}\quad\dfrac{\dfrac{\text{someone}}{S/(S\backslash NP)}\quad\dfrac{\dfrac{\text{did not}}{S\backslash NP/(S\backslash NP)}\quad\dfrac{\dfrac{\text{write}}{S\backslash NP/NP}\quad\dfrac{\text{Syntactic Structures}}{NP}}{S\backslash NP}>}{S\backslash NP}>}{S}>}{S}\langle\Phi\rangle$$

(119) Semantic composition of (84b)

$$\dfrac{\dfrac{\text{Someone}}{\lambda p.\begin{bmatrix} u : \begin{bmatrix} x : \mathsf{entity} \\ \mathbf{human}(x) \end{bmatrix} \\ p(\pi_1 u) \end{bmatrix}}\quad\dfrac{\dfrac{\text{wrote}}{\lambda y.\lambda x.\mathbf{write}(x,y)}\quad\dfrac{\text{Syntactic Structures}}{SS}}{\lambda x.\mathbf{write}(x,SS)}>}{\begin{bmatrix} u : \begin{bmatrix} x : \mathsf{entity} \\ \mathbf{human}(x) \end{bmatrix} \\ \mathbf{write}(\pi_1 u, SS) \end{bmatrix}}>$$

(120) Syntactic structure of (84b)

$$\dfrac{\dfrac{\text{someone}}{\lambda p.\begin{bmatrix} u : \begin{bmatrix} x : \mathsf{entity} \\ \mathbf{human}(x) \end{bmatrix} \\ p(\pi_1 u) \end{bmatrix}}\quad\dfrac{\dfrac{\text{did not}}{\lambda p.\lambda x.\neg p(x)}\quad\dfrac{\dfrac{\text{write}}{\lambda y.\lambda x.\mathbf{write}(x,y)}\quad\dfrac{\text{Syntactic Structures}}{SS}}{\lambda x.\mathbf{write}(x,SS)}>}{\lambda x.\neg\mathbf{write}(x,SS)}>}{\begin{bmatrix} u : \begin{bmatrix} x : \mathsf{entity} \\ \mathbf{human}(x) \end{bmatrix} \\ \neg\mathbf{write}(\pi_1 u, SS) \end{bmatrix}}>$$

Unlike (84a), this representation does not lead to a contradiction by itself.

16.3.4 Summary of the Puzzles

We have already shown that modifier puzzles and LNC puzzles are unavoidable as long as the semantic representation of quantificational expressions is of type entity. On the other hand, these puzzles are not necessarily solved by setting the syntactic type of the quantificational expression to $S/(S\backslash NP)$.

For example, suppose that the lexical items *everyone*, *someone* and *no one* are defined as follows (but suppose that *everyOne*, *someOne*, *noOne* are all of type entity), then these semantic representations have type $(\mathsf{entity} \to \mathsf{type}) \to \mathsf{type}$ and are semantically felicitous.

(121) a. $[\![\text{everyone} \vdash S/(S\backslash NP)]\!] \overset{def}{\equiv} \lambda p.p(everyOne)$

b. $[\![\text{someone} \vdash S/(S\backslash NP)]\!] \overset{def}{\equiv} \lambda p.p(someOne)$

c. $[\![\text{no one} \vdash S/(S\backslash NP)]\!] \overset{def}{\equiv} \lambda p.p(noOne)$

However, it can be seen that modifier puzzles, and LNC puzzles all occur under these lexical items. Therefore, setting the syntactic type $S/(S\backslash NP)$ is not in itself a sufficient condition for the resolution of puzzles.

A key point in the resolution of the puzzules in the previous section is that quantification can take a scope above the negation contained in the verb phrase. This is made possible by the fact that it is a function that takes a verb phrase (containing a negation) as an argument and then forms a proposition that uses it as a subterm. In this respect, it is made possible by setting the syntactic type of quantificational expressions to $S/(S\backslash NP)$.

16.4 Type-mismatch Problem

However, the analysis of quantificational expressions up to this point causes a problem when the quantificational expression appears in non-subject position, as in (122).

(122) John offended every linguist.

If the syntactic type of *every linguist* is $S/(S\backslash NP)$, then it is inconsistent in the object position of a transitive verb as in (123).

(123) Syntactic structure of (122)

$$\dfrac{\dfrac{\text{John}}{NP} \quad \dfrac{\dfrac{\text{offended}}{S\backslash NP/NP} \quad \dfrac{\dfrac{\text{every}}{S/(S\backslash NP)/N} \quad \dfrac{\text{linguist}}{N}}{S/(S\backslash NP)}{>}}{S\backslash NP}{??}}{S}$$

It was concluded in the last section that quantifiers have the syntactic type $S/(S\backslash NP)$ when they appear in the subject position, but that argument does not apply to quantifiers appearing in non-subject positions. This problem is known as the *type mismatch problem* for quantifiers.

The syntactic structure (123) suggests that the syntactic type of the transitive object position *every linguist* is $S\backslash NP\backslash(S\backslash NP/NP)$ and the syntactic structure is as follows.

(124) Syntactic structure of (122)

$$\dfrac{\dfrac{\text{John}}{NP} \quad \dfrac{\dfrac{\text{offended}}{S\backslash NP/NP} \quad \dfrac{\dfrac{\text{every}}{S\backslash NP\backslash(S\backslash NP/NP)/N} \quad \dfrac{\text{linguist}}{N}}{S\backslash NP\backslash(S\backslash NP/NP)}{>}}{S\backslash NP}{<}}{S}{<}$$

Therefore, the type of semantic representation of *every linguist* is

$$\begin{aligned} &\lceil S\backslash NP/(S\backslash NP/NP)\rceil \\ &= (\text{entity} \to \text{entity} \to \text{type}) \to \text{entity} \to \text{type} \end{aligned}$$

and accordingly, the semantic representation of *every linguist* also needs to be revised as follows.

(125) $\lambda p.\lambda x.\left(u : \begin{bmatrix} x : \text{entity} \\ \mathbf{linguist}(x) \end{bmatrix}\right) \to p(\pi_1 u)x$

However, “non-subject positions” are not limited to the object position of transitive verbs, but also includes the position of the direct object of ditransitive verbs, as in (126), in which case the syntactic type of *every* is as in (**??**).

(126) John gave every book to Mary.

(127) Syntactic structure of (126)

$$
\dfrac{\dfrac{\text{John}}{NP} \quad \dfrac{\dfrac{\dfrac{\text{gave}}{S\backslash NP/PP_{to}/NP} \quad \dfrac{\dfrac{\text{every}}{S\backslash NP/PP_{to}\backslash(S\backslash NP/PP_{to}/NP)/N} \quad \dfrac{\text{book}}{N}}{S\backslash NP/PP_{to}\backslash(S\backslash NP/PP_{to}/NP)}{>}}{S\backslash NP/PP_{to}}{<} \quad \dfrac{\dfrac{\text{to}}{PP_{to}/NP} \quad \dfrac{\text{Mary}}{NP}}{PP_{to}}{>}}{S\backslash NP}{>}}{S}{<}
$$

Therefore, the type of semantic representation of *every book* is

$$\lceil S\backslash NP/PP_{to}\backslash(S\backslash NP/PP_{to}/NP)\rceil = (\mathsf{entity} \to \mathsf{entity} \to \mathsf{entity} \to \mathsf{type}) \to \mathsf{entity} \to \mathsf{entity} \to \mathsf{type}$$

and the associated semantic representation of the *every book* must be as follows.

(128) $\lambda p.\lambda y.\lambda x.\left(u : \begin{bmatrix} x : \mathsf{entity} \\ \mathbf{book}(x) \end{bmatrix}\right) \to p(\pi_1 u)yx$

In both (125) and (128), apart from the term being quantified, there are variables for arguments the verb takes after that term (x in (125), y, x in (128)), which is given to the variable p representing the predicate, which is technically the key point.

At first glance, the observation above gives the impression that words such as *every* must be given separate lexical entries depending on their position of occurrence, but it is possible to give a unified semantic representation for each quantificational expression in a way that generalises the position of its occurrence, by defining the following Quantifier-Shift Operator, recursively.

Definition 351 (Quantifier-Shift Operator)**.** For any quantifier Q and S-reducible category $\boldsymbol{T}$, $|Q|^q_{\boldsymbol{T}}$ is recursively defined as follows:

$$
\begin{aligned}
|Q|^q_S &\overset{def}{\equiv} Q \\
|Q|^q_{Y|X} &\overset{def}{\equiv} \lambda p.\lambda x.|Q|^q_Y(\lambda y.(py)x)
\end{aligned}
$$

The quantifier Q taken by the quantifier-shift operator $|-|^q_{\boldsymbol{T}}$ is assumed to have type $(\mathsf{entity} \to \mathsf{type}) \to \mathsf{type}$ (but the definition above is a preterm definition. The requirements for the type of Q are described in refthQuantifierTypes).

$|Q|^q_{\boldsymbol{T}}$ is defined recursively by the structure of the syntactic type $\boldsymbol{T}$, which is an S-reducible category, and returns the quantifier Q as it is when $\boldsymbol{T}$ is a syntactic type S. When $\boldsymbol{T}$ is of the form $Y|X$, we recursively invoke $|Q|^q_Y$. Since Y is an S-reducible category, $|Q|^q_Y$ is defined. Also, as a whole, a new argument X is added and the original p is replaced by $\lambda y.(py)x$. The $|Q|^q_{\boldsymbol{T}}$ has the following types according to the syntactic type $\boldsymbol{T}$.

Theorem 352. For any S-reducible category $\boldsymbol{T}$,

$$Q : (\mathsf{entity} \to \mathsf{type}) \to \mathsf{type} \vdash |Q|^q_{\boldsymbol{T}} : (\mathsf{entity} \to \lceil \boldsymbol{T} \rceil) \to \lceil \boldsymbol{T} \rceil$$

Proof. By induction on the length of $\boldsymbol{T}$. For length 1, it is obvious because $\boldsymbol{T} \equiv S$, $|Q|^q_S \equiv Q$ and $\lceil S \rceil = \mathsf{type}$.

For lengths greater than 2, we can assume $\boldsymbol{T} \equiv Y|X$ (but Y is S-reducible). Let the induction hypothesis (IH) be that

$$Q : (\mathsf{entity} \to \mathsf{type}) \to \mathsf{type} \vdash |Q|^q_Y : (\mathsf{entity} \to \lceil Y \rceil) \to \lceil Y \rceil$$

A holds for Y, then

$$\dfrac{\dfrac{\dfrac{\dfrac{Q : (\mathsf{entity} \to \mathsf{type}) \to \mathsf{type}}{|Q|^q_Y : (\mathsf{entity} \to \lceil Y \rceil) \to \lceil Y \rceil}\,{\scriptstyle (IH)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{p : \mathsf{entity} \to \lceil X \rceil \to \lceil Y \rceil}{}^{3} \quad \dfrac{}{y : \mathsf{entity}}{}^{1}}{py : \lceil X \rceil \to \lceil Y \rceil}\,{\scriptstyle (\Pi E)} \quad \dfrac{}{x : \lceil X \rceil}{}^{2}}{(py)x : \lceil Y \rceil}\,{\scriptstyle (\Pi E)}}{\lambda y.(py)x : \mathsf{entity} \to \lceil Y \rceil}\,{\scriptstyle (\Pi I),1}}{|Q|^q_Y(\lambda y.(py)x) : \lceil Y \rceil}\,{\scriptstyle (\Pi E)}}{\lambda x.|Q|^q_Y(\lambda y.(py)x) : \lceil X \rceil \to \lceil Y \rceil}\,{\scriptstyle (\Pi I),2}}{|Q|^q_{Y|X} \;\equiv\; \lambda p.\lambda x.|Q|^q_Y(\lambda y.(py)x) : (\mathsf{entity} \to \lceil X \rceil \to \lceil Y \rceil) \to \lceil X \rceil \to \lceil Y \rceil}\,{\scriptstyle (\Pi I),3}$$

holds, and also $\lceil Y|X \rceil = \lceil X \rceil \to \lceil Y \rceil$, the following is proved.

$$Q : (\mathsf{entity} \to \mathsf{type}) \to \mathsf{type} \vdash |Q|^q_{Y|X} : (\mathsf{entity} \to \lceil Y|X \rceil) \to \lceil Y|X \rceil$$

□

By using the quantifier-shift operator $|-|^q_{\boldsymbol{T}}$, it is possible to define polymorphic quantifiers such as *every* in a unified way as follows.

(129)
$$\begin{array}{cl} & [\![\text{every} \vdash \boldsymbol{T}/(\boldsymbol{T}\backslash NP_{nom})/N]\!] \\ = & [\![\text{every} \vdash \boldsymbol{T}\backslash(\boldsymbol{T}/NP_{acc})/N]\!] \\ \stackrel{def}{\equiv} & \lambda n.\left|\lambda p.\left(u : \begin{bmatrix} x : \mathsf{entity} \\ n(x) \end{bmatrix}\right) \to p(\pi_1 u)\right|^q_{\boldsymbol{T}} \end{array}$$

Theorem 353. Lexical items in *every* satisfy the semantics felicity condition.

Proof. $\lceil \boldsymbol{T}/(\boldsymbol{T}\backslash NP_{nom})/N \rceil = \lceil \boldsymbol{T}\backslash(\boldsymbol{T}/NP_{acc})/N \rceil = \lceil N \rceil \to (\lceil NP \rceil \to \lceil \boldsymbol{T} \rceil) \to \lceil \boldsymbol{T} \rceil = (\mathsf{entity} \to \mathsf{type}) \to (\mathsf{entity} \to \lceil \boldsymbol{T} \rceil) \to \lceil \boldsymbol{T} \rceil$ so it is sufficient to show the following.

$$\dfrac{\dfrac{\dfrac{}{\lambda p.\left(u : \begin{bmatrix} x : \mathsf{entity} \\ n(x) \end{bmatrix}\right) \to p(\pi_1 u) : (\mathsf{entity} \to \mathsf{type}) \to \mathsf{type}}\,{\scriptstyle (Theorem\ 349)}}{\left|\lambda p.\left(u : \begin{bmatrix} x : \mathsf{entity} \\ n(x) \end{bmatrix}\right) \to p(\pi_1 u)\right|^q_{\boldsymbol{T}} : (\mathsf{entity} \to \lceil \boldsymbol{T} \rceil) \to \lceil \boldsymbol{T} \rceil}\,{\scriptstyle (Theorem\ 352)}}{\lambda n.\left|\lambda p.\left(u : \begin{bmatrix} x : \mathsf{entity} \\ n(x) \end{bmatrix}\right) \to p(\pi_1 u)\right|^q_{\boldsymbol{T}} : (\mathsf{entity} \to \mathsf{type}) \to (\mathsf{entity} \to \lceil \boldsymbol{T} \rceil) \to \lceil \boldsymbol{T} \rceil}$$

$\square$

Example 354. The semantic representation of *every* when $\boldsymbol{T}$ is S,$S\backslash NP$,$S\backslash NP/NP$ is calculated as follows, respectively.

$$
\begin{aligned}
& [\![\text{every} \vdash S/(S\backslash NP)/N]\!] \\
\equiv\ & \lambda n.|\lambda p.\,(u:(x:\mathsf{entity})\times n(x)) \to p(\pi_1 u)|^q_S \\
\twoheadrightarrow_\beta\ & \lambda n.\lambda p.\,(u:(x:\mathsf{entity})\times n(x)) \to p(\pi_1 u)
\end{aligned}
$$

$$
\begin{aligned}
& [\![\text{every} \vdash S\backslash NP/(S\backslash NP/NP)/N]\!] \\
\equiv\ & \lambda n.|\lambda p.\,(u:(x:\mathsf{entity})\times n(x)) \to p(\pi_1 u)|^q_{S\backslash NP} \\
\equiv\ & \lambda n.(\lambda p.\lambda x.|\lambda p.\,(u:(x:\mathsf{entity})\times n(x)) \to p(\pi_1 u)|^q_S(\lambda y.pyx)) \\
\twoheadrightarrow_\beta\ & \lambda n.\lambda p.\lambda x.(\lambda p.\,(u:(x:\mathsf{entity})\times n(x)) \to p(\pi_1 u))(\lambda y.pyx) \\
\twoheadrightarrow_\beta\ & \lambda n.\lambda p.\lambda x.\,(u:(x:\mathsf{entity})\times n(x)) \to (\lambda y.pyx)(\pi_1 u) \\
\twoheadrightarrow_\beta\ & \lambda n.\lambda p.\lambda x.\,(u:(x:\mathsf{entity})\times n(x)) \to p(\pi_1 u)x
\end{aligned}
$$

$$
\begin{aligned}
& [\![\text{every} \vdash S\backslash NP/NP/(S\backslash NP/NP/NP)/N]\!] \\
\equiv\ & \lambda n.|\lambda p.\,(u:(x:\mathsf{entity})\times n(x)) \to p(\pi_1 u)|^q_{S\backslash NP/NP} \\
\equiv\ & \lambda n.(\lambda p.\lambda y.|\lambda p.\,(u:(x:\mathsf{entity})\times n(x)) \to p(\pi_1 u)|^q_{S\backslash NP}(\lambda z.pzy)) \\
\twoheadrightarrow_\beta\ & \lambda n.\lambda p.\lambda y.(\lambda p.\lambda x.\,(u:(x:\mathsf{entity})\times n(x)) \to p(\pi_1 u)x)(\lambda z.pzy) \\
\twoheadrightarrow_\beta\ & \lambda n.\lambda p.\lambda y.\lambda x.\,(u:(x:\mathsf{entity})\times n(x)) \to (\lambda z.pzy)(\pi_1 u)x \\
\twoheadrightarrow_\beta\ & \lambda n.\lambda p.\lambda y.\lambda x.\,(u:(x:\mathsf{entity})\times n(x)) \to p(\pi_1 u)yx)
\end{aligned}
$$

The advantage of using the quantifier-shift operator is that other quantifiers can be defined in exactly the same way.

(130)
$$
\begin{aligned}
& [\![\text{a,an,some} \vdash \boldsymbol{T}/(\boldsymbol{T}\backslash NP_{nom})/N]\!] \\
=\ & [\![\text{a,an,some} \vdash \boldsymbol{T}\backslash(\boldsymbol{T}/NP_{acc})/N]\!] \\
=\ & \lambda n.\left|\lambda p.\begin{bmatrix} u:\begin{bmatrix} x:\mathsf{entity} \\ n(x) \end{bmatrix} \\ p(\pi_1 u) \end{bmatrix}\right|^q_{\boldsymbol{T}}
\end{aligned}
$$

(131)
$$
\begin{aligned}
& [\![\text{no} \vdash \boldsymbol{T}/(\boldsymbol{T}\backslash NP_{nom})/N]\!] \\
=\ & [\![\text{no} \vdash \boldsymbol{T}\backslash(\boldsymbol{T}/NP_{acc})/N]\!] \\
=\ & \lambda n.\left|\lambda p.\left(u:\begin{bmatrix} x:\mathsf{entity} \\ n(x) \end{bmatrix}\right) \to \neg p(\pi_1 u)\right|^q_{\boldsymbol{T}}
\end{aligned}
$$

(132)
$$
\begin{aligned}
& [\![\text{not every} \vdash \boldsymbol{T}/(\boldsymbol{T}\backslash NP_{nom})/N]\!] \\
=\ & [\![\text{not every} \vdash \boldsymbol{T}\backslash(\boldsymbol{T}/NP_{acc})/N]\!] \\
=\ & \lambda n.\left|\lambda p.\neg\left(u:\begin{bmatrix} x:\mathsf{entity} \\ n(x) \end{bmatrix}\right) \to p(\pi_1 u)\right|^q_{\boldsymbol{T}}
\end{aligned}
$$

For quantificational expressions in general, the following holds.

Theorem 355. Lexical items of quantificational expressions satisfy the semantics felicity condition.

Proof. In the same way as Theorem 353. □

By using lexical items of quantifiers employing the type-shift operator, (122) can be analysed as follows.

(133) Syntactic structure of (122)

$$\dfrac{\dfrac{\text{John}}{NP} \quad \dfrac{\dfrac{\text{offended}}{S\backslash NP/NP} \quad \dfrac{\dfrac{\dfrac{\text{every}}{\boldsymbol{T}\backslash(\boldsymbol{T}/NP)/N} \quad \dfrac{\text{linguist}}{N}}{\boldsymbol{T}\backslash(\boldsymbol{T}/NP)}>}{S\backslash NP\backslash(S\backslash NP/NP)}\forall E}{S\backslash NP}<}{S}<$$

(134) Semantic composition of (122)

$$\dfrac{\dfrac{\text{John}}{j} \quad \dfrac{\dfrac{\text{offended}}{\lambda y.\lambda x.\mathbf{offend}(x,y)} \quad \dfrac{\dfrac{\dfrac{\text{every}}{\lambda n.\left|\lambda p.\left(v:\begin{bmatrix} y:\text{entity} \\ n(y)\end{bmatrix}\right)\to p(\pi_1 v)\right|^{q}_{\boldsymbol{T}}} \quad \dfrac{\text{linguist}}{\mathbf{linguist}}}{\left|\lambda p.\left(v:\begin{bmatrix} y:\text{entity} \\ \mathbf{linguist}(y)\end{bmatrix}\right)\to p(\pi_1 v)\right|^{q}_{\boldsymbol{T}}}>}{\lambda p.\lambda x.\left(v:\begin{bmatrix} y:\text{entity} \\ \mathbf{linguist}(y)\end{bmatrix}\right)\to p(\pi_1 v)x}\forall E}{\lambda x.\left(v:\begin{bmatrix} x:\text{entity} \\ \mathbf{linguist}(x)\end{bmatrix}\right)\to \mathbf{offend}(x,\pi_1 v)}<}{\left(v:\begin{bmatrix} x:\text{entity} \\ \mathbf{linguist}(x)\end{bmatrix}\right)\to \mathbf{offend}(j,\pi_1 v)}<$$

16.5 Inverse Scope

However, the type-shifting analysis described in the previous section is known to cause problems in the derivation of the *inverse scope reading* described below.

It is known that sentences in which two quantifiers appear (here one in subject and one in object) have two different readings, called *canonical scope reasing* and *inverse scope reading*, as shown below.

(135) Some woman loves every Englishman.

First, the reading called canonical scope reading is the normal (i.e. canonical) reading discussed up to the previous section, i.e. the reading "a woman exists and she loves all English gentlemen."[*6] This is shown by the semantic representation in which the scope of the Σ-type introduced by *some woman* contains the scope of the Π-type introduced by *every Englishman*.

(136)
$$\begin{bmatrix} u:\begin{bmatrix} x:\text{entity} \\ \mathbf{W(x)}\end{bmatrix} \\ \left(v:\begin{bmatrix} y:\text{entity} \\ \mathbf{E}(y)\end{bmatrix}\right)\to \mathbf{L}(\pi_1 u,\pi_1 v)\end{bmatrix}$$

[*6] 別名、queen reading と呼ばれる。Queen はすべての英国紳士を愛しているからである。Also known as the *queen reading*. This is because the queen loved all English gentlemen.

In contrast, the reading known as inverse scope reading is that "there is a certain woman for each English man and she loves him."[*7]This is shown by the semantic representation in which the scope of Π-type introduced by *every Englishman* contains the scope of Σ-type introduced by *some woman*.

(137) $$\left(v : \begin{bmatrix} y : \mathsf{entity} \\ \mathbf{E(x)} \end{bmatrix}\right) \rightarrow \begin{bmatrix} u : \begin{bmatrix} x : \mathsf{entity} \\ \mathbf{W}(x) \end{bmatrix} \\ \mathbf{L}(\pi_1 u, \pi_1 v) \end{bmatrix}$$

The analysis by type shifting described in the previous section does not provide a way to derive a semantic representation equivalent to inverse scope reading for sentences like (135). A solution to this problem must simultaneously solve the type mismatch problem in Section **??**. This is provided in the next chapter.

Remark 356. We use the notation $\exists > \forall$ for the canonical scope reading and $\forall > \exists$ for the inverse scope reading of (135). This means that the existential quantification represented by *some* is $\exists$ and the universal quantification represented by *every* is $\forall$, and by $\exists > \forall$ we indicate that the scope of *every* is included within the scope of *some* (and vice versa for $\forall > \exists$).

Remark 357. We need to discuss whether there are really two readings in (135), i.e. whether canonical scope reading and inverse scope reading in (135) are independent readings. In a situation where there are only John, Bill, Susan and Mary (by defining entityas in (33)), assume the following context: John and Bill are English man, Susan and Mary are not. Susan and Mary are women, and John and Bill are not. Further, we assume that John loves Susan and Bill loves Mary, but no other combination. Then, (135) is true in its inverse scope reading (= for every English man there exists a woman who loves him), but not in its canonical scope reading (= there is a woman who loves every English man).

However, conversely, in situations where the canonical scope reading of (135) is true, the inverse scope reading is always true. This is because (136) entails (137).

One question arises here: Is inverse scope reading alone sufficient as the semantic representation of (135)? In other words, isn't (137) the only semantic representation of (135), and canonical scope reading is only a special case of a situation in which (137) is true?

The answer to this question is no. Consider the following sentence.

(138) Every Englishman loves some woman.

(138) is also said to have two readings, canonical and inverse scope reading.[*8]The canonical scope reading is "there exists a woman who seems to love for all English men," while the inverse scope reading is "there exists a woman who loves for all English men" . In this example, contrary to (135), the inverse scope reading is true whenever the situation in which the canonical scope reading is true (but the reverse is not true). This is because the semantic representation for canonical scope reading

[*7] Also known as the *wife reading*.

[*8] Historically, the construction (138) preceded the construction (135) when the existence of inverse scope reading was recognised. Indefinites was considered more likely to allow inverse scope reading, and it was only afterwards that it became known that universal quantifiers could also induce inverse scope readings.

entails the semantic representation for inverse scope reading. Therefore, contrary to (138), one may consider that the semantic representation for the canonical scope reading is the only semantic representation, and inverse scope reading is only a special case. Considering both cases, it is natural to analyse sentences with two quantifiers as having both canonical and inverse scope readings.

Remark 358. More direct evidence includes the readings for the following sentence.

(139) Two arrows penetrated five targets.

In (139), in the situation where two arrows have each penetrated five targets (i.e. there are ten targets), canonical scope reading is true, but inverse scope reading is not (because it is not the caes that each target has two arrows penetrating it). Conversely, in a situation where 10 arrows exist, and each of the 5 pairs of two arrows hit each target, canonical scope reading is not true (because there are no arrows that penetrated the five targets), but only inverse scope readinng is true. Thus, by avoiding the use of existential quantifications such as *some* and indefinite as quantificational expressions, we can identify situations where one reading is true but the other is not, and this can be done for both readings. Therefore, canonical scope reading and inverse scope reading are independent. However, the analysis of quantifiers such as *two arrows* and *five targets* requires a more complex toolkit in DTT, so we will postpone their analysis and just use the universal/existential quantification example for the time being.

Remark 359. In Generative syntax, a common analysis is to derive both canonical/inverse scope readings by Quantifier Raising (QR). This can be seen as an analysis that attributes scope taking to $\bar{A}$-movement. See Rodman (1976), May (1977, 1985).

What has been agreed in recent years is that, first, the scope relation is not as free and constrained via QR predicts.[*9] Second, for indefinite, it is freer than QR, thus indefinites are considered to have an alternative way of scope taking such as unselective binding.[*10]

However, Hayashishita (1999, 2000, 2003) subsequently pointed out that each speaker may have a different set of quantificational expressions that allow inverse scope reading. In other words, it is not just that there are speakers who say that inverse scope can be taken for any quantificational expression, and speakers who say that inverse scope reading is not acceptable at all, but there are also substantial number of speakers who accept inverse scope reading for other quantificational expressions. Therefore, on the side of semantic theory, those that view canonical and inverse scope readings as symmetrical (as in QR) are unlikely to account for this fact, and it is reasonable to think that, deriving inverse reading requires a combination of some property marked in its lexical entiry and a special operation licenced by it.

[*9] Aoun and Li (1993); Szabolci (1997); Fox (2000); Bruening (2001)

[*10] Heim (1982), choice function Reinhart (1997a,b); Winter (1997); Kratzer (1998)