



Abney, Steven & Keshet, Ezra. 2025. Plural Intensional Presuppositional predicate calculus (PIP). *Glossa: a journal of general linguistics* 10(1). pp. 1–48. DOI: <https://doi.org/10.16995/glossa.16422>



Plural Intensional Presuppositional predicate calculus (PIP)

Steven Abney, University of Michigan, abney@umich.edu

Ezra Keshet, University of Michigan, ekeshet@umich.edu

One successful branch of natural language semantics uses first order predicate calculus and set theory, well-studied systems of mathematics in their own right, as meta-languages for natural-language meanings. Now this traditional approach suffers from certain well-known deficiencies in treating what we call *improper scope anaphora*, such as cross-sentential anaphora to indefinites, donkey pronouns (Geach 1962), and quantificational (Karttunen 1969) and modal (Roberts 1987) subordination. Dynamic plural logics, such as those proposed by van den Berg (1996) and Brasoveanu (2007), successfully solve the problems presented by improper scope phenomena, but at the cost of diverging sharply from the traditional predicate-calculus approach. This paper aims instead to present a treatment of improper scope that is accessible to the broad audience of linguists who are familiar chiefly with the traditional approach. Surprisingly, we require only two changes to standard logic with set theory: (i) a method to extend the scope of existentials, and (ii) a mechanism to store and retrieve subformulas. We implement these changes in a logic we call *Plural Intensional Presuppositional predicate calculus* (PIP). PIP also introduces (iii) a new operator for the sake of defining presuppositional felicity conditions (independent of truth conditions). We show that the resulting logic captures the full range of improper scope phenomena covered by state-of-the-art intensional dynamic plural logics. But PIP also provides a ready analysis for paycheck pronouns, which are difficult to capture in most dynamic logics, and presupposition projection, which has not received as much attention in dynamic plural logic.



1 Introduction

One **traditional approach** to natural language semantics involves a simple compositional system that assigns truth conditions to sentences, describing how the world must be when they are true. These truth conditions may be represented using first-order predicate calculus (Coppock & Champollion 2024), or some mix of natural language and other notations (Heim & Kratzer 1998); and some rudimentary set and lambda notation is also usually employed.

Under this traditional approach, DPs denote quantifiers: *a dog*, for instance, forms a true statement only when combined with a property that at least one dog has. Such quantifiers may bind pronouns, but only within their c-command domains. However, a well-known constellation of counterexamples, which we call *improper-scope anaphora*, involve DPs which seem to bind pronouns outside their c-command domains. These include:

- (1) a. *Cross-sentential anaphora*
A dog sauntered in. It sat down and barked.
- b. *Summation pronouns*
Most students₁ wrote a paper₂. They₁ left them₂ on my desk.
- c. *Paycheck Pronouns* (Karttunen 1969; Jacobson 2000)
The employee who saved her paycheck was wiser than the one who cashed it.
- d. *Donkey pronouns* (Geach 1962)
Every farmer who owns a donkey pets it.
- e. *Quantificational Subordination* (Karttunen 1969; Sells 1985)
Most students wrote a paper. Some of them even turned it in.
- f. *Modal Subordination* (Roberts 1987)
A wolf might come in. It would eat you first.

This paper aims to present a fully formal treatment of improper scope anaphora that is accessible to the broad audience of linguists who are more familiar with the traditional approach to semantics than with alternatives such as dynamic semantics.

There are two major solutions for improper scope, and both treat improper-scope cases uniformly, as a natural class. The **E-type approach** (following Evans 1977) follows the traditional approach closely, but relaxes the c-command requirement somewhat. In these improper cases, a predicate previously made salient may be reused in a new location:

- (2) A dog entered. It [=the dog] barked.

Although largely successful, the E-type approach suffers from a number of well-known problems (Elbourne 2005), including the undefinedness of the key term “salience.” The **dynamic approach**,

conversely, maintains a contiguous syntactic scope for all anaphora by extending a DP's scope to the right, beyond its original c-command domain; but this comes at the cost of breaking sharply from the traditional approach (van den Berg 1996; Brasoveanu 2007). The dynamic approach has the advantage of being fully formal; it also handles cross-sentential and donkey anaphora well. Phenomena like subordination, though, require complex machinery for scope extension, because there are regions between the original c-command scope and the new "subordinate" scope, where a pronoun is not in fact bound:

- (3) Most ENG 101 students wrote an essay [=e] on a poem we read in class. It [$\neq e$] was very well written. They each left it [=e] on my desk after class.

The infelicity of the second sentence with *it* denoting *e* contrasts with an alternative in which the second sentence is instead *They each wrote it [=e] about the rhyme scheme*.

In contradistinction to both the E-type and dynamic approaches, we reject the assumption that improper-scope cases constitute a natural class. Rather, we propose that cross-sentential anaphora forms a class with the traditional proper-scope cases; we call this the class of *syntactic scope anaphora*. Briefly, we follow Heim (1982: Ch 2) in assuming that indefinites, like pronouns, are bound variables. Their "scope" is defined by the next higher quantifier; in cases of cross-sentential anaphora, this quantifier is a discourse-level existential closure. All other cases of improper scope form a class we call *resumptive scope anaphora*; these are cases in which a previously closed scope is resumed in a new context, as in example (3).

This new way of dividing up the empirical landscape allows us to present a system that, like E-type systems, should be readily understandable to our target audience, but, like dynamic systems, is fully formal. In fact, our core system adopts the logic of the traditional approach as it stands (namely, Zermelo-Fraenkel set theory), with the addition of only three defined notations:

- The vertical bar ('|') operator introduces presuppositions, and affects felicity only, not truth.
- Square brackets ('[]') mark the variable introduced by an indefinite, so it can take its scope from the next higher quantifier.
- Formula labels (uppercase letters) introduce notation to repeat previously occurring material, akin to ellipsis or E-type pronouns.

These new notations are definitions of convenience, and can be mechanically translated to standard set theory, and in that sense do not change what *can* be expressed, but only what operations are basic enough in natural language that one should be able to express them *concisely*. Because of this, the resulting semantics, which we name *Plural Intensional Presuppositional predicate*

calculus (PIP), is particularly amenable to a simple Heim & Kratzer-style system of compositional interpretation.¹

We show that the resulting logic captures essentially all the improper-scope examples that intensional dynamic plural logics handle. But PIP also provides a ready analysis for paycheck pronouns, which are difficult to capture in most dynamic logics, and presupposition projection, which is not part of any plural dynamic logic that we are aware of.²

We believe that this paper makes an empirical contribution. In addition to the broad claim that improper-anaphora cases do not constitute a natural class, we also make some new observations: we detail how plural pronouns refer back to weak donkey pronouns in a previous sentence; we examine quantificational subordination of pronouns in the restriction, rather than the nuclear scope, of a subordinate quantifier; and we reveal the simple interplay between presupposition projection and quantificational subordination.

Nonetheless, the main point is theoretical: we aim to analyze all these phenomena—new and old—using a bare minimum of theoretical apparatus, within a system that is familiar to a broader audience.

In the remainder of the paper, we first define the full PIP logic (§2) and give a compositional translation from natural language expressions to PIP (§3). We then show how PIP applies to improper scope phenomena (§4), and we close by discussing areas for future work (§5).

2 PIP

Semantic theories often make use of sets, a prime example being Barwise and Cooper’s (1981) definition of generalized quantifiers as relations over sets:

- (4) Half the dogs barked $\rightsquigarrow |\{x : \text{DOG}(x) \ \& \ \text{BARKED}(x)\}| = \frac{1}{2} |\{x : \text{DOG}(x)\}|$
‘The set of dogs who barked is half the size of the set of all dogs’

¹ Our proposal is similar in spirit to recent work in Mandelkern (2022), which employs a dual system: pure first-order logic, supplemented with an auxiliary system for licensing definite descriptions and pronouns. The classical portion of Mandelkern’s system maintains the well-understood properties of classical logic, such as double negation elimination. The supplement captures traditionally “dynamic” phenomena in singular logics, such as donkey anaphora and cross-sentential anaphora. In the same way, PIP maintains a classical base logic, with supplements to capture improper scope.

² The focus of this paper is the treatment of improper scope phenomena. As pointed out by an anonymous reviewer, there are several other phenomena that plural dynamic semantic systems address. For instance: dependent and wide-scope indefinites (Brasoveanu & Farkas 2011; Henderson 2014; DeVries 2016; Kuhn 2017), reciprocals (Murray 2008; Dotlačil 2013), and various readings of questions (Dotlačil & Roelofsen 2020; Roelofsen & Dotlačil 2023). A general technical comparison of PIP to dynamic systems is not the purpose of this paper, and we will leave the full analysis of such phenomena in (systems extending) PIP to future work. We will mention, however, that there is good reason for optimism that analyses comparable to the dynamic analyses will be available in PIP. Namely, a formula label in PIP contains at least as much information as a plural information state used in plural logics, as detailed in footnote 9.

(See also Montague 1973 and Heim & Kratzer 1998, among others.) Sets are likewise integral to PIP, and we adopt the clear consensus system for representing sets in mathematics, namely, Zermelo-Fraenkel set theory (ZF).³ ZF is an extension of standard predicate logic, which itself has been a mainstay of semantic theories since at least Russell (1905). As such, ZF already includes the standard elements of first-order logic: the operators negation, conjunction, disjunction, implication, and equivalence ($\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$), the two first-order quantifiers ($\exists$, $\forall$), and the equality operator ($=$). To these operations, ZF adds the set-membership operator ($\in$) and provides formal definitions for other important set **constructions**, such as set abstraction ($\{x : \dots\}$), subset ($\subseteq$), union ($\cup$), and intersection ($\cap$), among others. Finally, instead of a domain of individuals, as is usually assumed in a singular semantics, ZF assumes a domain of sets, which form the arguments for predicates.

PIP extends ZF, and thus adopts sets as its domain. Ontologically, PIP distinguishes four varieties of set:⁴

- (5) a. **singulars**, singleton sets that represent individual entities,
- b. **plurals**, which represent groups of entities,
- c. **worlds**, singleton sets that represent possible worlds,
- d. **propositions**, sets representing collections of possible worlds.

Every lexical predicate in PIP takes a world as its first argument; any subsequent arguments may be singulars or plurals but not other sets (for instance, not the empty set). To avoid clutter, we will often write the first (world) argument of a predicate as a subscript, or omit it altogether, if the world is not material to the point under discussion. Thus, the predicate DOG may appear as

- (6) DOG(w, x), DOG _{w} (x), or simply DOG(x).

PIP extends ZF by adding three new defined constructions:

- (7) a. Summation and local variables: $\Sigma x(\dots[y]\dots)$
- b. Formula labels (definition and use): $(X \equiv \phi) \dots X$
- c. Presuppositions: $\phi \mid \psi$

The remainder of this section will introduce and motivate the new PIP constructions, including a section on using the definitions of these new constructions to expand out expressions containing them into standard ZF, much as ZF expands out constructions such as set abstraction into the predicate calculus.

³ See Jech (2002) for a standard formulation of ZF.

⁴ These varieties do not partition the domain. For example, the empty set belongs to the domain, but not to any of these varieties. See §2.4 for a more detailed presentation.

2.1 Syntactic scope anaphora

One major motivation for the dynamic turn in semantics is the problem of cross-sentential anaphora to indefinites. To wit, the most straightforward way for the traditional approach to treat an indefinite like *a dog* in (8) is as an existential quantifier in first order logic:

$$(8) \quad [\text{DP } a \text{ } [\text{NP } \text{dog}]] [\text{VP } \text{appeared}] \rightsquigarrow \exists d (\text{DOG}_w(d) \wedge \text{APPEARED}_w(d))$$

The logical quantifier $\exists$ scopes over the conjunction of the indefinite's NP and its c-command domain, here the VP. Next, the most straightforward translation for two adjacent sentences is as a conjunction:

$$(9) \quad \text{It's raining. It's windy, too.} \rightsquigarrow \text{RAINING}_w() \wedge \text{WINDY}_w()$$

But these two techniques, when combined, give the wrong result for a discourse with an indefinite in one sentence and a coreferent pronoun in the next, as shown in (10a). Instead, the indefinite seems to scope over both the sentences, as shown in (10b):

$$(10) \quad \text{A dog appeared. It barked.}$$

- a. $\not\rightsquigarrow \exists d (\text{DOG}_w(d) \wedge \text{APPEARED}_w(d)) \wedge \text{BARKED}_w(d)$
- b. $\rightsquigarrow \exists d (\text{DOG}_w(d) \wedge \text{APPEARED}_w(d) \wedge \text{BARKED}_w(d))$

The problem with (10a) is that the variable d in $\text{BARKED}_w(d)$ is outside the scope of the logical quantifier $\exists d$, and therefore it does not represent a dog that appeared. Instead, to capture the two sentences correctly, $\exists d$ must scope over the translations for both sentences.

One way of resolving this paradox is to move to a dynamic logic, and assume that the first sentence of (10) introduces a new **discourse referent** which may be retrieved by a pronoun in a later sentence (Kamp 1981; Heim 1982: Ch 3; Groenendijk & Stokhof 1991). Doing so represents a rather drastic shift in one's logical foundation, though: dynamic logics were designed for programming languages, where formulas are commands that can store and retrieve information, rather than statements that are simply true or false.

PIP instead pursues a more conservative alternative, grounded in a proposal also due to Heim (1982), but Chapter 2 rather than Chapter 3. Heim, following work by Lewis (1975), represents indefinite noun phrases as simple variables rather than quantifiers; a sentence containing an indefinite is thus semantically an open formula (that is, a formula with free variables). The variables introduced by indefinites (and *only* these variables) are then bound by the next higher quantifier in the structure. These quantifiers are known as **unselective quantifiers** since they bind multiple variables at once.⁵ And one such quantifier is a silent **discourse closure** operator at

⁵ Lewis introduced the term *unselective quantifier* for a quantifier that binds all free variables in its scope. The quantifiers we adopt, following Heim, could more accurately be called *semiselective*.

the top of the entire discourse, existentially binding any indefinite variables that are not bound by lower quantifiers. And when the same quantifier also binds pronouns coreferent with an indefinite, we call this connection *syntactic scope anaphora*.

We implement this approach in PIP by translating indefinites as distinguished variables, notated within brackets:

$$(11) \quad \text{DOG}([d]) \wedge \text{APPEARED}(d) \quad [\text{PIP}]$$

These brackets mark certain free variables in a given scope as *local* variables, in contrast to unbracketed free variables, which we call *external* variables. (The local/external distinction exists in Lewis and Heim, though their notation and terminology differ from ours.)

Unselective closure in PIP is provided by the **summation** operator ‘ Σ ,’ which existentially binds local variables in its scope, but not external variables. ‘ Σ ’ is a shorthand for the generalized union⁶ of a set abstraction, with bracketed variables in its scope existentially bound:

$$(12) \quad \Sigma x(\dots[y]\dots) \rightsquigarrow \bigcup \{x : \exists y(\dots y \dots)\}$$

We view this construction as merely an alternate way to indicate which variables are bound by an existential quantifier: the Σ marks the scope of quantification, and the bound variables are marked *in situ* with brackets.

As the summation operator performs set abstraction, the proposition expressed by a discourse may be obtained, in PIP, by summing over the world variable, as shown in (13):

$$(13) \quad \begin{array}{ll} \text{It's raining. It's windy, too.} & \\ \text{a. } \Sigma w(\text{RAINING}_w() \wedge \text{WINDY}_w()) & [\text{PIP}] \\ \text{b. } \bigcup \{w : \text{RAINING}(w) \wedge \text{WINDY}(w)\} & [\text{ZF}] \end{array}$$

(13a–b) are the sets of worlds where it is raining and windy, a common way of representing the proposition expressed by the discourse. And, at the same time, the discourse-level summation illustrated in (13a) provides the discourse-level existential closure that Heim proposed:⁷

$$(14) \quad \begin{array}{ll} \text{a. } \Sigma w(\text{DOG}_w([d]) \wedge \text{APPEARED}_w(d)) & [\text{PIP}] \\ \text{b. } \bigcup \{w : \exists d(\text{DOG}(w, d) \wedge \text{APPEARED}(w, d))\} & [\text{ZF}] \end{array}$$

Returning now to the paradox of (10a) vs (10b), we can essentially distinguish between the *sentence* “a dog appeared” and the one-sentence *discourse* “a dog appeared.” We take the meaning

⁶ Since variables in ZF and PIP represent sets already, this union operation “flattens” the component sets into one new set.

⁷ Note that in PIP expressions, truth-functional equivalence does not guarantee intersubstitutability. For example, $\text{DOG}_w([d])$ and $\text{DOG}_w(d)$ are truth-functionally equivalent, but if we replace the former with the latter in (14a), we change the meaning. See also Section 2.4.5.

of the sentence to be the open formula (11), whereas the meaning of the discourse is (14), in which local variables are existentially closed by Σw . This approach captures the same intuition that motivates dynamic logic, namely, that the scope of the variable is wider than the sentence, without abandoning the simplicity of first order logic. (See also Cresswell 2002 for more discussion on this issue.)

2.2 Resumptive scope anaphora

Donkey anaphora provides a second main motivation for dynamic logic, as illustrated by the translation of (15) into predicate logic:

(15) Every farmer who owns a donkey pets it. (Geach 1962)

$$\rightsquigarrow \forall f \forall d ((\text{FARMER}(w, f) \wedge \text{DONKEY}(w, d) \wedge \text{OWNS}(f, d)) \rightarrow \text{PETS}(f, d))$$

Notice that we again have an unusual scoping for the variable d , which here corresponds to *a donkey*: it seems to scope alongside the variable f representing the farmers, at the top of the sentence. (Moreover, the indefinite appears to translate—atypically—as the universal logical quantifier $\forall$ instead of the existential $\exists$.) Now, several dynamic techniques have been proposed to handle this phenomenon (Kamp 1981; Groenendijk & Stokhof 1991), but they do not extend to full generalized quantifiers. The PIP treatment, by contrast, assumes generalized quantifiers from the outset.

In a nutshell, PIP uses summation terms for the restriction and nuclear scope of a quantifier, allowing the quantifier meaning itself to be a simple relation between pluralities (Barwise & Cooper 1981):⁸

(16) Most dogs bark

- a. $\text{MOST}(\Sigma d \text{DOG}_w([d]), \Sigma d (\text{DOG}_w([d]) \wedge \text{BARKS}_w(d)))$ [PIP]
- b. $\text{MOST}(\bigcup \{d : \text{DOG}(w, d)\}, \bigcup \{d : \text{DOG}(w, d) \wedge \text{BARKS}(w, d)\})$ [ZF]

Notice that the conservativity of natural language determiners is reflected in the repetition of the restriction of the quantifier in its scope: the set of dogs is compared to the set of dogs that bark, not the set of all barkers. We follow Barwise & Cooper (1981) in assuming this to be a universal property of natural-language quantifiers; we will thus want a way to facilitate such repetitions of restriction formulas, no matter how complex they are.

PIP's formula labels serve this very purpose. A formula label is a symbol (conventionally, upper-case) that acts as a shorthand for a given formula (cf. the *update variables* of Keshet 2018). The formula in (17) defines X using the ' $\equiv$ ' operator, but asserts nothing; it is tautologically true:

⁸ Note that the variable of abstraction itself is not existentially bound within the scope of a summation operator, even if it is local. See Section 2.4 below for details.

$$(17) \quad X \equiv \phi$$

Once defined, the label can be used as a formula, and the effect is exactly the same as repeating the original formula where the label occurs.⁹ With formula labels, we can represent (16a) with less repeated material:

$$(18) \quad \text{MOST}(\Sigma d(D \wedge D \equiv \text{DOG}_w([d])), \Sigma d(B \wedge B \equiv (D \wedge \text{BARKS}_w(d))))$$

This formula is logically equivalent to (16a), with the formula label D enforcing conservativity (and Section 3.5 below analyzes this formula label as the consequence of quantifier raising).¹⁰

As tautologies, formula-label definitions can always be “floated out” and conjoined at the top level, with no effect on truth value. We generally do so, and to further improve readability, we use “where” as a synonym for conjunction. Thus we write for example (19), which is to be understood to mean (18):

$$(19) \quad \text{MOST}(\Sigma dD, \Sigma dB) \text{ where} \\ D \equiv \text{DOG}_w([d]) \\ B \equiv (D \wedge \text{BARKS}_w(d))$$

(19) defines two formula labels: the restriction D asserts that d is a dog, and the nuclear scope B asserts that d is a dog that barks (with “ d is a dog” being contributed by the label D). The main formula then asserts that the set of dogs that bark (ΣdB) comprises most of the set of dogs (ΣdD).

Now, the summation defining the restriction normally closes the scope of any indefinite within it. However, a formula label may effectively resume any indefinite’s scope later in the discourse, a phenomenon we call *resumptive scope anaphora*. Donkey anaphora is simple resumptive anaphora in the nuclear scope (some predicates abbreviated to save space):

$$(20) \quad \text{Most farmers who own a donkey pet it.} \\ \text{a. } \text{MOST}(\Sigma fR, \Sigma fS) \text{ where} \quad \text{[PIP]} \\ R \equiv (\text{FARMER}_w([f]) \wedge \text{DONKEY}_w([d]) \wedge \text{OWNS}_w(f, d)) \\ S \equiv (R \wedge \text{PETS}_w(f, d))$$

⁹ As mentioned in footnote 2, a formula label contains at least as much information as a plural information state (set of sets of assignments). To be precise, a formula label X containing local variables L in the context of assignment g implicitly defines the plural information state consisting of the set of assignments that verify X and differ from g only with respect to L :

$$(i) \quad \{h : g[L]h \ \& \ \llbracket X \rrbracket^h = 1\}.$$

¹⁰ To be clear, a formula label is not a variable. After (17), X does not denote the value of ϕ ; X represents ϕ itself. And when one uses X , one does not refer back to the value of ϕ , rather (intuitively speaking) one pastes ϕ into the subsequent expression. It is possible to give a denotational account of formula labels, but they are hyperintensional. X denotes, not the value of ϕ , but the function from contexts to values that ϕ represents. See Section 2.4.5 for details.

$$\text{b. } \text{MOST} \left(\begin{array}{l} \bigcup \{f: \exists d (F(w, f) \wedge D(w, d) \wedge O(w, f, d))\}, \\ \bigcup \{f: \exists d (F(w, f) \wedge D(w, d) \wedge O(w, f, d) \wedge P(w, f, d))\} \end{array} \right) \quad [\text{ZF}]$$

The repetition of the restriction label R inside the definition of the nuclear scope label S includes the bracketed ‘ $[d]$ ’ for the donkey. This repetition of bracketed ‘ $[d]$ ’ allows the summation operator in the nuclear scope, $\Sigma f S$, to existentially close d for both the indefinite *a donkey* and the subsequent donkey pronoun *it*, just as in the cross-sentential case.

Finally, quantifiers also motivate the distinction between local and external variables. Variables used within the scope of a summation, but not introduced by an indefinite there, are not bound by that summation. For instance, the world variable w in (19) is not existentially bound within the sentence. Otherwise, it could not be bound by Σw at the discourse level to form the proposition expressed by the discourse.

2.3 Felicity and presupposition

Natural-language sentences not only make assertions; they also have presuppositions. To capture these, PIP provides expressions of the form $\phi|\psi$, which assert ϕ and presuppose ψ . Our notation is similar to Blamey’s (1986) transplication operator,¹¹ and may be considered a compact variant of the horizontal-bar notation employed, for instance, by Sauerland (2005):

$$(21) \quad \frac{\phi}{\psi}$$

Unlike Blamey, though, we consider the presuppositions of a formula to be independent of its truth conditions: $\phi|\psi$ is true iff ϕ is true. That is, in PIP, a presupposition has no direct effect on truth; rather, it contributes to determining whether or not a discourse is *felicitous*. (Bracketed variables in a presupposition can have indirect effects on truth when they also appear outside the presupposition; we have not encountered such a situation, though.)

As a quick example, a pronoun such as *it* in (22) carries a presupposition of singularity, and this presupposition is reflected in the portion of the formula after the ‘|’:

$$(22) \quad \text{It}_x \text{ barked.} \rightsquigarrow \text{BARKED}(x|\text{SG}(x))$$

Now, in PIP, this formula is true if x barked, regardless of the cardinality of x ; but it is felicitous only if x is singular.

2.4 Formal details

We now present a more formal treatment of the notations introduced above. Readers less interested in these details may safely skip ahead to Section 3.

¹¹ Except that the transplication operator has the presupposition on the left. See also Beaver & Krahmer (2001).

2.4.1 Preliminaries

As mentioned earlier, because PIP is an extension of ZF, everything in the domain of discourse is a set: we eschew so-called *urelements* that have no members but are distinct from the empty set. We do, however, draw formal and ontological distinctions within the domain. First, we distinguish a class of *atoms*, which formally play a role like that of urelements. Namely, *pluralities* are defined to be those sets whose elements consist entirely of atoms. We partition the non-empty pluralities into e-pluralities (corresponding to semantic type *e*), and s-pluralities (corresponding to semantic type *s*). E-pluralities with cardinality one are called *singulars*, e-pluralities with cardinality greater than one are called *plurals*, and s-pluralities with cardinality one are called *worlds*. Plurals are technically not sets of singulars but rather unions of singulars. Unions of worlds are called *propositions*. The empty set is both an e-plurality and an s-plurality.

We assume that nonlogical constants name relations among pluralities, not relations among arbitrary sets. This is not a syntactic requirement, but rather, a constraint on models (and thus, essentially, an axiom). Namely, if a nonlogical n -place predicate is true of sets $x_1, \dots, x_n$, then each argument x_i is a plurality. For example, if $\text{DOG}(x)$ is true, then x is a plurality. PIP is not a modal logic; the context of evaluation does not include a world of evaluation. Rather, we assume that the first argument of each nonlogical constant is a world. To say that x is one or more dogs, we more precisely write $\text{DOG}(w, x)$, meaning that x is a dog-plurality in world w . Again, there is no syntactic requirement that the first argument be a world, but we assume that $\text{DOG}(w, x)$ is only true if w is a world.

Standardly, the meaning of an expression in ZF is defined via a mechanical procedure that converts any expression containing ZF constructions, such as set abstraction, into an equivalent predicate-calculus expression, containing only ‘ $\in$ ’ and the primitive symbols of predicate calculus. In the same way, we define expressions of PIP via a mechanical procedure to convert them into expressions of ZF proper.

We write the translation function as $\mathcal{T}^A$:

$$(23) \quad \mathcal{T}^A \phi = \psi.$$

For example:

$$(24) \quad \mathcal{T}^A(\Sigma_w X) = \bigcup \{w : \exists d \text{DOG}(d)\}, \quad \text{if } A(X) = (\text{DOG}([d]) | \text{SG}(d)).$$

The function A is called a **label assignment function**; it maps formula labels to their definitions. Note that the functions $\mathcal{T}$ and A are meta-language functions: their inputs and outputs are *expressions*, not the values that those expressions denote. The output of $\mathcal{T}$ is an expression of ZF, containing no PIP-specific constructions; it is truth-functionally equivalent to the input expression, but it does *not* capture the full meaning of the input expression: information about presuppositions, local variables, and formula definitions is lost.

2.4.2 Local variables

The translation procedure uses the concept of *local variables* of a PIP expression. Intuitively, the local variables of ϕ are all the bracketed variables that occur at the “top level” in ϕ , which is to say, not embedded within a Σ expression inside of ϕ . Formally, we provide a recursive definition for the local variables of ϕ , written $\mathcal{L}^A\phi$, relative to a label assignment function A . The definition of this function is shown in (25) and (26). First, for the PIP expressions in (25), (a) a bracketed variable introduces a local variable, while (b) summation erases any local variables below it (since the summation binds them). Next, (c) formula label definitions introduce no local variables, but (d) any local variables in these definitions are reintroduced when the formula label is used. Finally, (e) the local variables of an assertion are combined with those from its presupposition.

(25) PIP expression local variables

- a. $\mathcal{L}^A[x] = \{x\}$
- b. $\mathcal{L}^A\Sigma x\phi = \emptyset$
- c. $\mathcal{L}^A(X \equiv \phi) = \emptyset$
- d. $\mathcal{L}^AX = \mathcal{L}^A\phi$ where $\phi = A(X)$
- e. $\mathcal{L}^A(\phi \mid \psi) = \mathcal{L}^A\phi \cup \mathcal{L}^A\psi$

As for standard ZF expressions, in (26), (a) unbracketed variables do not introduce a local variable, and selective binders (b)–(c), like quantifiers and set abstraction, remove their bound variable from the set of local variables. Other formulas (d)–(g) simply pool the local variables of their parts.

(26) ZF expression local variables

- a. $\mathcal{L}^Ax = \emptyset$
- b. $\mathcal{L}^A\bigcup\{x:\phi\} = \mathcal{L}^A\phi \setminus \{x\}$
- c. $\mathcal{L}^A\exists x\phi = \mathcal{L}^A\phi \setminus \{x\}$
- d. $\mathcal{L}^AP(\tau_1, \tau_2, \dots) = \mathcal{L}^A\tau_1 \cup \mathcal{L}^A\tau_2 \cup \dots$
- e. $\mathcal{L}^A(\tau_1 = \tau_2) = \mathcal{L}^A\tau_1 \cup \mathcal{L}^A\tau_2$
- f. $\mathcal{L}^A\neg\phi = \mathcal{L}^A\phi$
- g. $\mathcal{L}^A(\phi \wedge \psi) = \mathcal{L}^A\phi \cup \mathcal{L}^A\psi$

2.4.3 Translating PIP to ZF

We now give a recursive definition of the translation function $\mathcal{T}$ from PIP to standard ZF.¹²

¹² This does not include the “where” convention. One may eliminate “where” by replacing $\lceil \dots X \dots \text{ where } X \equiv \phi \rceil$ with $\lceil \dots (X \wedge (X \equiv \phi)) \dots \rceil$. If “X” occurs more than once preceding “where,” the definition is attached to the first occurrence.

(27) PIP operators

- a. $\mathcal{T}^A[x] = x$
- b. $\mathcal{T}^A(\phi|\psi) = \mathcal{T}^A\phi$
- c. $\mathcal{T}^A(X \equiv \phi) = \top$ where $\top$ is constant true
- d. $\mathcal{T}^AX = \mathcal{T}^A\phi$ where $\phi = A(X)$
- e. $\mathcal{T}^A\Sigma_{x_1}\phi = \bigcup\{x_1 : \exists x_2 \dots \exists x_n \mathcal{T}^A\phi\}$ where $\{x_2, \dots, x_n\} = \mathcal{L}^A\phi \setminus \{x_1\}$

To paraphrase in English, (a) brackets and (b) presuppositions are removed; (c) formula label definitions are tautologically true, and (d) the labels themselves are replaced by their defined values. The translation function is actually a partial function because of clause (d): if the label assignment A has no value for X , then the translation of X is undefined. The most complicated case is (e) summation, which denotes a set abstraction over the existential closure of the local variables in its scope, excluding the variable of abstraction itself.

Translation of expressions containing standard ZF operators is done transparently: the subexpressions are translated and then recombined using the same operator. At the risk of pedantry:

(28) ZF operators

- a. $\mathcal{T}^Ax = x$
- b. $\mathcal{T}^A\bigcup\{x:\phi\} = \bigcup\{x:\mathcal{T}^A\phi\}$
- c. $\mathcal{T}^A\exists x\phi = \exists x\mathcal{T}^A\phi$
- d. $\mathcal{T}^AP(\tau_1, \tau_2, \dots) = P(\mathcal{T}^A\tau_1, \mathcal{T}^A\tau_2, \dots)$
- e. $\mathcal{T}^A(\tau_1 = \tau_2) = (\mathcal{T}^A\tau_1 = \mathcal{T}^A\tau_2)$
- f. $\mathcal{T}^A\neg\phi = \neg\mathcal{T}^A\phi$
- g. $\mathcal{T}^A(\phi \wedge \psi) = \mathcal{T}^A\phi \wedge \mathcal{T}^A\psi$

In addition to this translation function conditioned on a label assignment function A , we define an unconditional translation function for complete discourses in (30). It makes use of the notation $\mathcal{A}\phi$, which represents the set of label assignments contained in ϕ :

$$(29) \quad \mathcal{A}\phi = \{ \langle X, \psi \rangle : \ulcorner X \equiv \psi \urcorner \text{ appears in } \phi \}$$

Then:

$$(30) \quad \phi \text{ discourse-translates to } \psi \text{ iff } \mathcal{A}\phi \text{ is a function and } \mathcal{T}^{\mathcal{A}\phi}\Sigma_w\phi = \psi.$$

A PIP formula ϕ may fail to have a discourse-translation if the set of ordered pairs $\mathcal{A}\phi$ is not a function (because some label has multiple inconsistent definitions in ϕ), or if $\mathcal{T}^{\mathcal{A}\phi}\phi$ is not defined (because some label that is used in ϕ lacks a definition in ϕ or has a circular definition).

2.4.4 Felicity

As noted above (§2.3), PIP's " $|$ " operator contributes nothing to a discourse's truth conditions; its contribution is to the felicity conditions. Many authors, including Heim & Kratzer, encode such felicity in truth values, either as a third truth value, or a lack of truth value. (See also Stalnaker's "Bridge Principle," which requires a truth value in every world of the conversational context set.) We consider it more straightforward to instead treat truth and felicity as independent properties of a sentence.

We consider a discourse to be subject to several felicity conditions. The following is (we propose) a partial list:

- (31) a. The discourse-translation (30) must be defined.
 b. The discourse-translation must contain no free variables. (Free variables would make the value indeterminate.)
 c. The discourse violates no presuppositions.

By condition (a), a discourse is infelicitous if it contains inconsistent or circular formula-label definitions, or if it uses formula labels that are not defined. By condition (b), any free variables that occur in the individual sentences of the discourse must be closed by the discourse closure operator Σw . That is, any free variables other than w must be local variables.

As for condition (c), let us define $\mathcal{F}^A \phi$ to mean that expression ϕ violates no presuppositions, which is to say that ϕ is **felicitous with respect to presuppositions**. More precisely, $\mathcal{F}$ translates a PIP expression to a ZF expression, but unlike $\mathcal{T}^A \phi$, which represents the truth conditions of ϕ , the ZF expression $\mathcal{F}^A \phi$ represents the presuppositional felicity conditions of ϕ . (As with $\mathcal{T}$, we permit the input to $\mathcal{F}$ to be either a formula or a term.) A discourse ϕ satisfies condition (c) just in case the following is true:

$$(32) \quad \mathcal{F}^A \phi \Sigma w \phi.$$

We will give a recursive definition for $\mathcal{F}$, which also defines the meaning of the " $|$ " operator, as the only operator that directly imposes a condition on felicity. The other PIP operators can first be eliminated by expanding them out using their definitions, so we require recursive clauses only for the presupposition operator and the ZF operators.

Every presuppositional infelicity (a false value for an $\mathcal{F}$ formula) can be traced to a presupposition violation. That is, the ultimate source of any presuppositional infelicity is an expression of form $\phi | \psi$, for which:

$$(33) \quad \mathcal{F}^A(\phi | \psi) \text{ iff } \mathcal{F}^A \phi \wedge \mathcal{F}^A \psi \wedge \mathcal{T}^A \psi.$$

In English, an expression of form $\phi|\psi$ is felicitous just in case the body ϕ and presupposition ψ are both felicitous, and the presupposition is true. Conversely, if $\mathcal{T}^A\psi$ is false, then $\phi|\psi$ is infelicitous.

In a conjunction, we follow Karttunen (1974) in holding that the first conjunct may satisfy presuppositions of the second (e.g., *France has a King and the King of France is bald*). This is captured in the following clause:

$$(34) \quad \mathcal{F}^A(\phi \wedge \psi) \text{ iff } \mathcal{F}^A\phi \wedge (\mathcal{T}^A\phi \rightarrow \mathcal{F}^A\psi)$$

For the whole to be felicitous, the first conjunct must be felicitous outright, but the second conjunct need only be felicitous when the first conjunct is true. Note one immediate consequence: the felicity conditions for $\phi \wedge \psi$ are not the same as the felicity conditions for $\psi \wedge \phi$. More generally, if ϕ and ψ are truth-functionally equivalent, it does *not* follow that $\mathcal{F}^A\phi \leftrightarrow \mathcal{F}^A\psi$.¹³

For quantification, the scope formula must be true for all values of the quantified variable:

$$(35) \quad \mathcal{F}^A\forall x\phi \text{ iff } \forall x\mathcal{F}^A\phi.$$

And the recursion bottoms out at simple terms, which are always felicitous:

$$(36) \quad \mathcal{F}^A\alpha = \top, \text{ if } \alpha \text{ is a variable or constant.}$$

For other primitive ZF expressions, the whole is felicitous just in case each of the parts is felicitous:

$$(37) \quad \begin{array}{ll} \text{a. } \mathcal{F}^A\mathcal{P}(\tau_1, \dots, \tau_n) & \text{iff } \mathcal{F}^A\tau_1 \wedge \dots \wedge \mathcal{F}^A\tau_n, \\ \text{b. } \mathcal{F}^A(\tau_1 = \tau_2) & \text{iff } \mathcal{F}^A\tau_1 \wedge \mathcal{F}^A\tau_2, \\ \text{c. } \mathcal{F}^A\neg\phi & \text{iff } \mathcal{F}^A\phi, \\ \text{d. } \mathcal{F}^A\bigcup S & \text{iff } \mathcal{F}^AS. \end{array}$$

For expressions containing defined operators, felicity is determined by expanding out the defined operator. In particular, if ϕ expands to ϕ' :

$$(38) \quad \mathcal{F}^A\phi \leftrightarrow \mathcal{F}^A\phi'.$$

In this way, one may obtain the following theorems, which are useful for practical computations:¹⁴

¹³ This is another case where truth-functional equivalence fails to guarantee intersubstitutability in PIP.

¹⁴ Theorem (f) in particular predicts universal projection of presuppositions out of generalized quantifiers. For instance, both *No student in my class quit smoking* and *No student in my class who quit smoking was happy* would presuppose that every student in my class has smoked. While this is a common assumption (Heim 1983; Schlenker 2009: a.o.), it has also often been denied (Beaver 2001; Chierchia 1995: a.o.), and recent experimental work has revealed a more nuanced empirical landscape (Chemla 2009; Zehr et al. 2015). (We thank an anonymous reviewer for pointing this out.) Since the focus of this paper is improper scope phenomena, we will not address this point further here. But the current formulation of PIP could certainly resort to a system of local accommodation to explain apparent non-universal projection, as Heim and Schlenker must in their systems. See also Charlow (2009) for related discussion.

- (39) a. $\mathcal{T}^A(\phi \vee \psi)$ iff $\mathcal{T}^A\phi \wedge (\neg\mathcal{T}^A\phi \rightarrow \mathcal{T}^A\psi)$,
 b. $\mathcal{T}^A(\phi \rightarrow \psi)$ iff $\mathcal{T}^A\phi \wedge (\mathcal{T}^A\phi \rightarrow \mathcal{T}^A\psi)$,
 c. $\mathcal{T}^A(\phi \leftrightarrow \psi)$ iff $\mathcal{T}^A\phi \wedge \mathcal{T}^A\psi$,
 d. $\mathcal{T}^A\exists x\phi$ iff $\forall x\mathcal{T}^A\phi$,
 e. $\mathcal{T}^A(\{x:\phi\})$ iff $\forall x\mathcal{T}^A\phi$,
 f. $\mathcal{T}^A\Sigma y\phi$ iff $\forall y\forall x_1\dots\forall x_n\mathcal{T}^A\phi$ where $\{x_1, \dots, x_n\} = \mathcal{L}^A\phi$.

2.4.5 Meanings

For a ZF expression ϕ , we write $v(\phi, M, g)$ for the standard ZF value of ϕ with respect to model M and variable assignment g . The *truth function* of ϕ is $\lambda g\lambda M.v(\phi, M, g)$. Two expressions are *truth-equivalent* just in case they express the same truth function.

For a PIP expression ϕ , we define a *PIP-value* $V(\phi, M, g, A)$ as follows:

$$(40) \quad V(\phi, M, g, A) = (v(\mathcal{T}^A\phi, M, g), v(\mathcal{F}^A\phi, M, g), \mathcal{L}^A\phi, \mathcal{A}\phi).$$

In words, the PIP-value of an expression is a tuple consisting of its truth value, its felicity value, its free local variables, and the formula-label definitions it contains. The *PIP-value function* or *meaning* $\llbracket \phi \rrbracket$ of a PIP expression ϕ is defined as:

$$(41) \quad \llbracket \phi \rrbracket = \lambda A\lambda g\lambda M.V(\phi, M, g, A).$$

Two expressions are *intersubstitutable* just in case they express the same meaning.

2.4.6 Lambda expressions

We use lambda expressions exclusively during the translation of natural language into PIP and thence ZF. Like propositional functions in ZF axiom schemata, lambda expressions are essentially metalanguage constructions: a lambda function takes an expression as input and produces a new expression as output. That is, $\lambda x\phi$ takes an input expression α and produces an output expression $\phi[x:=\alpha]$, replacing the variable x with α wherever x occurs in ϕ (beta-reduction), provided that no free variables in α are captured as a result. In the same way that we define the meaning of (other) PIP operators and defined ZF operators by specifying how expressions containing them are to be rewritten to equivalent expressions without them, we give the meaning of an expression ϕ containing λ operators by rewriting ϕ (through application of beta-reduction) to an equivalent expression that contains no λ operators. The meaning of ϕ is undefined unless beta-reduction eliminates all occurrences of λ . Since the definedness of meaning is a felicity condition, this amounts to an additional special case: lambda expressions may only be used in such a way that they are eliminated by beta-reduction.

3 Interpreting natural-language trees

One motivation for PIP is to provide concise representations of the meanings of natural-language expressions. Accordingly, in this section, we present a semantic fragment using PIP.

3.1 Heim & Kratzer's system

We take the system of Heim & Kratzer (1998) as representative of what we have been calling the traditional approach. Heim & Kratzer, like Higginbotham (1985), adopt a direct-interpretation philosophy of semantics, in which the semantic system does not *translate* natural-language expressions to logical expressions, but rather natural-language expressions *are* logical expressions, and a formal logic is used only to define and explicate their meanings. To make the idea sharper, we recast Heim & Kratzer's rules of interpretation as *semantic operations*, functions that take the meanings of the parts and yield the meaning of the whole (we split their FA into left-headed FA and right-headed AF):¹⁵

- | | | | |
|------|----|---|------------------------------------|
| (42) | a. | $NN(\phi) = \phi,$ | Nonbranching Nodes |
| | b. | $FA(\phi, \psi) = \phi(\psi),$ | Functional Application |
| | c. | $AF(\phi, \psi) = \psi(\phi),$ | Reverse Functional Application |
| | d. | $IFA(\phi, \psi) = \phi(\Sigma_w \psi),$ | Intensional Functional Application |
| | e. | $PM(\phi, \psi) = \lambda x(\phi(x) \wedge \psi(x)),$ | Predicate Modification |
| | f. | $PA(x, \phi) = \lambda x \phi.$ | Predicate Abstraction |

One should understand the symbols NN, FA, etc. as defined function symbols in PIP. The rules of interpretation now serve simply to determine which semantic operator combines the meanings of the children to produce the meaning of the parent:

- (43)
- a. $\llbracket [\beta] \rrbracket = NN(\llbracket \beta \rrbracket).$
 - b. $\llbracket [\beta \gamma] \rrbracket = FA(\llbracket \beta \rrbracket, \llbracket \gamma \rrbracket),$ if $\beta: \langle \sigma, \tau \rangle$ and $\gamma: \sigma$, for some σ, τ .
 - c. $\llbracket [\beta \gamma] \rrbracket = AF(\llbracket \beta \rrbracket, \llbracket \gamma \rrbracket),$ if $\gamma: \langle \sigma, \tau \rangle$ and $\beta: \sigma$, for some σ, τ .
 - d. $\llbracket [\beta \gamma] \rrbracket = IFA(\llbracket \beta \rrbracket, \llbracket \gamma \rrbracket),$ if $\beta: \langle s, \tau \rangle$, for some τ .
 - e. $\llbracket [\beta \gamma] \rrbracket = PM(\llbracket \beta \rrbracket, \llbracket \gamma \rrbracket),$ if $\beta: \langle e, t \rangle$ and $\gamma: \langle e, t \rangle$.
 - f. $\llbracket [\beta \gamma] \rrbracket = PA(x, \llbracket \gamma \rrbracket),$ if β is a relative operator with syntactic index x .

We have used some new notation in (43). $[\beta]$ represents a parse-tree node with a single child β , and $[\beta \gamma]$ represents a parse-tree node with two children β and γ , in that order. $\llbracket \alpha \rrbracket$ represents the meaning of node α , in the sense of a PIP-value function. We explicate the meaning by equating $\llbracket \alpha \rrbracket$ to a PIP metalanguage expression that is intersubstitutable with it. The notation $\alpha: \tau$ indicates that τ is the semantic type of node α , which is defined to be the semantic type of a ZF expression

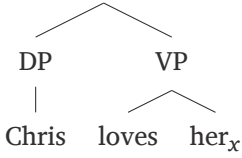
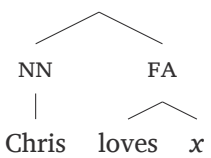
¹⁵ One might analogously introduce a left-headed IAF operation, but we will have no occasion to use it, and so we omit it.

that is truth-equivalent to α . Recall that semantic type e represents individuals and groups of individuals, and s represents worlds and propositions.

The cases in (43) are clauses of a recursive definition. The recursion ends with terminal nodes. We assume that words in the parse tree are sense-disambiguated, that is, the original English words are replaced with representations of word senses. For simplicity, we identify word-sense representations with nonlogical constants (names and predicate symbols) of PIP. Thus:

- (44) a. $\llbracket \alpha \rrbracket = \alpha$, if α is a lexical terminal.
 b. $\llbracket \alpha \rrbracket = x$, if α is a trace or pronoun with syntactic index x .

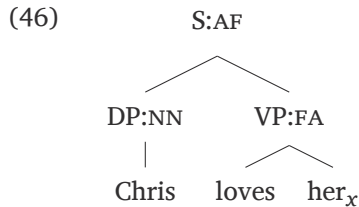
The sense in which PIP directly represents natural-language meaning is this: the meaning of a sentence, represented as a PIP expression using the operators defined in (42), is isomorphic to the LF syntactic tree. For example, consider the sentence (45a), whose LF parse tree is (45b). Applying the rules of interpretation (43) to the LF tree yields the meaning as a PIP expression (45c), and the parse tree of the PIP expression is given in (45d).

- (45) a. Chris loves her_x.
 b. 
 c. $\text{AF}(\text{NN}(\text{CHRIS}), \text{FA}(\text{LOVES}, x))$.
 d. 

Note that (45c) is intersubstitutable with $\text{LOVES}(\text{CHRIS}, x)$, as one can confirm by expanding out the operator definitions using (42) and simplifying. Also, to conform to convention, we have written nonlogical constants in small caps in the PIP expression (45c), though not in the PIP tree (45d).

The LF tree (45b) and PIP tree (45d) have the same shape, and that is guaranteed by the rules of interpretation. The terminal nodes whose LF label is a lexical item have the same label in the PIP tree, and terminal nodes that are traces or pronouns in the LF tree are labeled in the PIP tree with the syntactic index of the trace or pronoun. Nonterminal nodes in PIP are labeled with the semantic operator determined by the rules of interpretation, and that operator is applied to

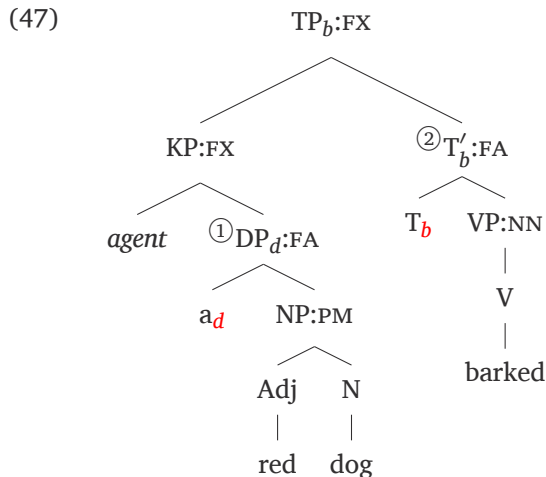
the meanings of the children to produce the meaning of the whole. We may conveniently combine the two trees by annotating the LF nonterminal nodes with the semantic operation that applies:



In the remainder of this section, we expand and modify the Heim & Kratzer system just sketched, in order to cover a range of relevant improper-scope examples.

3.2 Basic noun phrases and T-bar phrases

First, we adopt a Davidsonian treatment of verb meanings—without it, an account of verb modifiers is hardly possible (Davidson 1967; Parsons 1985). A verb denotes a natural kind of event or state, and the syntactic arguments of a verb are, semantically, modifiers whose connection to the event is mediated by a thematic role, whose syntactic category we represent as *K*. For example, we assume the following structure for *a red dog barked*. (The circled numbers are not part of the structure but are included solely for ease of reference, and the red indices are explained in section 3.3 below.)



We have labeled the upper nodes with semantic operations, including one new one (FX) that we will describe shortly. Broadly, (47) asserts that the *agent* relation holds between the dog and the barking event. As is usual, the relation has been binarized; *agent* takes its arguments (nodes ① and ②) one at a time. We focus first on the two arguments.

NP and VP: Working bottom-up, we analyze the NP *red dog* exactly as Heim & Kratzer do: it denotes a function of type $\langle e, t \rangle$ that is true of red dogs. Unlike Heim & Kratzer, however, we take

barked to denote a function of type $\langle e, t \rangle$ that is true, not of barkers, but of events of barking; for clarity, we will often use BARK-EVT as a synonym.

D and T: Terminal nodes with indices, such as the indefinite article a_d and the empty tense element T_b , denote nonlogical constants combined with their first argument. Thus, for example:

$$(48) \quad \llbracket a_d \rrbracket = \lambda P(A(d, P)).$$

The indefinite article (which we write in small caps when used as a PIP predicate symbol) takes two arguments, the first of which is provided by the index d . For convenience, we will abbreviate the right-hand side of (48) as A_d . With that abbreviatory convention in hand, we revise (44a) to read:

$$(49) \quad \llbracket \alpha_x^X \rrbracket = \alpha_x^X, \text{ if } \alpha \text{ is a lexical terminal with label } X \text{ and index } x. \text{ The label and index are both optional.}$$

The predicates A and T have the same definition:

$$(50) \quad \begin{array}{ll} \text{a. } A(x, P) & \text{means } [x] = x \wedge P(x). \\ \text{b. } T(x, P) & \text{means } [x] = x \wedge P(x). \end{array}$$

Thus, A_x and T_x both abbreviate $\lambda P([x] = x \wedge P(x))$.

DP and T': Applying the operators A_d and T_b to the corresponding $\langle e, t \rangle$ complements (NP and VP) in (47) yields an open formula that contains a bracketed free variable:

$$(51) \quad \begin{array}{ll} \text{a. } \llbracket \textcircled{1}DP_d \rrbracket &= ([d] = d \wedge \text{RED}(d) \wedge \text{DOG}(d)). \\ \text{b. } \llbracket \textcircled{2}T'_b \rrbracket &= ([b] = b \wedge \text{BARK-EVT}(b)). \end{array}$$

Note that the expressions of (51) are intersubstitutable with the simplified forms in (52). We will simplify in this manner routinely.

$$(52) \quad \begin{array}{ll} \text{a. } \llbracket \textcircled{1}DP_d \rrbracket &= (\text{RED}([d]) \wedge \text{DOG}(d)). \\ \text{b. } \llbracket \textcircled{2}T'_b \rrbracket &= \text{BARK-EVT}([b]). \end{array}$$

The reader may find it surprising that the denotation of the DP in (52a) is sentential (type t). To explain, we view both the DP and T' , semantically, as being *restricted variables*, by which we mean a variable paired with a description of the value the variable takes. The variable is provided by the syntactic index, and the description is provided by the denotation. We say that the DP *indexes* the variable d and *asserts* the description (52a).¹⁶

Top level—“agent” applied to its arguments: Our new operator FX is what allows a standard predicate expecting a type- e argument to combine with a restricted variable. The

¹⁶ The idea of treating DPs as type- t expressions is not actually our invention. It is also found in Heim (1982).

predicate takes the restricted variable's index as its argument, and the restricted variable's assertion is conjoined to the result. Thus, in our example (47), the standard predicate *agent*, of type $\langle e, \langle e, t \rangle \rangle$, may apply to the two restricted variables DP_d and T'_b via two successive applications of FX, yielding:

$$(53) \quad \text{RED}([d]) \wedge \text{DOG}(d) \wedge \text{BARK-EVT}([b]) \wedge \text{HAS-AGENT}(b, d),$$

where $\text{HAS-AGENT}(b, d)$ is defined to mean $\text{AGENT}(d)(b)$. This intuitive account of the FX operation should suffice for computing interpretations. The technical details are given in §3.7.

In the interest of readability, we define:

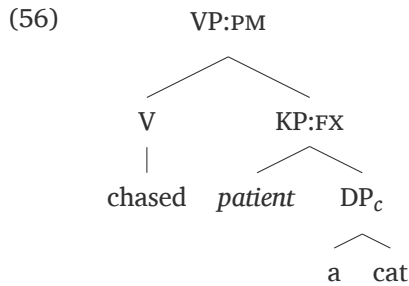
$$(54) \quad \text{BARKED}(e, x) \quad \text{means} \quad \text{BARK-EVT}(e) \wedge \text{HAS-AGENT}(e, x).$$

Then (53) may more compactly be written as:

$$(55) \quad \text{RED}([d]) \wedge \text{DOG}(d) \wedge \text{BARKED}([b], d).$$

Going forward, we will liberally assume defined “Davidsonian relation” symbols like *BARKED* without explicitly writing out their definitions.

In a transitive sentence, the direct object is a KP headed by the thematic role *patient*. As before, the KP denotes a property of events, which may be combined with the verb via PM, as shown in (56):



The reader should be able to confirm that the meaning comes out as:

$$(57) \quad \lambda e(\text{CHASE-EVT}(e) \wedge \text{HAS-PATIENT}(e, c) \wedge \text{CAT}([c])).$$

3.3 Indices

We next explain how indexed elements such as the DP and T' in our example above obtain their indices. We distinguish three classes of elements of category D and T:

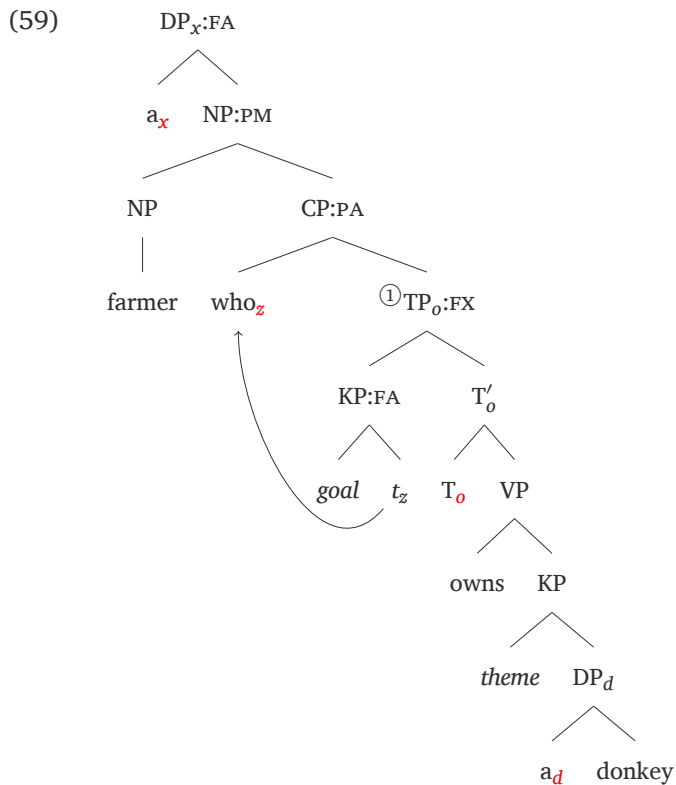
- (58)
- a. Non-anaphoric terminals (D, T, terminal DP/pronoun).
 - b. Non-anaphoric nonterminals (D' , DP, T' , TP).
 - c. Anaphoric elements (D, terminal DP).

Each non-anaphoric terminal (58a) bears an *intrinsic* index, and each intrinsic index must be fresh to the discourse.¹⁷ The terminal DPs in category (58a) are deictic pronouns and relative pronouns. Traces and all remaining pronouns, by contrast, are anaphoric, and belong to category (58c). They do bear indices, but inherit their indices from their antecedents. Finally, nonterminal D' , DP , T' , and TP nodes constitute category (58b). They also bear indices, but inherit their indices from their heads. As an aid to the reader, we have colored all intrinsic indices red in the tree diagrams; all other indices are inherited either from head or antecedent.

The trees we have considered so far do not contain formula labels. They are intrinsic to summation nodes, and inherited (like indices) from antecedents and heads, but we postpone fuller discussion to Section 3.5 below.

3.4 Relative clauses

Before analyzing quantifiers, it will be helpful to examine relative clauses. Consider an indefinite noun phrase containing a relative clause, as in (59). (The trace is anaphoric, but the relative pronoun is non-anaphoric; its index is intrinsic.)



¹⁷ This is to be understood as a syntactic constraint. The syntactic structure of a discourse is a sequence of syntactic trees, one for each sentence, and it is well-formed only if every non-anaphoric terminal has an index, and no two have the same index.

The meaning of ${}^{\textcircled{1}}\text{TP}_o$ is analogous to the meaning of ${}^{\textcircled{1}}\text{TP}_b$ in (47), namely:

$$(60) \quad \text{DONKEY}([d]) \wedge \text{OWNS}([o], z, d)$$

The only notable difference is that the relative pronoun's trace denotes a simple variable, not a restricted variable, and for that reason FA is used, rather than FX, to combine it with the thematic role *goal*:

$$(61) \quad \llbracket t_z \rrbracket = \text{DP-T}_z = z.$$

$$(62) \quad \llbracket \text{KP} \rrbracket = \text{FA}(\text{GOAL}, z) = \lambda e(\text{HAS-GOAL}(e, z))$$

In PIP, we write DP-T instead of T to avoid ambiguity with the tense element of (50b). And, to be clear, DP-T is the identity function, thus $\text{DP-T}_z = \text{DP-T}(z) = z$. The KP meaning (62) combines in turn with $\llbracket \text{T}'_o \rrbracket$ via FX, with the result:

$$(63) \quad \text{HAS-GOAL}(o, z) \wedge \text{OWN-EVT}([o]) \wedge \text{HAS-PATIENT}(o, d) \wedge \text{DONKEY}([d]),$$

which abbreviates to (60) via the introduction of a defined Davidsonian relation symbol OWNS.

Going up one node:

$$(64) \quad \begin{aligned} \text{a. } \llbracket \text{CP} \rrbracket &= \text{PA}(z, \llbracket \text{TP}_o \rrbracket) \\ \text{b. } &= \lambda z \llbracket \text{TP}_o \rrbracket \\ \text{c. } &= \lambda z (\text{DONKEY}([d]) \wedge \text{OWNS}([o], z, d)), \end{aligned}$$

as in Heim & Kratzer. Next, (64c) combines with FARMER and a_x via PM and FA to yield:

$$(65) \quad \llbracket \text{DP}_x \rrbracket = (\text{FARMER}([x]) \wedge \text{DONKEY}([d]) \wedge \text{OWNS}([o], x, d)).$$

Note that DP_x is a restricted variable; it indexes x and asserts (65).

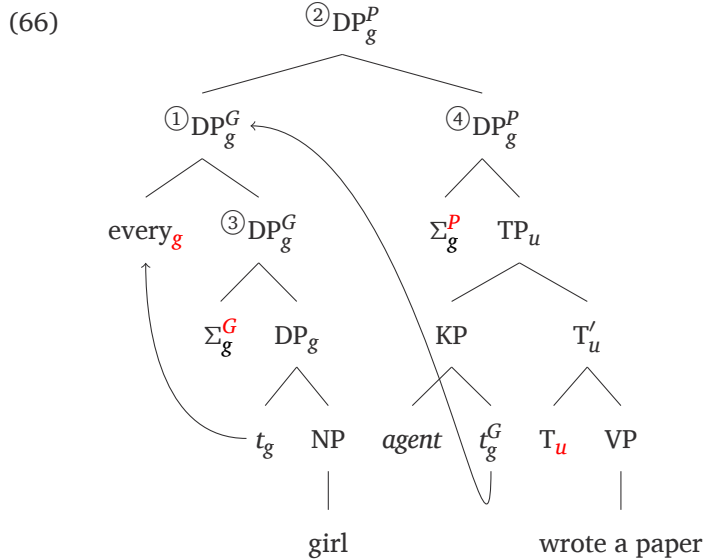
3.5 Generalized quantifiers, summation, and labels

As mentioned above, PIP generalized quantifiers take summation terms as their arguments. We propose a syntactic Σ operator, inserted by quantifier raising, to achieve this. Thus, the structure for generalized quantifiers is as shown in (66).¹⁸ Categories of terminal nodes and headship are left implicit, but to be clear: the Σ operators are of category D, and each of them is the head of the two DP nodes above it (parent and grandparent).

In our analysis, QR involves two movements. First, the quantificational determiner raises, leaving a co-indexed trace: in (66), *every_g* raises to form ${}^{\textcircled{1}}\text{DP}$. As part of the movement operation,

¹⁸ Matthewson (2001) also proposes a structure where the restriction (but not the scope) of a generalized quantifier is a plural individual.

a Σ operator is introduced, sharing an index with the raised element: hence the subscript g on the first Σ (cf. the numerical variable binder Heim & Kratzer (1998: 186) introduced during QR). Next, the same operation is applied to the whole DP: $①DP_g^G$ raises, leaving a trace t_g^G and introducing a Σ operator indexed g again.



Just as T and D elements bear intrinsic lowercase indices, each Σ element bears an intrinsic uppercase label, which is used as a formula label in the interpretation. This is not a mere technical detail; it represents a strong empirical constraint on the use of formula labels. Superscripts on Σ operators are the *only* places where intrinsic formula labels are syntactically permitted to appear.¹⁹

As with intrinsic lowercase indices, the intrinsic label of each Σ element must be unique to that element. This label is passed up to the Σ 's parent and grandparent nodes. Thus, in (66), $③DP$ and $①DP$ inherit the label G and $④DP$ and $②DP$ inherit the label P . Anaphors also inherit the labels of their antecedents, just as they inherit their indices: thus the G superscript on t_g^G , the trace of $①DP$.

Turning now to interpretation, beginning at the top level, the meaning of *every* is:

$$(67) \quad \llbracket \text{every} \rrbracket = \text{EVERY} = \lambda x \lambda y (x \subseteq y),$$

¹⁹ To be completely accurate, intrinsic labels appear as superscripts on syntactic Σ operators and associated with Σ operators that are imputed by the IFA rule (see §3.6 below). Syntactic Σ operators occur in only two places: when inserted by QR, or as part of a summation pronoun.

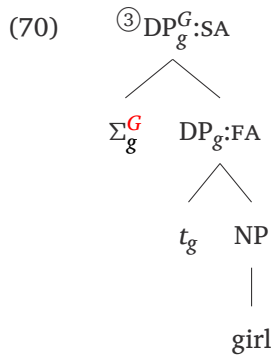
and two applications of FA yield:

$$(68) \quad \llbracket \textcircled{3}\text{DP} \rrbracket \subseteq \llbracket \textcircled{4}\text{DP} \rrbracket.$$

Before considering $\textcircled{3}\text{DP}$ and $\textcircled{4}\text{DP}$, let us dispense with the traces. We assume three types of traces, as in (69), whose meanings correspond to bracketed variables, unbracketed variables, and labels. The three are distinguished syntactically: (69a) is the trace of a D, (69b) of an unlabeled DP, and (69c) of a labeled DP.

- (69) a. $\llbracket t_g \rrbracket = \text{D-T}_g = \lambda P([g] = g \wedge P(g))$ (trace of *every*, bracketed var.)
 b. $\llbracket t_z \rrbracket = \text{DP-T}_z = z$ (trace of relative pronoun, simple var.)
 c. $\llbracket t_g^G \rrbracket = \text{LDP-T}_g^G = G$ (trace of labeled DP, label)

Let us now consider the first argument of *every*, $\textcircled{3}\text{DP}$, which (intuitively) denotes the set of girls:



Beginning with the lower DP, we have:

$$(71) \quad \llbracket \text{DP}_g \rrbracket = \text{D-T}_g(\text{GIRL}) = ([g] = g \wedge \text{GIRL}(g)).$$

The upper DP node uses a new semantic operation (SA), defined as:

$$(72) \quad \text{SA}(x, X, \phi) = (\Sigma_x X \text{ where } X \equiv \phi).$$

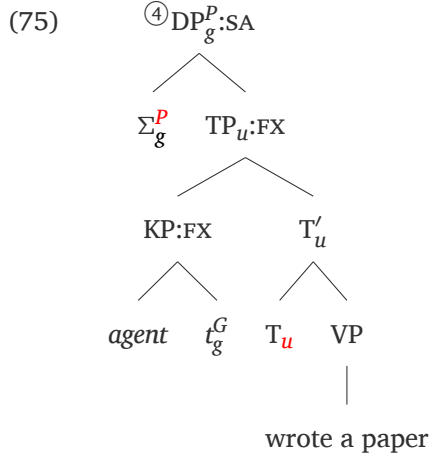
There is a corresponding new rule of interpretation:

$$(73) \quad \llbracket [\beta \ \gamma] \rrbracket = \text{SA}(x, X, \llbracket \gamma \rrbracket), \text{ if } \beta \text{ is the syntactic operator } \Sigma \text{ with label } X \text{ and index } x.$$

The Σ element itself is treated syncategorematically, though its index and label do matter. Applying it to the upper DP node:

$$(74) \quad \llbracket \textcircled{3}\text{DP}_g^G \rrbracket = \Sigma_g G \text{ where } G \equiv \text{GIRL}([g]).$$

We turn now to the second argument of *every*, which is $\textcircled{4}\text{DP}$, denoting the set of girls who wrote a paper:



Applying AGENT to g and G by FX yields:

$$(76) \quad \llbracket \text{KP} \rrbracket = \lambda e(\text{HAS-AGENT}(e, g) \wedge G).$$

That combines with the meaning of T'_u via FX in the same manner that we have already seen, yielding (with some obvious abbreviations):

$$(77) \quad \begin{aligned} \text{a. } \llbracket \text{TP}_u \rrbracket &= (G \wedge \text{WR-EVT}([u]) \wedge \text{PA}([p]) \wedge \text{HAS-PT}(u, p) \wedge \text{HAS-AG}(u, g)). \\ \text{b. } &= (G \wedge \text{PAPER}([p]) \wedge \text{WROTE}([u], g, p)). \end{aligned}$$

Finally, SA applies to yield the meaning of the DP:

$$(78) \quad \llbracket \textcircled{4}\text{DP}_g^P \rrbracket = \Sigma_g P \text{ where } P \equiv (G \wedge \text{PAPER}([p]) \wedge \text{WROTE}([u], g, p)).$$

Combining (68) with (74) and (78), we obtain the meaning of (66):

$$(79) \quad \text{EVERY}(\Sigma_g G, \Sigma_g P) \text{ where} \\ G \equiv \text{GIRL}([g]) \\ P \equiv (G \wedge \text{PAPER}([p]) \wedge \text{WROTE}([u], g, p)).$$

The structure we have proposed predicts that the two sets available for subsequent pronouns are the restriction set and the reference set (the combination of restriction with the scope, see Nouwen 2003a), and this prediction is borne out as shown in (80a) and (80b). By contrast, our analysis provides no DP that denotes the simple scope set (the set of *individuals* that wrote a paper), and the simple scope set in fact is not a legitimate antecedent for anaphora, as illustrated in (80c).²⁰

²⁰ Nouwen argues that the complement set (members of the restriction *not* in the reference set) is also available for future reference under limited circumstances. We adopt his analysis that complement set anaphora arises via a bridging inference and its distribution is highly constrained by independently motivated pragmatic principles. See Nouwen (2003a) for details.

- (80) Most girls wrote a paper.
- a. All of them [the girls] did something for a grade.
 - b. But a few of them [the girls that wrote a paper] left it at home.
 - c. #In fact, they [individuals that wrote a paper] were mostly girls.

3.6 Negation and modals

Denotations for one-place modals and negation (which we treat in essence as a variety of modal) are given in (81). We assume that every intensional terminal must appear syntactically with a superscript label, as shown inside the interpretation brackets $\llbracket \cdot \rrbracket$. (Note that the superscript X labels of NOT^X , MIGHT^X , and MUST^X are required by (49), but are locally semantically vacuous.) These are precisely the elements interpreted by the IFA rule (82), which applies the modal to intensional arguments, and incorporates the modal's syntactic label X into the result.

- (81)
- a. $\llbracket \text{not}^X \rrbracket = \text{NOT}^X = \lambda\psi(w \notin \psi)$
 - b. $\llbracket \text{might}^X \rrbracket = \text{MIGHT}^X = \lambda\beta\lambda\phi\lambda\psi(\beta \cap \phi \cap \psi \neq \emptyset)$
 - c. $\llbracket \text{must}^X \rrbracket = \text{MUST}^X = \lambda\beta\lambda\phi\lambda\psi(\beta \cap \phi \subseteq \psi)$
- (82) Intensional Functional Application (revised)
- a. $\text{IFA}(X, \phi, \psi) = (\phi(\Sigma w X) \text{ where } X \equiv \psi)$
 - b. $\llbracket [\beta \ \gamma] \rrbracket = \text{IFA}(X, \llbracket \beta \rrbracket, \llbracket \gamma \rrbracket)$, if $\beta: \langle s, \tau \rangle$, for some τ , and β is labeled X .

The modals proper (excluding negation) take two arguments from context: the argument β is the modal base, provided by general discourse context, and the argument ϕ is the set of worlds satisfying the restriction, usually provided via an *if* clause. All three lexical items in (81) (including negation) take a body (also called the prejacent or nuclear scope), and the argument ψ is the set of worlds satisfying the body. IFA stores the body formula in the label X and sends the set of worlds satisfying the formula X as an argument to the modal. Note that negation asserts that the current world w is not a member of the set of worlds satisfying formula X .

3.7 Summary

As is usual, we assume that the input to interpretation is fully disambiguated, both syntactically and lexically. To be precise, we assume that the input to interpretation is a syntactic parse tree whose terminal nodes are nonlogical constants representing word senses. Although we have used conventional English orthography for words in parse trees, in contrast to small caps for nonlogical constants, that should be understood merely as a nod to convention.

(83) Indices and labels

- a. Intrinsic indices are assigned to non-anaphoric D, non-anaphoric terminal DP, and T
- b. Intrinsic labels are assigned to Σ , negation, and modal elements
- c. Indices and labels are inherited from head to parent and from antecedent to anaphor
- d. The index, but not the label, is inherited by Σ from its governor.
- e. No other nodes bear indices or labels

(84) $\uparrow$ -lifting a function to take a restricted variable (recursively defined):

- a. $\uparrow f = \lambda\phi\lambda x(f(x) \wedge \phi)$ when f is type $\langle e, t \rangle$
- b. $\uparrow f = \lambda\phi\lambda x(\uparrow(f(x))(\phi))$ otherwise

(85) Semantic operations

- a. $NN(\phi) = \phi$,
- b. $FA(\phi, \psi) = \phi(\psi)$,
- c. $AF(\phi, \psi) = \psi(\phi)$,
- d. $IFA(\phi, \psi) = (\phi(\Sigma wX) \text{ where } X \equiv \psi)$,
- e. $FX(P, x, \phi) = \uparrow P(x, \phi)$,
- f. $PM(\phi, \psi) = \lambda x(\phi(x) \wedge \psi(x))$,
- g. $PA(x, \phi) = \lambda x\phi$.
- h. $SA(x, X, \phi) = (\Sigma xX \text{ where } X \equiv \phi)$.

(86) Rules of interpretation for nonterminal nodes, in order of precedence.

- a. $\llbracket [\beta] \rrbracket = PA(x, \llbracket \beta \rrbracket)$, if β is a restricted variable with index x .²¹
- b. $\llbracket [\beta] \rrbracket = NN(\llbracket \beta \rrbracket)$, otherwise (one child).
- c. $\llbracket [\beta \gamma] \rrbracket = FA(\llbracket \beta \rrbracket, \llbracket \gamma \rrbracket)$, if $\beta: \langle \sigma, \tau \rangle$ and $\gamma: \sigma$, for some σ, τ .
- d. $\llbracket [\beta \gamma] \rrbracket = AF(\llbracket \beta \rrbracket, \llbracket \gamma \rrbracket)$, if $\gamma: \langle \sigma, \tau \rangle$ and $\beta: \sigma$, for some σ, τ .
- e. $\llbracket [\beta \gamma] \rrbracket = IFA(X, \llbracket \beta \rrbracket, \llbracket \gamma \rrbracket)$, if $\beta: \langle s, \tau \rangle$, for some τ ; β has label X .
- f. $\llbracket [\beta \gamma] \rrbracket = FX(\llbracket \beta \rrbracket, x, \llbracket \gamma \rrbracket)$, for γ a restricted variable with index x .
- g. $\llbracket [\beta \gamma] \rrbracket = PM(\llbracket \beta \rrbracket, \llbracket \gamma \rrbracket)$, if $\beta: \langle e, t \rangle$ and $\gamma: \langle e, t \rangle$.
- h. $\llbracket [\beta \gamma] \rrbracket = PA(x, \llbracket \gamma \rrbracket)$, if β is a syntactic relative operator with index x .

The following replaces Heim & Kratzer's Traces and Pronouns rule.

(87) Rule of interpretation for terminal nodes

- a. $\llbracket \alpha_x^X \rrbracket = \alpha_x^X$, if α is a lexical terminal with label X and index x . The label and index are both optional.

²¹ See Section 4.4.

(88) Defined constants

- a. $A_x = \lambda P([x] = x \wedge P(x))$ (indefinite article)
- b. $T_e = \lambda P([e] = e \wedge P(e))$ (Tense)
- c. $D-T_x = \lambda P([x] = x \wedge P(x))$ (trace of D)
- d. $DP-T_x = x$ (trace of relative pronoun)
- e. $E_x = x$ (pronominal core, see Section 4.1.)
- f. $LDP-T_x^X = X$ (trace of labeled DP)
- g. $SHE = \lambda x(x|FEM(x) \wedge SG(x))$
- h. $EVERY = \lambda x \lambda y(x \subseteq y)$.
- i. $NOT^X = \lambda \phi(w \notin \phi)$.
- j. $MIGHT^X = \lambda \beta \lambda \phi(\beta \cap \phi \neq \emptyset)$.
- k. $MUST^X = \lambda \beta \lambda \phi(\beta \subseteq \phi)$
- l. $AGENT = \lambda x \lambda e(HAS-AGENT(e, x))$.
- m. $PATIENT = \lambda x \lambda e(HAS-PATIENT(e, x))$.
- n. $BARKED(e, x) = (BARK-EVT(e) \wedge HAS-AGENT(e, x))$.

Implementation of the FX operation: The type-lifting operator (84) used to implement the FX operation has not been previously introduced. It allows a function to take a restricted variable as its argument, and FX takes a restricted variable as argument and type-lifts the function to correctly apply to this argument.²² This is illustrated in (89) for the meaning of *agent*:

- (89) a. $\llbracket agent \rrbracket = AGENT = \lambda x \lambda e(HAS-AGENT(e, x))$.
 b. $\uparrow AGENT = \lambda \phi \lambda x \lambda e(HAS-AGENT(e, x) \wedge \phi)$.

When employing FA, the meaning in (89a) is used; when employing FX, this meaning is lifted as in (89b). Note that FX, like FA, may be applied iteratively. To interpret (47), the KP meaning was both an output of and an input to FX:

- (90) a. $\llbracket KP \rrbracket = FX(AGENT, d, \llbracket DP_d \rrbracket) = \lambda e(HAS-AGENT(e, d) \wedge \llbracket DP_d \rrbracket)$
 b. $\uparrow \llbracket KP \rrbracket = \lambda \phi \lambda e(HAS-AGENT(e, d) \wedge \llbracket DP_d \rrbracket \wedge \phi)$

Whence:

- (91) $\llbracket TP_b \rrbracket = FX(\llbracket KP \rrbracket, b, \llbracket T'_b \rrbracket) = HAS-AGENT(b, d) \wedge \llbracket DP_d \rrbracket \wedge \llbracket T'_b \rrbracket$

Substituting in the values of $\llbracket DP_d \rrbracket$ and $\llbracket T'_b \rrbracket$ lets us obtain the interpretation of (47):

- (92) $HAS-AGENT(b, d) \wedge RED([d]) \wedge DOG(d) \wedge BARK-EVT([b])$.

²² Cf. B uring's (2005: 100) *combine* operator. Also, recall the argument reversal in the notation $f(x, \phi)$, which is equivalent to $f(\phi)(x)$.

4 Applications

The previous sections motivated each special PIP construction without relying on summation pronouns, paycheck pronouns, quantificational subordination, or modal subordination. Next, we demonstrate how the handful of constructions already introduced can capture the full range of improper scope phenomena, plus cross-sentential presupposition projection.

4.1 Summation pronouns

PIP has two kinds of terms—simple variables and summations—corresponding to variables and set abstractions in ZF. Summation terms were motivated above to allow quantifiers to denote simple, two-place predicates over pluralities.

With two kinds of term, it stands to reason that we might find two kinds of pronouns in natural language, and this is exactly what we find. In particular, we distinguish between *simple pronouns* and *summation pronouns*, and we propose distinct structures for them. A simple pronominal DP has the form:

$$(93) \quad \text{DP:FA} \\ \swarrow \quad \searrow \\ \text{she} \quad e_x$$

The pronoun *she* has the meaning (94a). The empty element e_x is a purely anaphoric element that we call the *pronominal core*. As defined in (88e), it simply denotes its index. Thus:

$$(94) \quad \begin{array}{ll} \text{a. } \llbracket \text{she} \rrbracket = \text{SHE} = \lambda z(z | \text{FEM}(z) \wedge \text{SG}(z)) \\ \text{b. } \llbracket e_x \rrbracket = E_x = x \\ \text{c. } \llbracket \text{DP} \rrbracket = \text{SHE}(x) = (x | \text{FEM}(x) \wedge \text{SG}(x)). \end{array}$$

The resulting DP is a simple variable with a presupposition: it denotes x and presupposes $\text{FEM}(x) \wedge \text{SG}(x)$. We have previously rendered simple pronouns simply as their index—in this case, x . In doing so, we were just suppressing the presupposition in the interest of simplicity.

A summation pronoun has the following structure:

$$(95) \quad \begin{array}{c} \textcircled{1} \text{DP:FA} \\ \swarrow \quad \searrow \\ \text{pron} \quad \textcircled{2} \text{DP}_x^Y:\text{SA} \\ \quad \quad \swarrow \quad \searrow \\ \quad \quad \Sigma_x^Y \quad t_x^X \end{array}$$

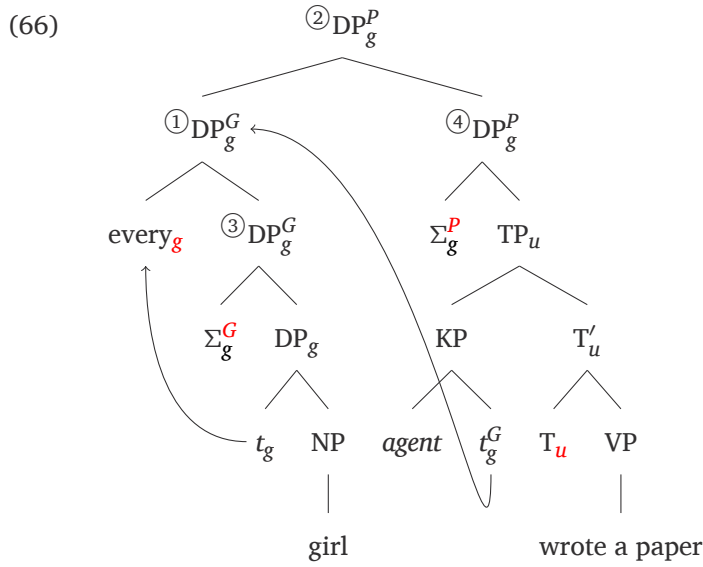
The pronominal core t_x^X is identical to a labeled DP trace (69c). Its semantic value is just the label. Thus the interpretation of the lower DP is just as with the examples of SA in the preceding sections:

- (96) a. $\llbracket \textcircled{2} \text{DP}_x^Y \rrbracket = \Sigma x Y$ where $Y \equiv X$,
 b. $= \Sigma x X$.

The meaning of a pronoun that presupposes property Q , as we have seen in (94a), is $\lambda z(z|Q(z))$. Thus, the upper DP in (95) has interpretation:

- (97) $\llbracket \textcircled{1} \text{DP} \rrbracket = (\Sigma x X) | Q(\Sigma x X)$.

To give an example best captured using a summation pronoun, consider again the structure and interpretation for (66) (“every girl wrote a paper”), repeated here:



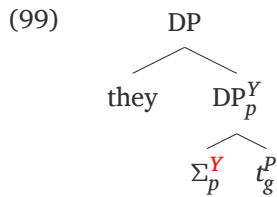
The meaning, written in PIP, is:

- (79) $\text{EVERY}(\Sigma_g G, \Sigma_g P)$ where
 $G \equiv \text{GIRL}([g])$
 $P \equiv (G \wedge \text{PAPER}([p]) \wedge \text{WROTE}([u], g, p))$.

The next sentence after this might be:

- (98) They_p^P are on Ms. Marple’s desk,

where they_p^P is a summation pronoun. Its structure is:



The antecedent of t_g^P is 2 DP in (66), denoting $\Sigma_g P$, the set of girls that wrote a paper; however in this case, only the label P contributes its meaning to the summation pronoun. (99) denotes $\Sigma p P$ with a presupposition:

$$(100) \quad (\Sigma p P) |_{PL(\Sigma p P)}.$$

That is, it denotes the set of papers that $\Sigma_g P$ wrote, and it presupposes that this set has cardinality greater than one.

Cases where a pronoun refers to the restriction or reference set of a quantifier are also best captured using summation pronouns:

$$(101) \quad \text{Most dogs bark. } \underline{\text{They}} \text{ are loud.}$$

Consider again the PIP analysis of the first sentence of (101):

$$(102) \quad \text{MOST}(\Sigma d D, \Sigma d B) \text{ where}$$

$$D \equiv \text{DOG}([d])$$

$$B \equiv (D \wedge \text{BARKS}(d))$$

A pronoun appearing after this sentence, like *they* in (101), can refer to the dogs that satisfy the nuclear scope B (those that bark). This may be captured as follows (where the structure of the summation pronoun *they* is not shown, but the indices of the empty core are preserved):

$$(103) \quad \text{They}_d^B \text{ are loud} \rightsquigarrow \text{LOUD}(\Sigma d B)$$

Replacing formula labels with their definitions, (103) expands out to (104a), equivalent to the ZF expression (104b):

$$(104) \quad \text{a. } \text{LOUD}(\Sigma d (\text{DOG}([d]) \wedge \text{BARKS}(d))) \quad [\text{PIP}]$$

$$\text{b. } \text{LOUD}(\bigcup \{d : \text{DOG}(d) \wedge \text{BARKS}(d)\}) \quad [\text{ZF}]$$

Formula (104) asserts that all barking dogs are loud.

Finally, as discussed by (Keshet 2018: §4.2), summation pronouns (*compound terms* in Keshet's DUAL system) can solve a problem due to Nouwen (2003b) for Dynamic Plural Logic (van den Berg 1996). Consider Nouwen's example:

$$(105) \quad \text{Three}_s \text{ students each [wrote exactly two}_p \text{ papers]}^W. \text{ They each sent them to L\&P.}$$

[Nouwen's (5.8), labels and indices added]

There are two readings here. One, captured in DUAL, PIP, and Dynamic Plural Logic via simple pronouns, is completely distributive: each student sent in their own two papers to L&P. The other reading is collective with respect to the papers: each student submitted all six papers to L&P

(perhaps fraudulently claiming to have written them all). This reading is not available in Dynamic Plural Logic, but PIP/DUAL can handle it easily with a summation pronoun: ‘ $\Sigma_p W$ ’ denotes all the papers that the three students wrote.

An anonymous reviewer points out cases where this second reading seems restricted, such as Brasoveanu’s 2008 example here:

- (106) Every parent who gives three balloons to two boys expects them to end up fighting (each other) for them. [Brasoveanu’s (8)]

We maintain that the reading is still available in cases like (106), but it is dispreferred when there is less pragmatic support. Here, there is no mention of the boys gathering, a pragmatic prerequisite for *them* to refer to all the boys or all the balloons. Similar sentences with such support fare better:

- (107) a. Every parent who had two children at Tappan Middle School was pleased when they (all) formed a rock band, the Tappin’ Twofers, just for kids with siblings at the school.
b. Everyone who ordered two or more drinks decided it was easier to just split the bill for them (all) evenly.

Discussion: We are not alone in observing two such kinds of pronouns. For instance, our summation pronouns are quite like E-type pronouns (Evans 1977; 1980), which are similarly complex, denoting the unique individual (or maximal group) satisfying some salient predicate. And van Rooy (2001) proposes two similar types: *descriptive pronouns*, which denote “the exhaustive set of individuals denoted by the description recovered from the clause in which [an] antecedent occurs,” and *referential pronouns*, which are not exhaustive in the same way.²³

We add the following evidence for two different interpretations of pronouns. PIP predicts that summation pronouns are exhaustive in a way that simple pronouns are not. Consider:²⁴

- (108) a. Some_g girls were having lunch in the cafeteria.
b. They_g waved to some (boys/other girls) having lunch there, too.
(109) a. Most_g girls^L [were having lunch in the cafeteria].
b. They_g^L waved to some (boys/#other girls) having lunch there, too.

Variables introduced by unembedded indefinites are available for later simple pronouns, such as the *g* introduced by *some* and used by *they* in (108a). Such cases are not exhaustive; *g* may denote any plurality of girls having lunch, as shown by the felicity of mentioning “other” girls in (108b). Variables introduced by generalized quantifiers, however, are bound inside the quantification

²³ These pronouns, van Rooy suggests, are made exhaustive via a connection to the speaker’s intended referents.

²⁴ This evidence is similar to a test proposed in Szabolcsi (1997), wherein the follow-up would be *Perhaps some other girls were having lunch there, too*.

and any reference to them is necessarily via summation terms, such as *they* in (109b).²⁵ And summations are exhaustive: $they_g^L$ denotes the complete set of girls lunching in the cafeteria. This correctly predicts the infelicity of mentioning “other” girls also lunching there in (109b).²⁶

Such empirical considerations require theories to provide two interpretations of pronouns: one exhaustive and one not. While these interpretations could be captured in various ways, the two pronoun types provided by PIP are a welcome empirical result.

4.2 Paycheck pronouns

Summation pronouns can also handle **paycheck pronouns** (Karttunen 1969), so-called because of examples like (110).

- (110) The woman who saved her paycheck was wiser than the woman who spent it. (cf. Jacobson 2000)

Although paycheck pronouns are easily accommodated within E-type approaches, they are notoriously difficult for plural dynamic logics to capture (Nouwen 2020). Take the case in (111), for instance. Standard plural logics only store individuals already mentioned or quantified over. So, after (111a), they would only store the dioramas that girls made and brought to class. And yet the pronoun *it* in (111b) seems to refer to dioramas left at home, new individuals not mentioned before.

- (111) a. Almost every_x girl brought the_d Σ_d^D [diorama she_x made] to class.
b. Very few_x of them forgot it_d^D at home.

The PIP analysis of paycheck pronouns (following Keshet 2018) uses summation pronouns. The definite description in (111a) stores the formula corresponding to *diorama she made* in the label *D*, giving the summation pronoun in (111b) the value in (112):

²⁵ We follow Milsark (1977) in assuming that certain determiners are ambiguous between strong and weak versions, which we take to be parallel to our distinction between quantifiers, which require summation over the restriction and nuclear scope, and indefinites, which lack such summation. The determiner *most* in (109) is always strong, and therefore must be analyzed using summation terms. *Some* in (108), on the other hand, may be weak or strong. The weak version, without an exhaustive summation term, must be assumed for (108) to sound felicitous.

²⁶ An anonymous reviewer points out that not literally all the girls having lunch have to wave in order to make (108b) true. This is reminiscent of an observation due to Dowty (1987) about the following sentence:

(i) At the end of the press conference, the reporters asked the president questions.

Not all of the reporters need to have asked a question in order for (i) to be true. Such *non-maximality* (Brisson 1998) is a general property of plural predication, especially involving definite DPs (including pronouns): an apparently distributive predicate may sometimes be true of a group even when it is not true of all the members of that group. We will not take a strong stance on this topic, since it is largely orthogonal to our concerns. See Križ (2016), though, for a nice overview of the topic and one influential analysis.

- (112) a. ΣdD where $D \equiv (\text{DIORAMA}([d]) \wedge \text{MADE}(x, d))$ [PIP]
 b. $\bigcup \{d : \text{DIORAMA}(d) \wedge \text{MADE}(x, d)\}$ [ZF]

Notice that x is an external variable here, remaining free even in the ZF translation of ΣdD . This allows x to be bound by another, higher operator, in this case the generalized quantifier *few*. Thus, *it* in (111b) can refer to dioramas not mentioned before: those made by the few forgetful students. This is the defining feature of a paycheck pronoun; its antecedent formula contains a free variable bound by a different operator in its antecedent position than when it is used for the paycheck pronoun.

4.3 Donkey pronouns

To wrap up our discussion of pronouns, we will take a closer look at donkey anaphora. The donkey pronouns examined so far have been analyzed as so-called weak donkey pronouns (Schubert & Pelletier 1989). Take the following sentence, for example, with its PIP translation:

- (113) Everyone who owns an umbrella brought it to school today.

$$\rightsquigarrow \text{EVERY}(\Sigma xO, \Sigma xB) \text{ where}$$

$$O \equiv (\text{UMBRELLA}([u]) \wedge \text{OWNS}(x, u))$$

$$B \equiv (O \wedge \text{BROUGHT}(x, u))$$

The most likely scenario verifying (113) is that even those who own more than one umbrella only brought one, and the sentence is perfectly acceptable in this scenario, as predicted. This weak donkey reading is also compatible with the less likely scenario that one or more of the multiple-umbrella owners brought more than one umbrella, perhaps as a backup or for a friend.

We make a new observation about these latter scenarios. In particular, a summation pronoun after (113) denoting “ ΣuB ”, as in (114), will denote the exhaustive set of umbrellas that were actually brought. When at least one person brought more than one umbrella, (114) asserts that all of the umbrellas are in the rack, not just one per owner (modulo the sort of non-maximality mentioned in footnote (i)). So, although only one per owner is required for the truth of (113), all brought umbrellas are included in later anaphora like (114). And this is also exactly as predicted.

- (114) They _{u} ^{B} are in that rack.

The other major reading of donkey pronouns is the strong reading, where the nuclear scope must be true of all instantiations of the donkey pronoun (e.g., x brought all x ’s umbrellas). Consider:

- (115) Everyone who brought something valuable locked it up.

As with the umbrellas, (115) is again most acceptable in a situation where each person brought exactly one valuable item. And yet, hearers will again accept the sentence when a few people

brought more than one valuable item. In this case, though, the most salient reading is that each person who brought multiple valuable items locked up all of their valuables.

PIP provides a neat account for strong donkey anaphora. If we allow indefinites like *something* to optionally denote generalized quantifiers, the strong donkey meaning for (115) is immediately predicted, with *it* as a summation pronoun:

(116) EVERY(ΣxA , ΣxL) where

$A \equiv (\text{PERSON}([x]) \wedge \text{SOME}(\Sigma vV, \Sigma vB))$ where

$V \equiv \text{VALUABLES}([v])$

$B \equiv (V \wedge \text{BROUGHT}(x, v))$

$L \equiv (A \wedge \text{LOCKED}(x, \Sigma vB))$

The value of label *B* is “ $\text{VALUABLES}([v]) \wedge \text{BROUGHT}(x, v)$,” with *x* free and external. This allows *it* to denote ΣvB , the complete set of valuables that *x* brought.

If no one brought more than one valuable item, as implicated by the singular *something valuable*, this set is always a singleton, and the singular presupposition of *it* is satisfied. It seems, though, that this presupposition may be relaxed in edge cases, where *v* is singular for most values of *x*, even if it is plural for a minority.²⁷

Finally, we note that Keshet (2018: §3.6) shows how a quite similar system can generate the same sorts of mixed weak and strong readings as Brasoveanu (2008), who also encapsulates the weak/strong difference in two different versions of indefinites.

4.4 Quantificational subordination

The phenomenon of quantificational subordination is illustrated in (117), with a first sentence, followed by two different possible second sentences:

(117) Almost every student brought an umbrella today.

- a. Most (of them) used it, too.
- b. Every one (of them) who used it stayed dry.

Previous works (e.g. Karttunen 1969; Sells 1985) have noted that an indefinite embedded under a quantifier, such as *an umbrella* in (117), may serve as an antecedent to pronouns in the nuclear scope of later quantifiers, such as *most (of them)* in (117a). These later quantifiers are said to be **subordinate** to the previous quantifier. We make the novel observation that such pronouns may also appear in subordinate restrictions, as in *every one (of them) who used it* in (117b).

²⁷ Similar facts are noted in scenarios invoking alternatives; see Sudo (2012) and Sauerland (2013). Alternatively, Keshet (2018) proposes a silent distributive operator binding such a strong donkey pronoun, which can account for its singular marking; this operator could be easily ported to PIP as well.

Quantificational subordination is accommodated in PIP by adding a formula label in the restriction of the subordinate quantifier, anaphoric to the main quantification. This is parallel to the formula label which incorporates the restriction into the nuclear scope of a single quantified sentence. In fact, a thread of formula labels can be traced from the first restriction of the main quantifier to the nuclear scope of the subordinate quantifier. Consider the analyses below for (117a) and (117b):

(118) ALMOST-EVERY(ΣsS , ΣsB) where

$$S \equiv \text{STUDENT}([s])$$

$$B \equiv (S \wedge \text{UMBRELLA}([u]) \wedge \text{BROUGHT}([b], s, u))$$

(119) EVERY(ΣsE , ΣsD) where

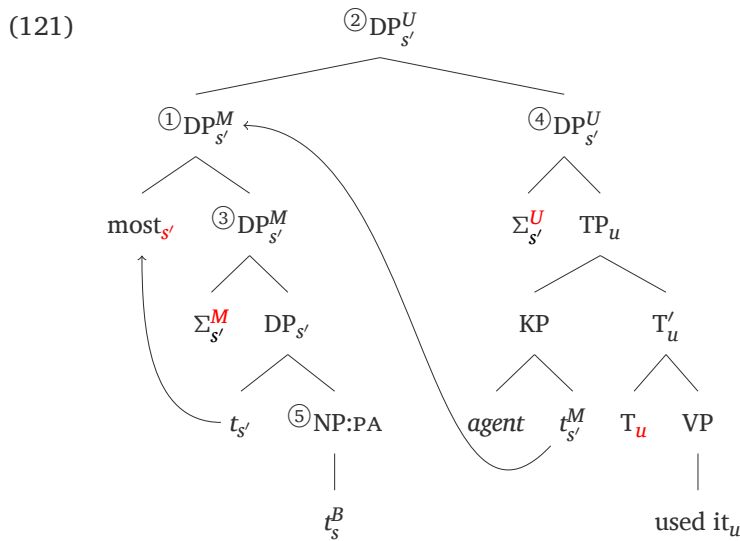
$$E \equiv (B \wedge \text{USED}([e], s, u))$$

$$D \equiv (E \wedge \text{DRY}([d], s))$$

Note that the label S is a clause of the label B , which is a clause of E , which is a clause of D . In this way, any variable introduced in an earlier part of the chain (as u is introduced in B) may be used in a later part (as u is used in E).

Let us illustrate the structure of a subordinate sentence using the simpler (117a). The structure, shown in (121), is nearly identical to the structure of our main quantifier example (66). The chief difference between this structure and (66) is ^5NP in the restriction. The empty element t_s^B behaves like a full DP trace, such as t_s^M in the KP. Its antecedent, though, is the top node of sentence (117), from which it takes its index s and label B , as required by (83c). It represents a restricted variable that indexes s and asserts B :

(120) $\llbracket t_s^B \rrbracket = B$.



As required by (83a), the index s' of *most* is intrinsic, and thus new in the discourse. Thus s' is a distinct index from s , and yet we do need a connection between them. That connection is established by the interpretation of $^{\textcircled{5}}\text{NP}$ and its parent.

The semantic operation for $^{\textcircled{5}}\text{NP}$ is PA, which takes a variable x and an assertion ϕ and produces $\lambda x\phi$. Heim & Kratzer use PA for relative clause interpretation, by the rule of interpretation (43f). To this, we have added another, separate rule of interpretation, where the semantic operation PA applies to a single restricted variable; this is rule (86a), repeated here:

$$(122) \quad \llbracket [\beta] \rrbracket = \text{PA}(x, \llbracket \beta \rrbracket), \text{ if } \beta \text{ is a restricted variable with index } x.$$

Thus:

$$(123) \quad \llbracket ^{\textcircled{5}}\text{NP} \rrbracket = \lambda sB = \lambda s(\text{STU}([s]) \wedge \text{UMB}([u]) \wedge \text{BROUGHT}([b], s, u)).$$

The trace $t_{s'}$ of *most* is interpreted just as in (66). Namely:

$$(124) \quad \llbracket t_{s'} \rrbracket = \lambda Q([s'] = s' \wedge Q(s'))$$

Combining $\llbracket t_{s'} \rrbracket$ with $\llbracket ^{\textcircled{5}}\text{NP} \rrbracket$ by FA, we obtain:

$$(125) \quad \llbracket \text{DP}_{s'} \rrbracket = ([s'] = s' \wedge (\lambda sB)(s'))$$

which simplifies to:

$$(126) \quad \llbracket \text{DP}_{s'} \rrbracket = (\text{STUDENT}([s']) \wedge \text{UMBRELLA}([u]) \wedge \text{BROUGHT}([b], s', u)).$$

We then apply the summation:

$$(127) \quad \llbracket ^{\textcircled{3}}\text{DP}_{s'}^M \rrbracket = \Sigma_{s'} M \text{ where} \\ M \equiv \text{STUDENT}([s']) \wedge \text{UMBRELLA}([u]) \wedge \text{BROUGHT}([b], s', u).$$

The definition of M is identical to the definition of B , apart from the replacement of s with s' . The rest of the interpretation proceeds just as in (66). The final result is:

$$(128) \quad \text{MOST}(\Sigma_{s'} M, \Sigma_{s'} U) \text{ where} \\ \text{a. } M \equiv \text{STUDENT}([s']) \wedge \text{UMBRELLA}([u]) \wedge \text{BROUGHT}([b], s', u). \\ \text{b. } U \equiv M \wedge \text{USED}([e], s', u).$$

Additional clauses can be conjoined in the same fashion, yielding cases like (117b) above.

Notice that s' is in every instance a bound variable in (128), with the result that we could have omitted the prime without affecting the meaning. For this reason, in similar examples that arise below, we generally do omit the prime.

4.5 Modals and modal subordination

Let us examine the analysis of modals a little further. Modals in PIP, analogous to generalized quantifiers, are relations between sets of possible worlds. The difference comes in the fact that modals must indicate a **modal base** (Kratzer 1981), which determines what flavor of modality is in effect: epistemic, deontic, etc. We make no specific claim about the source of this modal base, but we can represent it as an argument (usually implicit in natural language sentences) β_w , where β_w is a set of worlds accessible from w .²⁸ This approach allows us to define existential and universal modals roughly as follows:

- (129) a. $\text{MIGHT}(\beta_w, W_1, W_2) \approx (\beta_w \cap W_1 \cap W_2) \neq \emptyset$
 b. $\text{MUST}(\beta_w, W_1, W_2) \approx (\beta_w \cap W_1) \subseteq W_2$

After the modal base, modals take two sets of possible worlds as arguments. The first is the modal's quantificational restriction, while the second is its nuclear scope. Notice that under this analysis, the world-dependence of a modal derives mainly from the world-dependence of its modal base.²⁹

Since the restriction of a modal is usually provided by an *if*-clause, let us first consider a conditional donkey sentence:

- (130) a. If she_x has a_p pet, it_p must be a donkey [LF]
 b. $\text{MUST}(\beta_w, \Sigma wP, \Sigma wD)$ where [PIP]
 $P \equiv \text{PET-OF}_w([p], x)$
 $D \equiv (P \wedge \text{DONKEY}_w(p))$

The first argument of **MUST** is the modal base, the second represents the *if* clause restriction (*if she has a pet*), and the third represents the nuclear scope / prejacent (*it must be a donkey*). Just as with individual quantifiers, the nuclear scope of a modal is subordinate to its restriction. Thus, the pronoun *it* in the nuclear scope may access an antecedent indefinite in the restriction.

In cases without an *if*-clause, we assume a tautological restriction. Accordingly, we define two-place versions of the modals as follows: $\text{MIGHT}(\beta_w, W)$ asserts that $\beta_w \cap W$ is nonempty and $\text{MUST}(\beta_w, W)$ asserts that $\beta_w \subseteq W$. For example, (131a) translates as (131b):

- (131) a. It might rain.
 b. $\text{MIGHT}(\beta_w, \Sigma w \text{RAIN}_w())$
 $\leftrightarrow \text{MIGHT}(\beta_w, \Sigma w \top, \Sigma w \text{RAIN}_w())$, with $\top$ always true
 $\leftrightarrow (\beta_w \cap \Sigma w \text{RAIN}_w()) \neq \emptyset$.

²⁸ Another option would be to assume an accessibility relation $R(w, u)$ true of any world u accessible from w . Then, β_w would be equivalent to $\Sigma u R(w, u)$.

²⁹ We do not discuss *de re* elements here, but those would be another source of world-dependence.

With these definitions, modal subordination may be treated precisely parallel to quantificational subordination. Consider the following example (Roberts 1987):

- (132) a. A wolf might enter.
 b. It would eat Tasty Tim first.
- (133) a. $\text{MIGHT}(\beta_w, \Sigma wW)$ where $W \equiv (\text{WOLF}_w([x]) \wedge \text{ENTERS}_w(x))$
 b. $\text{MUST}(\beta_w, \Sigma wW, \Sigma wE)$ where $E \equiv (W \wedge \text{TIM}_w([t]) \wedge \text{EATS}_w(x, t))$

The nuclear scope of (132a) is stored in the label W . The second modal, in (132b), is subordinate to the first. Its restriction is anaphoric to the nuclear scope of (132a), as indicated by the use of the label W , just as in quantificational subordination. The nuclear scope of (132b) is subordinate to its restriction, as usual, and hence the nuclear-scope meaning E includes W , giving the pronoun in the nuclear scope access to the antecedent that occurs in the preceding sentence.

4.6 Negation

Natural language negation seems to involve the existential closure of indefinite variables:

- (134) $\text{He}_x \text{ doesn't have a}_p \text{ pen.} \rightsquigarrow \neg \exists p (\text{PEN}(w, p) \wedge \text{HAS}(w, x, p))$ [ZF]

And negation can serve as the antecedent for modal subordination (Sells 1985):

- (135) a. He doesn't own a car.
 b. It would be too expensive.

Together, we take these facts to indicate that natural-language negation involves summation over possible worlds, as sketched here:

- (136) $(134) \rightsquigarrow (w \notin \Sigma wH) \text{ where } H \equiv (\text{PEN}_w([p]) \wedge \text{HAVE}_w(x, p))$
 (137) $(135a) \rightsquigarrow (w \notin \Sigma wO) \text{ where } O \equiv (\text{CAR}_w([c]) \wedge \text{OWN}_w(x, c))$
 (135b) $\rightsquigarrow \text{MUST}(\beta_w, \Sigma wO, \Sigma wE) \text{ where } E \equiv (O \wedge \text{EXPENSIVE}_w(c))$

Essentially, negation asserts that the possible world w is not among those verifying the prejacent.

Negation also demonstrates how formula-label anaphora is much less constrained than simple-variable anaphora. Although negation renders embedded indefinites inaccessible for later pronouns, embedded formula labels are still quite accessible. This freedom helps explain cases that traditional dynamic systems struggle with, such as double negation:

- (138) a. It's not like $\text{he}_x \text{ doesn't}^O \text{ [own a}_c \text{ car]. It}_c^O \text{ is just in the shop.}$
 b. $(w \notin \Sigma w(w \notin \Sigma wO)) \wedge \text{IN-SHOP}_w(\Sigma cO | \text{SG}(\Sigma cO)) \text{ where}$
 $O \equiv (\text{CAR}_w([c]) \wedge \text{OWNS}_w(x, c))$

The pronoun *it* in (138) is perfectly acceptable, even though its antecedent is embedded (twice) under negation. It denotes the sum of all cars that *x* owns. The singular pronoun presupposes via the predicate ‘SG’ that this sum is a singleton—i.e., *x* owns one car—which is easily accommodated.

Despite this flexibility, the system also correctly restricts anaphora out of negation where it should be restricted:

- (139) a. He doesn’t^O [own a_c car].
 b. #It_c^O is in the shop.

Here, (139a) asserts that *x* doesn’t own a car, and therefore the presupposition of the pronoun cannot be met; the set of *x*’s cars is empty rather than singleton.

Relatedly, Matthew Mandelkern (p.c.) suggests that PIP might incorrectly predict the following sentence to be felicitous:

- (140) Mary thinks I don’t have a child, but I *am* a parent. #He lives in England.

The issue here is that *he* could be a summation pronoun over the labeled formula under the negation, the PIP translation of *I have a child*. And with such a summation pronoun, no bridging inference would even be necessary: *he* would straightforwardly denote the child(ren) of the speaker, like any other summation pronoun. We agree that this analysis is available, but contend that (140) is rendered odd due to the difficulty of accommodating the gender and and number presuppositions of the pronoun *he*. Examples that better support such accommodation sound much improved:

- (141) a. Mary thinks I don’t have a son, but she’s wrong. She just hasn’t met him yet.
 b. Mary thinks I don’t have any children, but I am actually a father several times over. She just hasn’t ever met any of them!

4.7 Negation and disjunction

PIP can also handle Barbara Partee’s “bathroom” example, shown in (142), which involves negation under disjunction:

- (142) Either there is no bathroom here or it’s in a funny place.
 (attributed to Barbara Partee in Roberts 1987)

PIP assigns a formula like the following to this sentence:

- (143) $(w \notin \Sigma wX \vee \text{FUNNY-PLACE}_w(\Sigma bX \mid \text{SG}(\Sigma bX)))$ where
 $X \equiv (\text{SG}(b) \wedge \text{BATHROOM}_w([b]) \wedge \text{HERE}_w(b))$

The use of *it* as a summation pronoun is facilitated by negation in the first clause, as sketched in (143). The question is whether the existential presupposition of this pronoun is met.

As described in section 2.4.4, the felicity conditions for disjunction are as follows, repeated from (39a):

$$(144) \quad F(\phi \vee \psi) \text{ iff } F\phi \wedge (\neg\phi' \rightarrow F\psi)$$

Since $w \notin \Sigma wX$ contains no presuppositions itself, this translates to the ZF formula in (145) for (143) (note that $\neg(w \notin \Sigma wX)$ is equivalent to $\exists bX$):

$$(145) \quad \left(\begin{array}{l} \exists b (SG(b) \wedge \text{BATHROOM}(w, b) \wedge \text{HERE}(w, b)) \\ \rightarrow SG(\bigcup \{b : (SG(b) \wedge \text{BATHROOM}(w, b) \wedge \text{HERE}(w, b))\}) \end{array} \right) \quad [\text{Felicity}]$$

The presupposition can then be accommodated by assuming there is at most one bathroom here. And even without such an assumption the plural version works: *Either there are no bathrooms here or they are in a funny place.*

4.8 Presupposition and plurality

Our account of presupposition projection in Section 2.4.4 hewed closely to standard accounts such as Karttunen (1974) and Heim (1983). As such, we do not expect any new observations to arise for cases well examined in the literature. However, having an account of presupposition wedded with a full plural semantics lets us examine the interaction of these two systems, as we did in the analysis of Partee's bathroom sentences just now.

We show next how PIP allows us to analyze presuppositions under quantificational subordination.³⁰ To start, note that systems of presupposition projections like Heim (1983) allow presuppositions in the nuclear scope of a quantifier to be satisfied in a pointwise fashion for each member of the restriction. For instance, the sentences in (146) ought not to have presuppositions as a whole:

- (146) a. Every monarchy cherishes its monarch.
 b. Some people in my apartment building who own a bicycle own a car, too.
 c. Most of my friends who used to smoke have quit smoking by now.

In each case, under Heim's analysis, the nuclear scope is evaluated in a context already updated by the restriction (in a pointwise fashion). And this updated context necessarily satisfies the presuppositions raised in the nuclear scope: any monarchy has a monarch, anyone who owns a bicycle owns some means of transportation, and anyone who used to smoke has indeed smoked

³⁰ Please note that we do not mean to imply that PIP is *uniquely* situated to do so. If other plural semantic systems were to add a parallel account of presupposition, they could likewise capture these data. PIP is simply the first to do this, to our knowledge.

regularly. PIP also correctly captures these cases, since a PIP nuclear scope includes its restriction, yielding quite similar results.

Next, Heim (1992) extends the analysis to cases involving attitude reports:

- (147) John believes that Mary is the only one here, but he wishes that Susan were here too.
[Heim's (53)]

PIP can handle cases like (147) via the same mechanism as modal subordination. Let us assume that the translation of the *believes*-clause above defines a formula label including the PIP translation of *Mary is here*, as in (148) (where β_w is written out as John's belief worlds):

- (148) $\text{MUST}(\Sigma u \text{BELIEF-WORLD}_w(\text{JOHN}, u), \Sigma w(M \wedge \text{SG}(\Sigma_x \text{HERE}_w(x))))$
where $M \equiv \text{HERE}_w(m)$

Then, the *wishes*-clause can be subordinate to this labeled formula, as in (149), where the presupposition of *too* is satisfied by the conjunction with the label M :

- (149) $\text{MUST}(\Sigma u \text{WISH-WORLD}_w(\text{JOHN}, u), \Sigma w S)$ where
 $S \equiv (M \wedge \text{HERE}_w(s) | (\exists x \neq s \text{HERE}_w(x)))$

Next, we may apply this PIP analysis to an empirical domain not analyzed to our knowledge in the prior literature. Namely, presupposition satisfaction may similarly extend across quantificational subordination:

- (150) a. In the 1700s, every European country was a monarchy. Most of them cherished their monarchs.
b. Everyone in my apartment building owns a bicycle. Some of them own a car, too.
c. Every friend of mine used to smoke. But most of them have quit smoking by now.

As before, the label defined in the nuclear scope of the first sentences above will be incorporated into the second sentences, satisfying the presuppositions there, with the result that the discourse as a whole makes no presuppositions.

5 Conclusion

We have pared down the mechanisms required for an account of improper scope phenomena plus presuppositions to three:

- i. Scope extension of indefinites,
- ii. Repetition of subformulas, and
- iii. Standard presupposition projection principles.

Beyond this original motivation, the resulting system has revealed certain surprising applications, especially for summation pronouns, which

- correctly capture certain exhaustive readings of pronouns,
- provide an implementation for strong donkey pronouns,
- allow anaphora out of double negation, and
- help implement Partee’s bathroom sentences.

In related work (Keshet & Abney 2024), we also show how summation pronouns correctly capture anaphoric restrictions in intensional sentences.

That said, we do leave (much) room for further work on PIP. For instance, the translation procedure in §3 above constrains formula label meanings to three positions in syntax: the trace of a generalized quantifier DP, the restriction of a subordinate quantifier, and the pronominal core of a summation pronoun. If formula labels were available in other locations, unattested readings could arise; for instance, indefinites introduced within quantified structures could conceivably (incorrectly) take scope outside of those structures. We leave to future work what precisely characterizes the three positions for formula label meanings, along with many more small mysteries arising from this new work.

Competing interests

The authors have no competing interests to declare.

Author contributions

Authors are listed in alphabetic order. Both authors contributed equally to all aspects of the research and preparation of the work.

References

- Barwise, Jon & Cooper, Robin. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy* 4(2). 159–219. DOI: <https://doi.org/10.1007/BF00350139>
- Beaver, David & Krahmer, Emiel. 2001. A partial account of presupposition projection. *Journal of Logic, Language and Information* 10. 147–182. DOI: <https://doi.org/10.1023/A:1008371413822>
- Beaver, David I. 2001. *Presupposition and assertion in dynamic semantics*, vol. 29. CSLI publications, Stanford.
- Blamey, Stephen. 1986. Partial logic. In *Handbook of philosophical logic: Volume III: Alternatives in classical logic*, 1–70. Springer. DOI: https://doi.org/10.1007/978-94-009-5203-4_1
- Brasoveanu, Adrian. 2007. *Structured nominal and modal reference*. Rutgers University New Brunswick, NJ dissertation.
- Brasoveanu, Adrian. 2008. Donkey pluralities: Plural information states versus non-atomic individuals. *Linguistics and Philosophy* 31. 129–209. DOI: <https://doi.org/10.1007/s10988-008-9035-0>
- Brasoveanu, Adrian & Farkas, Donka F. 2011. How indefinites choose their scope. *Linguistics and Philosophy* 34(1). 1–55. DOI: <https://doi.org/10.1007/s10988-011-9092-7>
- Brisson, Christine M. 1998. *Distributivity, maximality, and floating quantifiers*. Rutgers The State University of New Jersey, School of Graduate Studies.
- Büiring, Daniel. 2005. *Binding Theory*. Cambridge: Cambridge University Press.
- Charlow, Simon. 2009. “Strong” predicative presuppositional objects. In *Proceedings of ESSLLI*, vol. 109.
- Chemla, Emmanuel. 2009. Presuppositions of quantified sentences: Experimental data. *Natural Language Semantics* 17. 299–340. DOI: <https://doi.org/10.1007/s11050-009-9043-9>
- Chierchia, Gennaro. 1995. *Dynamics of meaning: Anaphora, presupposition, and the theory of grammar*. The University Of Chicago Press. DOI: <https://doi.org/10.7208/chicago/9780226104515.001.0001>
- Coppock, Elizabeth & Champollion, Lucas. 2024. Invitation to formal semantics. Manuscript, Boston University and New York University. <https://eecoppock.info/semantics-boot-camp.pdf>.
- Cresswell, Max J. 2002. Static semantics for dynamic discourse. *Linguistics and Philosophy* 25(5/6). 545–571. DOI: <https://doi.org/10.1023/A:1020834910542>

- Davidson, Donald. 1967. The logical form of action sentences. *Essays on actions and events* 5. 105–148. DOI: <https://doi.org/10.1093/0199246270.003.0006>
- DeVries, Karl. 2016. *Independence friendly dynamic semantics: Integrating exceptional scope, anaphora and their interactions*. UC Santa Cruz dissertation.
- Dotlačil, Jakub. 2013. Reciprocals distribute over information states. *Journal of Semantics* 30(4). 423–477. DOI: <https://doi.org/10.1093/jos/ffs016>
- Dotlačil, Jakub & Roelofsen, Floris. 2020. A dynamic semantics of single-wh and multiple-wh questions. In Rhyne, Joseph & Lamp, Kaelyn & Dreier, Nicole & Kwon, Chloe (eds.), *Proceedings of the 30th semantics and linguistic theory conference*, 376–395. DOI: <https://doi.org/10.3765/salt.v30i0.4839>
- Dowty, David. 1987. Collective predicates, distributive predicates, and all. In Miller, Ann & Zhang, Zheng-sheng (eds.), *Proceedings of the 3rd eastern states conference on linguistics (ESCOL)*, 97–115.
- Elbourne, Paul. 2005. *Situations and individuals*. Cambridge: The MIT Press.
- Evans, Gareth. 1977. Pronouns, quantifiers, and relative clauses (I). *Canadian Journal of Philosophy* 7(3). 467–536. DOI: <https://doi.org/10.1080/00455091.1977.10717030>
- Evans, Gareth. 1980. Pronouns. *Linguistic Inquiry* 11(2). 337–362.
- Geach, Peter T. 1962. *Reference and generality: an examination of some medieval and modern theories*. Ithaca, NY: Cornell University Press.
- Groenendijk, Jeroen & Stokhof, Martin. 1991. Dynamic predicate logic. *Linguistics and Philosophy* 14(1). 39–100. DOI: <https://doi.org/10.1007/BF00628304>
- Heim, Irene. 1982. *The semantics of definite and indefinite noun phrases*. Univeristy of Massachusetts, Amherst dissertation.
- Heim, Irene. 1983. On the projection problem for presuppositions. In Barlow, Michael & Flickinger, Daniel P. & Wescot, Michael T. (eds.), *WCCFL 2: Second annual west coast conference on formal linguistics*, vol. 2. 114–125.
- Heim, Irene. 1992. Presupposition Projection and the Semantics of Attitude Verbs. *Journal of Semantics* 9(3). 183–221. DOI: <https://doi.org/10.1093/jos/9.3.183>
- Heim, Irene & Kratzer, Angelika. 1998. *Semantics in generative grammar*. Oxford: Blackwell.
- Henderson, Robert. 2014. Dependent indefinites and their post-suppositions. *Semantics and Pragmatics* 7. 6–1. DOI: <https://doi.org/10.3765/sp.7.6>
- Higginbotham, James. 1985. On semantics. *Linguistic inquiry* 16(4). 547–593.
- Jacobson, Pauline. 2000. Paycheck pronouns, Bach-Peters sentences, and variable-free semantics. *Natural Language Semantics* 8(2). 77–155. DOI: <https://doi.org/10.1023/A:1026517717879>
- Jech, Thomas. 2002. *Set theory: The third millennium edition, revised and expanded* (Springer Monographs in Mathematics). Springer Berlin Heidelberg 3rd edn. DOI: <https://doi.org/10.1007/3-540-44761-X>
- Kamp, Hans. 1981. A theory of truth and semantic representation. *Formal Semantics*, 189–222. DOI: <https://doi.org/10.1002/9780470758335.ch8>

- Karttunen, Lauri. 1969. Pronouns and variables. In *Fifth regional meeting of the Chicago Linguistic Society*, 108–115.
- Karttunen, Lauri. 1974. Presupposition and linguistic context. *Theoretical Linguistics* 1(1–3). 181–194. DOI: <https://doi.org/10.1515/thli.1974.1.1-3.181>
- Keshet, Ezra. 2018. Dynamic update anaphora logic: A simple analysis of complex anaphora. *Journal of Semantics* 35(2). 263–303. DOI: <https://doi.org/10.1093/jos/ffx020>
- Keshet, Ezra & Abney, Steven. 2024. Intensional anaphora. *Semantics and Pragmatics* 17. DOI: <https://doi.org/10.3765/sp.17.9>
- Kratzer, Angelika. 1981. The notional category of modality. In Eikmeyer, Hans J. & Rieser, Hannes (eds.), *Words, worlds, and contexts* (New Approaches in Word Semantics), 38–74. Berlin, Boston: De Gruyter. DOI: <https://doi.org/10.1515/9783110842524-004>
- Križ, Manuel. 2016. Homogeneity, non-maximality, and all. *Journal of Semantics* 33(3). 493–539. DOI: <https://doi.org/10.1093/jos/ffv006>
- Kuhn, Jeremy. 2017. Dependent indefinites: the view from sign language. *Journal of Semantics* 34(3). 407–446. DOI: <https://doi.org/10.1093/jos/ffx007>
- Lewis, David. 1975. Adverbs of quantification. In Keenan, Edward L. (ed.), *Formal semantics of natural language*, 3–15. Cambridge Univ. Press. DOI: <https://doi.org/10.1017/CBO9780511897696.003>
- Mandelkern, Matthew. 2022. Witnesses. *Linguistics and Philosophy* 45(5). 1091–1117. DOI: <https://doi.org/10.1007/s10988-021-09343-w>
- Matthewson, Lisa. 2001. Quantification and the Nature of Crosslinguistic Variation. *Natural Language Semantics* 9(2). 145–189. DOI: <https://doi.org/10.1023/A:1012492911285>
- Milsark, Gary. 1977. Towards the Explanation of Certain Peculiarities of Existential Sentences in English. *Linguistic Analysis* 3. 1–29.
- Montague, Richard. 1973. The proper treatment of quantification in ordinary English. *Approaches to Natural Language* 49. 221–242. DOI: https://doi.org/10.1007/978-94-010-2506-5_10
- Murray, Sarah E. 2008. Reflexivity and reciprocity with (out) underspecification. In *Proceedings of Sinn und Bedeutung*, vol. 12. 455–469.
- Nouwen, Rick. 2003a. Complement anaphora and interpretation. *Journal of Semantics* 20(1). 73–113. DOI: <https://doi.org/10.1093/jos/20.1.73>
- Nouwen, Rick. 2020. E-type pronouns: Congressmen, sheep, and paychecks. *The Wiley Blackwell Companion to Semantics*, 1–28. DOI: <https://doi.org/10.1002/9781118788516.sem091>
- Nouwen, Rick Willem Frans. 2003b. *Plural pronominal anaphora in context: Dynamic aspects of quantification*. Utrecht University dissertation.
- Parsons, Terence. 1985. Underlying events in the logical analysis of English. In LePore, Ernest & McLaughlin, Brian P. (eds.), *Actions and events: Perspectives in the philosophy of Donald Davidson*, 235–267. Oxford: Blackwell.
- Roberts, Craige. 1987. *Modal subordination, anaphora, and distributivity*. UMass Amherst dissertation.

- Roelofsen, Floris & Dotlačil, Jakub. 2023. Wh-questions in dynamic inquisitive semantics. *Theoretical Linguistics* 49(1–2). 1–91. DOI: <https://doi.org/10.1515/tl-2023-2001>
- Russell, Bertrand. 1905. On denoting. *Mind* 14(56). 479–493. DOI: <https://doi.org/10.1093/mind/XIV.4.479>
- Sauerland, Uli. 2005. Don't interpret focus!: why a presuppositional account of focus fails, and how a presuppositional account of givenness works. In Maier, Emar & Bary, Corien & Huitink, Janneke (eds.), *Sinn und Bedeutung* 9.
- Sauerland, Uli. 2013. Presuppositions and the alternative tier. In Snider, Todd (ed.), *Proceedings of the 23rd semantics and linguistic theory conference*, 156–173. DOI: <https://doi.org/10.3765/salt.v23i0.2673>
- Schlenker, Philippe. 2009. Local contexts. *Semantics and Pragmatics* 2(3). 1–78. DOI: <https://doi.org/10.3765/sp.2.3>
- Schubert, Lenhart K. & Pelletier, Francis Jeffry. 1989. Generically speaking, or, using discourse representation theory to interpret generics. In *Properties, types and meaning*, 193–268. Springer. DOI: https://doi.org/10.1007/978-94-009-2723-0_6
- Sells, Peter. 1985. *Restrictive and non-restrictive modification*, vol. 28. Center for the Study of Language and Information, Stanford University.
- Sudo, Yasutada. 2012. *On the semantics of phi features on pronouns*. Massachusetts Institute of Technology dissertation.
- Szabolcsi, Anna. 1997. Background Notions in Lattice Theory and Generalized Quantifiers. In Szabolcsi, Anna (ed.), *Ways of scope taking*, 1–27. Kluwer. DOI: https://doi.org/10.1007/978-94-011-5814-5_1
- van den Berg, Martin H. 1996. *Some aspects of the internal structure of discourse: the dynamics of nominal anaphora*. ILLC dissertation.
- van Rooy, Robert. 2001. Exhaustivity in dynamic semantics; referential and descriptive pronouns. *Linguistics and Philosophy* 24(5). 621–657. DOI: <https://doi.org/10.1023/A:1017597801178>
- Zehr, Jérémy & Bill, Cory & Tieu, Lyn & Romoli, Jacopo & Schwarz, Florian. 2015. Existential presupposition projection from none? An experimental investigation. In *Proceedings of the 20th Amsterdam colloquium*, 448–457.

