Proceedings of SALT 34: 356–369, 2024

Determiners do not need to manage donkey anaphora*

Anand Abraham
University of California, Los Angeles

Abstract If a quantificational determiner denotes a relation between the restrictor and the scope, then in order to handle donkey anaphora, the transmission of anaphoric information from the restrictor to the scope must be baked into the lexical semantics of the determiner. By analyzing quantification as unary, rather than relational, I show that donkey anaphora can be accounted for without defining determiners which explicitly manage the flow of anaphoric information.

Keywords: semantics, donkey anaphora, modification, quantification

1 Introduction

Quantificational determiners have relatively simple core meanings. For example, *most* means something like (1).

(1) $\textbf{most}(A)(B) = |A \cap B| > \frac{1}{2}|A|$

But such a meaning for *most* is inadequate for dealing with donkey anaphora, sentences such as (2).

(2) Most students who saw a dog pet it.

The problem is that there is an anaphoric relationship between *dog* and *it*, but there is no way for this relation to be established in a simple semantics.

To give an analysis of donkey anaphora, it is necessary to add more complexity to the semantics. Something has to facilitate the transmission of anaphoric information, so that the donkey antecedent can bind the donkey pronoun. In doing so, it is necessary to complicate the semantics of quantificational determiners. Dynamic approaches (e.g. Heim 1982; Muskens 1996) replace truth values with context update functions, so that anaphoric information can be stored in and retrieved from the context. A dynamic generalized quantifier, like Chierchia's (1992) in (3),[1] dynamically conjoins the restrictor to the scope to ensure that anaphoric information introduced in the restrictor is accessible in the scope.

* Thanks to audiences at SALT 34 and UCLA for helpful comments and questions. Thanks also to Dylan Bumford for much useful discussion.

1 Given as part of a general scheme for lifting ordinary generalized quantifiers to dynamic versions.

(3) $[\![\text{most}]\!](P)(Q) = \uparrow \mathbf{most}(\{x \mid \downarrow {}^{\vee}P(x)\})(\{x \mid \downarrow {}^{\vee}P(x) \mathbin{\triangle}^{\vee} Q(x)\})$

Situation-based e-type approaches (e.g. Heim 1990; Elbourne 2001) analyze donkey pronouns as definite descriptions, which can pick out the appropriate antecedent from a minimal situation. A situation-based generalized quantifier, like Heim's (1990) in (4), checks for each element whether any minimal situation satisfying the restrictor can be extended to a situation satisfying the scope, passing anaphoric information through the situation variable.

(4) $[\![[\text{most}_{x,s_1}\zeta]_{s_2}\varphi]\!]^g =$
$|\{\mathbf{x} \mid \mathbf{min}(\{\mathbf{s_1} \mid [\![\zeta]\!]^{g_{x\backslash \mathbf{x}, s_1 \backslash \mathbf{s_1}}}\}) \subseteq \{\mathbf{s_1} \mid \exists \mathbf{s_2}.\mathbf{s_1} \leq \mathbf{s_2} \wedge [\![\varphi]\!]^{g_{x\backslash \mathbf{x}, s_1 \backslash \mathbf{s_1}, s_2 \backslash \mathbf{s_2}}}\}\}|$
$> \frac{1}{2}|\{\mathbf{x} \mid \exists \mathbf{s_1}.[\![\zeta]\!]^{g_{x\backslash \mathbf{x}, s_1 \backslash \mathbf{s_1}}}\}|$

One thing both of these approaches have in common is that it is the role of the quantificational determiner to manage the transmission of anaphoric information from the restrictor to the scope. Under standard assumptions about the syntax-semantics of quantification, this choice is forced. If a quantifificational deteminer denotes a relation between the restrictor and the scope, then there is no place for the restrictor and scope to communicate without the mediation of the determiner.

There is reason to be dissastisfied by this. Why should determiners be responsible for handling donkey anaphora? In principle, quantification and anaphora are separate phenomena. Tying the analyses of quantification and anaphora together is undesirable for the sake of theoretical modularity: any change in the analysis of anaphora requires also coming up with a scheme for writing generalized quantifiers in a way that respects anaphora.

One analysis of donkey anaphora which avoids this is that of Barker & Shan (2008). They accomplish this by splitting the semantics of determiners into two levels, allowing indefinites to scope between these layers. As a result, a donkey indefinite scopes over a donkey pronoun and binds it normally. Unfortunately, this turns out to be too liberal, allowing for violations of the Binder Roof Constraint (Brasoveanu & Farkas 2011), which they discuss in Barker & Shan 2014. The present approach draws heavy inspiration from theirs, treating quantifiers and pronouns as involving scopal layers.

There is a way to avoid hard-coding management of anaphoric information into the semantics of determiners: by changing the syntax-semantics of quantification. I propose that the semantic contribution of a quantificational phrase be decomposed into two steps. When a quantificational phrase enters a structure, it introduces its domain, which is semantically available for anaphora and modification. At a higher position, a unary quantificational operator applies. The determiner in its base position has no semantic contribution, only marking a phrase for future quantification. On this approach, anaphoric relationships can be established before quantification happens, and quantification only has to deal with counting, not anaphora. Donkey anaphora

in particular arises as a special interaction between indefiniteness and restrictive modification, where a modifier with an indefinite in it can add extra information to its host, which is available for anaphora.

2 Separating the domain from quantificational force

In order to introduce the domain of quantification separately from quantification, I extend Jäger's (2001) use of partial identity functions to all noun phrases, recording domain information in the domain restriction of the partial function. These are associated with higher operators, which actually perform the quantification over the domain. I write $\alpha \rhd_\chi \beta$ as a type of partial functions from D_α to D_β, where χ is an annotation encoding quantificational force. For example, *most students* has the denotation and type (separated by ::) in (5).

(5) $[\![\text{most students}]\!] = \lambda x : \mathsf{student}(x).x \ :: \ e \rhd_{\mathbf{m}} e$

The domain of quantification (students) is encoded in the domain of the partial function.

A type like $\alpha \rhd_\chi \beta$ behaves functorially: that is, it can behave as an element of type β for the purposes of combination, while propagating the partial dependency on α to the result. Formally, this is handled through the use of combinators which generate new rules of combination from existing ones. See Bumford & Charlow 2024 for more on this sort of compositional architecture.

(6) Any $a :: \alpha$ and $b :: \alpha \to \beta$ can be combined via **app**, with $\mathbf{app}(a,b) = b(a)$

(7) a. If r is a rule of combination that takes any $a :: \alpha$ and $b :: \beta$ to $r(a,b) :: \gamma$, then $\mathbf{L}(r)$ is a rule of combination that takes any $a :: \delta \rhd_\chi \alpha$ and $b :: \beta$ to $\mathbf{L}(r)(a,b) = \lambda x : x \in \mathbf{dom}\ a.r(a(x),b)$, of type $\delta \rhd_\chi \gamma$.

b. If r is a rule of combination that takes any $a :: \alpha$ and $b :: \beta$ to $r(a,b) :: \gamma$, then $\mathbf{R}(r)$ is a rule of combination that takes any $a :: \alpha$ and $b :: \delta \rhd_\chi \beta$ to $\mathbf{R}(r)(a,b) = \lambda x : x \in \mathbf{dom}\ b.r(a,b(x))$, of type $\delta \rhd_\chi \gamma$.

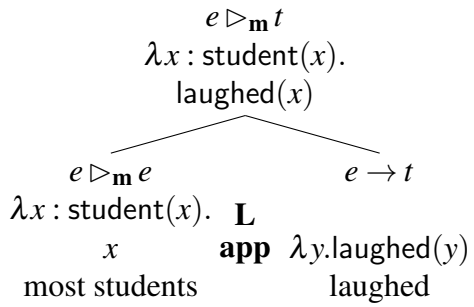
L makes it possible to propagate a partial dependency from the argument in function application, **R** makes it possible to propagate one from the function. As an example, **L** can be used to combine *most students* and a simple predicate like *laughed*.

(8) $[\![\text{most students laughed}]\!] = \mathbf{L}(\mathbf{app})([\![\text{most students}]\!], [\![\text{laughed}]\!])$
$= \mathbf{L}(\mathbf{app})(\lambda x : \mathsf{student}(x).x, \lambda y.\mathsf{laughed}(y))$
$= \lambda x : \mathsf{student}(\mathrm{x}).\mathbf{app}(x, \lambda y.\mathsf{laughed}(y))$
$= \lambda x : \mathsf{student}(x).\mathsf{laughed}(x) \ :: \ e \rhd_{\mathbf{m}} t$

It will often be more clear to represent derivations like (8) using the notation in (9).[2]

2 Taking inspiration from the tower notation of Barker & Shan 2008

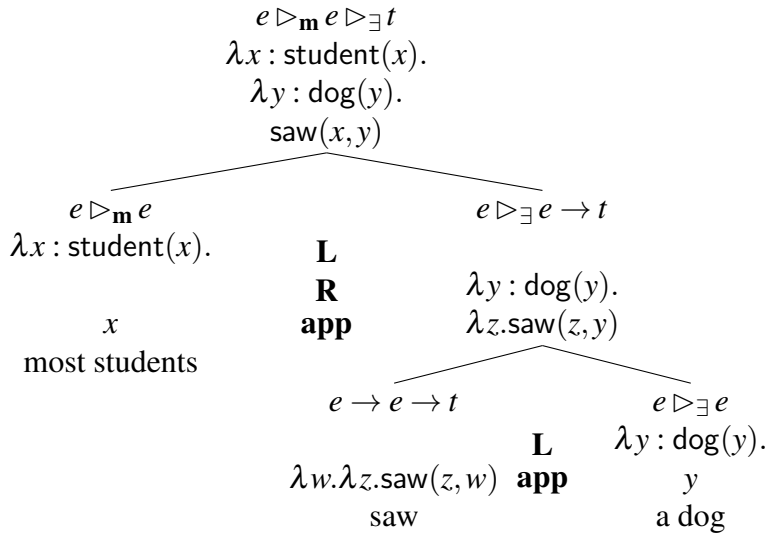
(9)



In such a representation, each extra vertical level represents the application of a combinator. At the bottom level, there is function application, and at the second level there is the application of **L**, to allow the $\lambda x : \mathsf{student}(x)$ to "pass over" the function application at the base level.

L and **R** can apply to rules created by each other, allowing the layering of different quantifiers. In (10), a derivation is given of *most students saw a dog*.

(10)



When composing two constituents with quantifiers, there are multiple ways to put them together, corresponding to different scopings. For example, if the rule $\mathbf{R}(\mathbf{L}(\mathbf{app}))$ was used in the last step of (10), inverse scope would have been derived.

Quantification happens through the application of a closure operator keyed to the partial dependency. For *most*, there is the associated operator **m**.

(11) $\mathbf{m} = \lambda \varphi.|\{x \mid x \in \mathbf{dom}\ \varphi, \varphi(x)\}| > \frac{1}{2}|\{x \mid x \in \mathbf{dom}\ \varphi\}|\ ::\ (e \rhd_{\mathbf{m}} t) \to t$

This can be applied to the result of (8) to get a truth value.

(12) $\mathbf{app}(\lambda x : \mathsf{student}(x).\mathsf{laughed}(x), \mathbf{m}) = |\{x \mid \mathsf{student}(x), \mathsf{laughed}(x)\}| > \frac{1}{2}|\{x \mid \mathsf{student}(x)\}|$

These operators can also make use of **L** and **R** for combination. For example, to evaluate (10) it is necessary to apply existential closure under the $e \rhd_{\mathbf{m}}$, as in (13).

(13) $\exists = \lambda\varphi.\exists x.x \in \textbf{dom}\ \varphi \wedge \varphi(x)\ ::\ (e \rhd_{\exists} t) \to t$
$\mathbf{L}(\textbf{app})(\lambda x : \mathsf{student}(x).\lambda y : \mathsf{dog}(y).\mathsf{saw}(x,y), \exists)$
$= \lambda x : \mathsf{student}(x).\textbf{app}(\lambda y : \mathsf{dog}(y).\mathsf{saw}(x,y), \exists)$
$= \lambda x : \mathsf{student}(x).\exists y : \mathsf{dog}(y) \wedge \mathsf{saw}(x,y)\ ::\ e \rhd_{\mathbf{m}} t$

The idea that quantification should be decomposed into the introduction of the domain and the quantification over that domain is not new. Bumford (2017) uses a similar strategy to handle the semantics of definites, decomposing the semantics of *the* into a referent-introducing and uniqueness component. Arguably, any invocation of existential closure is an instance of split quantification, too. What I am proposing here is that all quantification can be understood in this way. Rather than taking in the restrictor and scope separately, quantification is unary, taking in one object which contains information about both the restrictor and scope.

A benefit of this approach to quantification is that only conservative quantification is possible. The only quantifiers definable as functions of type $(e \rhd t) \to t$ are the conservative ones. An object of type $e \rhd t$ contains three pieces of information: what the domain is, which elements of the domain the scope is true on, and which elements of the domain the scope is false on.[3] It does not contain any information about the scope on elements outside of the domain. As such, non-conservative quantification is impossible. This is a good thing, as probably all natural language determiners are conservative (Keenan & Stavi 1986; Barwise & Cooper 1981).

3 Anaphora as unification

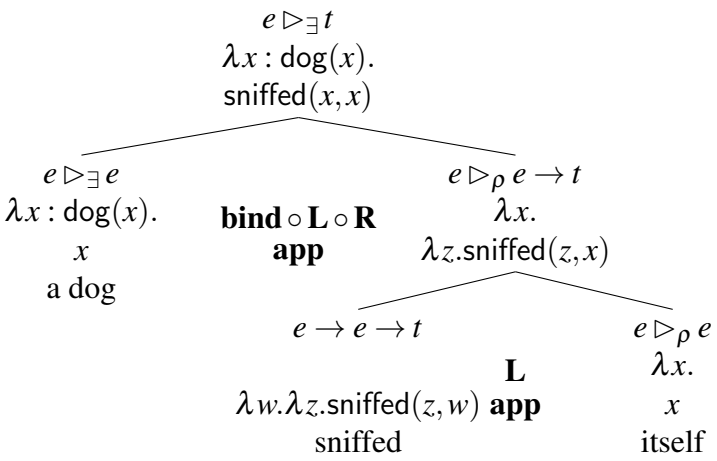
Anaphora is handled in the manner of Jacobson 1999: by treating pronouns as identity functions. A pronoun is an identity function of type $e \rhd_{\rho} e$, where ρ is a special annotation reserved for pronouns. Unlike quantifiers, pronouns can be bound, which necessitates a special rule for binding them (to be generalized in § 4).

(14) If r is a rule that takes any $a\ ::\ \alpha$ and $b\ ::\ \beta$ to $r(a,b)\ ::\ \delta \rhd_{\chi} \delta \rhd_{\rho} \gamma$, then $\textbf{bind}(r)$ is a rule that takes any $a\ ::\ \alpha$ and $b\ ::\ \beta$ to $\textbf{bind}(r)(a,b) =$
$\lambda x : x \in \textbf{dom}\ r(a,b), x \in \textbf{dom}\ r(a,b)(x).r(a,b)(x)(x)\ ::\ \delta \rhd_{\chi} \gamma$

This makes binding possible at combination. Any time a rule would scope a pronoun over a quantifier, **bind** makes it possible for that pronoun to be bound instead. (15) gives a derivation of the sentence *A dog sniffed itself*.

3 This approach to quantification is very similar to that of Pietroski (2005), who argues that quantifiers are predicates of pluralities of Frege-pairs (roughly, individuals paired with truth-values).

(15)



It should be noted that the present system of binding allows for complete freedom in binding: a pronoun can be bound by anything that scopes higher than it. There is no c-command restriction, nor is there any linear order restriction. The former fact is necessary to the account, as donkey anaphora involves binding without c-command. The latter does require a stipulation to fix, to prevent crossover. One such stipulation would be to prevent any quantifier from scoping over any pronoun to its left, thus preventing binding. While it remains unexplained why such a restriction exists, it only has to be stipulated in one place, rather than both in the rules of binding and in the denotations of each quantifier.

4 Modification and donkey anaphora

A characteristic feature of donkey anaphora is that it involves the presence of an indefinite in the restrictor of a quantificational operator. I propose that this is how donkey quantification arises: an indefinite inside a restrictive modifier does not have to undergo existential closure, and may instead attach itself to whatever that modifier modifies. In order to facilitate this, I propose that modification works in the same way as anaphora, involving the unification of two partial dependencies. The difference is that a modifier, unlike a pronoun, is not devoid of restrictive content.

The idea that modification is like anaphora is not unprecedented. This is more commonly proposed for non-restrictive modification: Del Gobbo 2003 and Nouwen 2007 analyze appositives as involving e-type anaphora and dynamic binding, respectively. For restrictive modification, Wittenburg 1987 and Kiss 2005 make similar proposals to the one I give here in order to account for facts about relative clause extraposition.

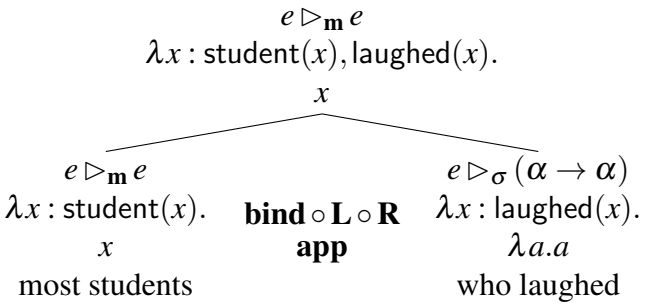
Like pronouns, modifiers are associated with a special annotation, σ. A modifier is something of type $e \rhd_\sigma (\alpha \to \alpha)$, polymorphic in α. It participates in combination

by further restricting the domain of a partial function. For example, *who laughed* has the denotation in (16).

(16) $[\![\text{who laughed}]\!] = \lambda x : \mathsf{laughed}(x).\lambda a.a \;::\; e \rhd_\sigma (\alpha \to \alpha)$

Modification is binding: σ is the same as ρ for the purposes of the **bind** rule. (17) demonstrates the composition of *most students who laughed*.[4]

(17)

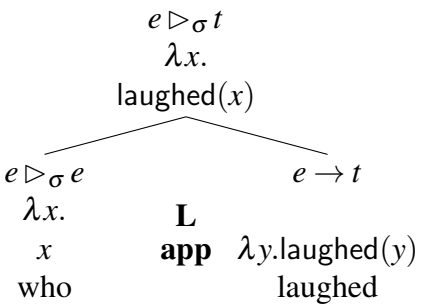


A modifier is formed by first creating something of type $e \rhd_\sigma t$, and then type shifting it to type $e \rhd_\sigma (\alpha \to \alpha)$. Restriction is done through the type shifter **restrict**.

(18) $\textbf{restrict} = \lambda \varphi.\lambda x : \varphi(x).\lambda a.a \;::\; \delta \rhd_\sigma t \to \delta \rhd_\sigma (\alpha \to \alpha)$

This moves content into the restrictor of a partial function. A relative pronoun has the same meaning as an ordinary pronoun, an identity function. For example, the meaning of *who laughed* is derived in (19).

(19) **Step 1:** Make something of type $e \rhd_\sigma t$



Step 2: Apply **restrict**
$\textbf{restrict}(\lambda x.\mathsf{laughed}(x)) = \lambda x : \mathsf{laughed}(x).\lambda a.a \;::\; e \rhd_\sigma \alpha \to \alpha$

4 While the derivation in (17) involves the relative clause attaching after the determiner does, this is not semantically forced: the relative clause can attach to the noun if the noun is of type $e \rhd e$. The difference between these analyses is not relevant for me here, though Lasersohn 2021 makes a case that nouns should be analyzed as restricted variables, which a type like $e \rhd e$ is the variable-free version of.

Donkey anaphora happens when a quantificational element and an indefinite in its restrictor get entangled, restricting each other. In a donkey quantifier like *most students who saw a dog*, quantification is limited to students who have seen dogs, and for each student only dogs who were seen by that student. I model corestriction by having the denotation of a donkey quantifier be a partial function from tuples to individuals. For example, *most students who saw a dog* has a denotation as in (20).

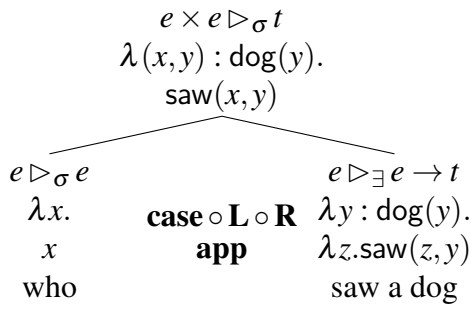
(20) $\lambda(x,y) : \mathsf{student}(x), \mathsf{dog}(y), \mathsf{saw}(x,y).x \ :: \ e \times e \rhd_{\mathbf{m}} e$

An object like this contains all the information that is necessary to do donkey binding, so long as the definition of binding is appropriately generalized to tuples. These arise through an interaction between modification and indefiniteness. When an indefinite is contained within a modifier, it is possible for the indefinite to pair up with the modifier, through a special rule.

(21) If r is a rule that takes any $a \ :: \ \alpha$ and $b \ :: \ \beta$ to $r(a,b) \ :: \ \delta \rhd_\sigma \delta' \rhd_\exists \gamma$, then **case**$(r)$ is a rule that takes any $a \ :: \ \alpha$ and $b \ :: \ \beta$ to **case**$(r)(a,b) =$ $\lambda(x,y) : x \in \mathbf{dom}\ r(a,b), y \in \mathbf{dom}\ r(a,b)(x).r(a,b)(x)(y) \ :: \ \delta \times \delta' \rhd_\sigma \gamma$

This is similar to binding, except instead of unifying two partial dependencies, it puts them side by side. (22) demonstrates the composition of *who saw a dog*.

(22)



To this, **restrict** can apply just as it did in (19), yielding (23).

(23) $[\![\text{who saw a dog}]\!] = \lambda(x,y) : \mathsf{dog}(y), \mathsf{saw}(x,y).\lambda a.a \ :: \ e \times e \rhd \alpha \to \alpha$

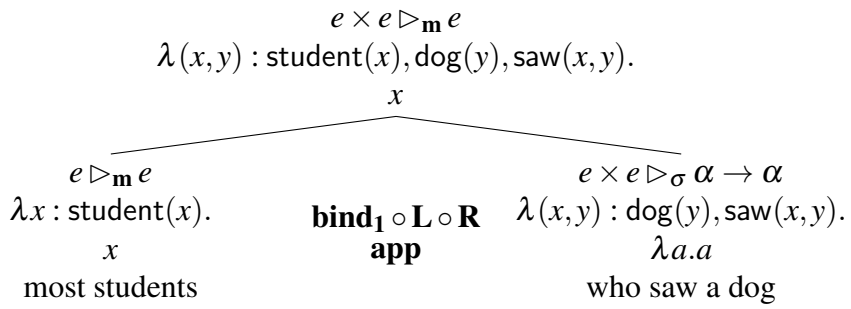
The last formal ingredient necessary here is a generalization of **bind** to tuples. A generalized definition of binding is given in (24).

(24) If r is a rule that takes any $a \ :: \ \alpha$ and $b \ :: \ \beta$ to $r(a,b) \ :: \ \Delta \rhd_\chi (\delta \times \Delta') \rhd_\theta \gamma$, where Δ is a product of types with δ in its nth coordinate and θ is either ρ or σ. Then **bind**$_n(r)$ is a rule that takes any $a \ :: \ \alpha$ and $b \ :: \ \beta$ to **bind**$_n(r)(a,b) =$ $\lambda(\vec{x};\vec{y}) : \vec{x} \in \mathbf{dom}\ r(a,b), (\vec{x}_n;\vec{y}) \in \mathbf{dom}\ r(a,b)(\vec{x}).r(a,b)(\vec{x})(\vec{x}_n;\vec{y})$, which is of type $\Delta \times \Delta' \rhd_\chi \gamma$

This definition features two extensions to (14). First, any coordinate of a tuple can bind: binding from the nth coordinate corresponds to **bind**$_n$. Second, the bindee

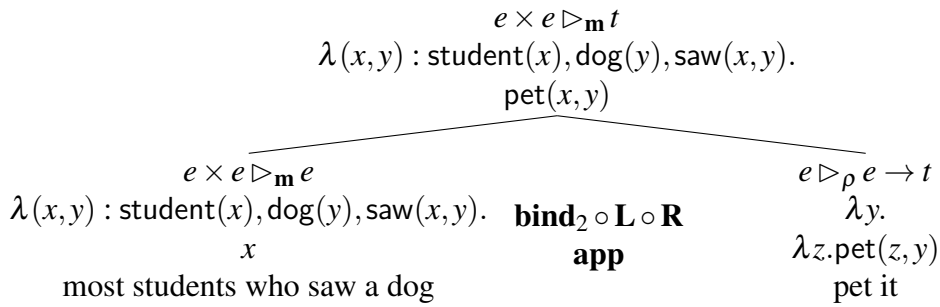
can have extra information outside of its first coordinate, corresponding to donkey antecedents. This information is tacked onto the end after binding occurs. This enables the composition of a donkey quantifier.

(25)



The pieces are also now in place to give an account of donkey binding. A donkey quantifier can bind a pronoun through the use of $\mathbf{bind}_2$ (or in general, $\mathbf{bind}_n$).

(26)



Donkey binding thus proceeds completely separately from quantification.

5 Donkey quantification

The last step in the interpretation of a donkey sentence is quantification, turning something of type $e \times e \rhd_{\chi} t$ to type t. This is not possible with a unary closure operator like the **m** given in (11), whose argument is of type $e \rhd_{\mathbf{m}} t$, not generalized to tuples. What is necessary is a procedure for upgrading operators like this to ones that are capable of quantifying over multiple things at the same time.

The right way to do this is one of the biggest questions in the analysis of donkey anaphora. The simplest answer is to just quantify over each tuple in the same way that one might quantify over each individual. Unfortunately, this gives incorrect truth conditions for proportional quantifiers like *most*—what is known as the proportion problem (Kadmon 1987). Similarly, while universally quantifying over the indefinite often seems correct (the strong reading), it cannot be the only option due to the existence of weak readings, where the indefinite has existential quantificational force (Schubert & Pelletier 1989; Chierchia 1992). I will adopt the approach of

Brasoveanu 2008, positing a strong/weak ambiguity at the level of each indefinite. In the context of the approach here, this involves two type shifters: one for strong readings, one for weak readings.

(27) a. $\textbf{strong} = \lambda q.\lambda\varphi.q(\lambda x : \exists y.(x,y) \in \textbf{dom}\ \varphi.\forall y.(x,y) \in \textbf{dom}\ \varphi \supset \varphi(x,y))$
$:: (\delta \rhd_\chi t \to t) \to \delta \times e \rhd_\chi t \to t$

b. $\textbf{weak} = \lambda q.\lambda\varphi.q(\lambda x : \exists y.(x,y) \in \textbf{dom}\ \varphi.\exists y.(x,y) \in \textbf{dom}\ \varphi \wedge \varphi(x,y))$
$:: (\delta \rhd_\chi t \to t) \to \delta \times e \rhd_\chi t \to t$

These type shifters upgrade a unary closure operator to one that is capable of quantifying over donkeys. Through repeated application, they can upgrade a closure operator to quantify over multiple donkeys. (28) demonstrates **weak** in action, generating the weak reading of (2).

(28) $\textbf{weak}(\mathbf{m}) = \lambda\varphi.\mathbf{m}(\lambda x : \exists y.(x,y) \in \textbf{dom}\ \varphi.\exists y.(x,y) \in \textbf{dom}\ \varphi \wedge \varphi(x,y)) =$
$\lambda\varphi.|\{x \mid \exists y.(x,y) \in \textbf{dom}\ \varphi \wedge \varphi(x,y)\}|$
$> \frac{1}{2}|\{x \mid \exists y.(x,y) \in \textbf{dom}\ \varphi\}| \qquad :: e \times e \rhd_{\mathbf{m}} t \to t$
$\textbf{weak}(\mathbf{m})(\lambda(x,y) : \mathsf{student}(x), \mathsf{dog}(y), \mathsf{saw}(x,y).\mathsf{pet}(x,y)) =$
$|\{x \mid \exists y.\mathsf{student}(x) \wedge \mathsf{dog}(y) \wedge \mathsf{saw}(x,y) \wedge \mathsf{pet}(x,y)\}|$
$> \frac{1}{2}|\{x \mid \exists y.\mathsf{student}(x) \wedge \mathsf{dog}(y) \wedge \mathsf{saw}(x,y)\}| \qquad :: t$

This is true if for all the students who saw at least one dog, the majority of them saw at least one dog and pet that dog, which is the weak reading of (2).

It should be noted that this account of the strong/weak ambiguity is not forced under this approach to donkey binding. Practically any scheme of donkey quantification is definable in a similar way. But unlike a dynamic scheme for lifting a relational determiner (which (3) is an instance of), quantifiers here do not manage anaphoric information.

6 Conditionals

The other classic case of donkey anaphora is from conditionals. In a sentence like (29), an indefinite within the antecedent antecedes a pronoun in the consequent.

(29) If a dog entered, it barked.

Standard analyses of donkey anaphora handle conditionals in the same way as quantifiers: there is an operator that denotes a relation between the antecedent and the consequent, and this operator governs the transmission of anaphoric information from the former to the latter. In general, it is not enough to say that *if* denotes a relation between the antecedent and consequent. *If*-clauses do not contain their own quantificational force, as evidenced by modalized conditionals (as argued in Kratzer 1979, building off Lewis 1975).

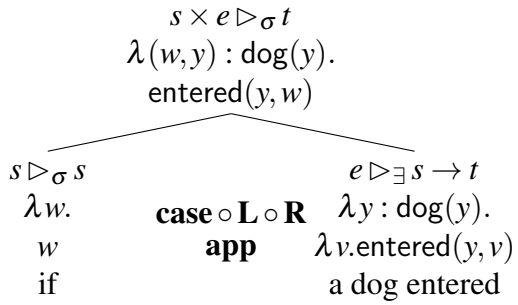
(30) If a dog entered, it might have barked.

In (30), there is existential quantification over cases where a dog entered, unlike (29), where there is universal quantification over them. Heim 1982 handled this by modeling terms like *might* as relations between propositions, rather than *if*. On this view, both the antecedent and the consequent are arguments of a modal. Donkey anaphora can be accounted for by having this operator manage the transmission of anaphoric information.

There is a certain awkwardness to modeling the *if*-clause as a genuine semantic argument, though. *If*-clauses are optional, and moreover, they can be stacked. They behave more like modifiers than genuine arguments. Kratzer's (1981) analysis resolves this mismatch by having *if*-clauses update a contextual parameter that determines the domain of modal quantification. The approach I take here works similarly: an *if*-clause is a restrictive modifier, in the same way as a relative clause. Rather than restricting individuals, it restricts worlds. As donkey anaphora arises in contexts of modification, the same machinery to derive relative clause donkey anaphora also derives conditional donkey anaphora.

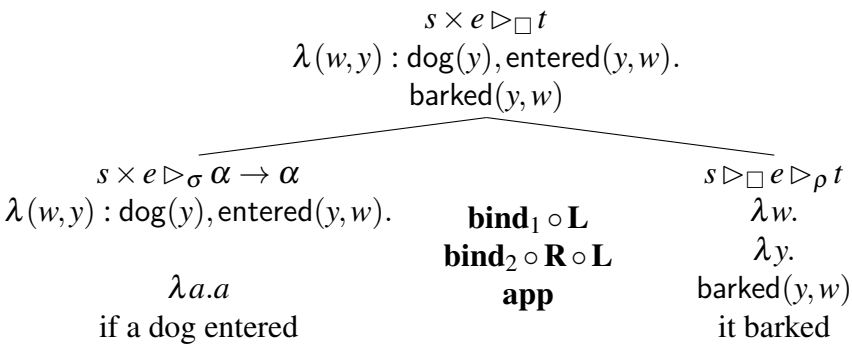
Composing an *if*-clause works in the same way as composing a relative clause. *If* functions as a relative pronoun in the world argument. This means that an indefinite inside an *if*-clause can form a tuple with the world-dependency, as in (31).

(31)



After applying **restrict**, this can participate in binding when it combines with the consequent. Assume that the consequent to a conditional contains a hidden necessity modal, which can be encoded by adding a $s \triangleright_\square$ to its type.

(32)



The mechanics of this are somewhat complicated, since it involves two instances of binding: one for the donkey pronoun, and one for the modifier. The $\mathbf{bind}_2 \circ \mathbf{R} \circ \mathbf{L}$ scopes the modifier over the pronoun and has the second coordinate of the modifier bind the pronoun. The $\mathbf{bind}_1 \circ \mathbf{L}$ then scopes the world-dependency over the modifier and binds it. This derives donkey binding.

The last step in the interpretation of a donkey sentence is once again quantification, and the same strategy holds as with relative clauses: a unary quantificational operator may be upgraded into one that is capable of handling tuples. (33) demonstrates this, finishing the derivation of the strong reading of (29).

(33) $\square = \lambda\varphi.\forall w.w \in \mathbf{dom}\ \varphi \supset \varphi(w)\ ::\ s \rhd_\square t \to t$
$\mathbf{strong}(\square) = \lambda\varphi.\forall w.\exists y.(w,y) \in \mathbf{dom}\ \varphi \supset \forall y.(w,y) \in \mathbf{dom}\ \varphi \supset \varphi(w,y)$
$::\ s \times e \rhd_\square t \to t$
$\mathbf{strong}(\square)(\lambda(w,y) : \mathsf{dog}(y), \mathsf{entered}(y,w).\mathsf{barked}(y,w)) =$
$\forall w.\exists y.\mathsf{dog}(y) \wedge \mathsf{entered}(y,w) \supset \forall y.\mathsf{dog}(y) \wedge \mathsf{entered}(y,w) \supset \mathsf{barked}(y,w)\ ::\ t$

This derives appropriate truth conditions for a sentence with conditional donkey anaphora (modulo other restrictions on the domain of world-quantification).

7 Conclusion

Here I have shown that it is possible to give an analysis of donkey anaphora which does not hard-code management of anaphoric information into the semantics of determiners. What is necessary to do this is separating the introduction of the domain of quantification from the quantificational force, allowing for donkey indefinites to hook onto this domain before quantification happens. In locating the cause of donkey anaphora as an interaction between indefiniteness and modification, I also gave an analysis of conditional donkey anaphora which avoids analyzing modals as binary operators.

References

Barker, Chris & Chung-chieh Shan. 2008. Donkey anaphora is in-scope binding. *Semantics and Pragmatics* 1. 1–46. doi:10.3765/sp.1.1.

Barker, Chris & Chung-chieh Shan. 2014. *Continuations and Natural Language* (Oxford Studies in Theoretical Linguistics 53). New York, NY: Oxford University Press 1st edn.

Barwise, Jon & Robin Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy* 4. 159–219. doi:10.1093/oso/9780195136975.003.0023.

Brasoveanu, Adrian. 2008. Donkey pluralities: Plural information states versus non-atomic individuals. *Linguistics and Philosophy* 31(2). 129–209. doi:10.1007/s10988-008-9035-0.

Brasoveanu, Adrian & Donka F. Farkas. 2011. How indefinites choose their scope. *Linguistics and Philosophy* 34. 1–55. doi:10.1007/s10988-011-9092-7.

Bumford, Dylan. 2017. Split-scope definites: Relative superlatives and Haddock descriptions. *Linguistics and Philosophy* 40. 549–593. doi:10.1007/s10988-017-9210-2.

Bumford, Dylan & Simon Charlow. 2024. *Effect-Driven Interpretation: Functors for Natural Language Composition*. Oxford University Press (to appear).

Chierchia, Gennaro. 1992. Anaphora and dynamic binding. *Linguistics and Philosophy* 15. 111–183. doi:10.1007/bf00635805.

Del Gobbo, Francesca. 2003. Appositives and Quantification. *University of Pennsylvania Working Papers in Linguistics* 9(1). 7.

Elbourne, Paul. 2001. E-type anaphora as NP-deletion. *Natural Language Semantics* 9(3). 241–288. doi:10.1023/A:1014290323028.

Heim, Irene. 1982. *The Semantics of Definite and Indefinite Noun Phrases*: University of Massachusetts Amherst PhD dissertation.

Heim, Irene. 1990. E-type pronouns and donkey anaphora. *Linguistics and Philosophy* 13. 137–177. doi:10.1007/bf00630732.

Jacobson, Pauline. 1999. Towards a Variable-Free Semantics. *Linguistics and Philosophy* 22(2). 117–184. doi:10.1023/A:1005464228727.

Jäger, Gerhard. 2001. Indefinites and Sluicing. A type logical approach. In Robert van Rooy & Martin Stokhof (eds.), *The 13th Amsterdam Colloquium*, 114–119. ILLC, University of Amsterdam.

Kadmon, Nirit. 1987. *On Unique and Non-Unique Reference and Asymmetric Quantification*: University of Massachusetts Amherst PhD dissertation.

Keenan, Edward L. & Jonathan Stavi. 1986. A semantic characterization of natural language determiners. *Linguistics and Philosophy* 9(3). 253–326. doi:10.1007/BF00630273.

Kiss, Tibor. 2005. Semantic constraints on relative clause extraposition. *Natural Language & Linguistic Theory* 23(2). 281–334. doi:10.1007/s11049-003-1838-7.

Kratzer, Angelika. 1979. Conditional necessity and possibility. In *Semantics from Different Points of View*, 117–147. Springer.

Kratzer, Angelika. 1981. The notional category of modality. In Hans-Jürgen Eikmeyer & Hannes Rieser (eds.), *Words, Worlds, and Contexts: New Approaches to Word Semantics*, 38–74. De Gruyter.

Lasersohn, Peter. 2021. Common nouns as modally non-rigid restricted variables. *Linguistics and Philosophy* 44(2). 363–424. doi:10.1007/s10988-019-09293-4.

Lewis, David. 1975. Adverbs of quantification. In Edward Keenan (ed.), *Formal Semantics of Natural Language*, vol. 178, 3–15. Cambridge: Cambridge University Press.

Muskens, Reinhard. 1996. Combining Montague semantics and discourse representation. *Linguistics and Philosophy* 19(2). 143–186. doi:10.1007/BF00635836.

Nouwen, Rick. 2007. On appositives and dynamic binding. *Research on language and computation* 5. 87–102. doi:10.1007/s11168-006-9019-6.

Pietroski, Paul M. 2005. *Events and Semantic Architecture*. Oxford: Oxford University Press.

Schubert, Lenhart K. & Francis Jeffry Pelletier. 1989. Generically speaking, or, using discourse representation theory to interpret generics. In *Properties, Types and Meaning*, vol. 2 Studies in Linguistics and Philosophy, 193–268. Springer.

Wittenburg, Kent. 1987. Extraposition from NP as Anaphora. In Geoffrey J. Huck & Almerindo E. Ojeda (eds.), *Discontinuous Constituency* (Syntax & Semantics 20), 427–445. New York: Academic Press.

Anand Abraham
335 Portola Plaza
3125 Campbell Hall
Los Angeles, CA 90095-1543
aabr@ucla.edu