Notes on a situation-free fragment for donkey anaphora

Chris Barker, *NYU*, `homepages.nyu.edu/~cb125`

Introduction. These notes were produced at the request of Chris Kennedy, for the purpose of providing something to look at in advance of a visit to Chicago on 25 May. My remarks will have two main goals: a critique the D-type approach to donkey anaphora, as championed by Elbourne in his excellent 2005 book; and a presentation of an alternative analysis based on continuations, as developed by me and my collaborator Chung-chieh Shan (Rutgers). These notes will expire at the end of May 2007. If you want a more up-to-date version, please contact me at chris.barker@nyu.edu. Feel free to ask questions in advance of the visit—

My comments on Elbourne will not be difficult to follow, especially if you are familiar with the literature on donkey anaphora. Elbourne's first chapter provides a superb general discussion that introduces most of the issues and much of the data I will be evaluating. The presentation of the Barker-Shan fragment, however, will be more challenging to follow, since it will be unfamiliar to most people in the audience. Therefore this note will concentrate on presenting the fragment, up to the traditional donkey sentence.

The main competitors to the D-type analysis considered by Elbourne are Dynamic Predicate Logic (DPL), and Jacobson's variable-free framework. One of the more telling objections to the DPL approach involves disjustion (e.g., *If a farmer owns a donkey or a goat, he beats nit*), and I will discuss disjunctive examples in some detail. I will suggest that although Elbourne's criticisms of DPL are telling, they involve aspects of the DPL system that are not essential to a dynamic approach. Therefore Elbourne does not argue that dynamic accounts of anaphora are misguided, and in fact, I will suggest that anaphora is clearly dynamic. The fragment I will present is dynamic in the relevant sense.

There is a presentation of the current fragment in a more developed paper available on the semantics archive ('Reconstruction as delayed evaluation'). For the conceptual motivation for continuations, see my 2002 Natural Language Semantics paper ('Continuations and the nature of quantification'). For a discussion of the fragment presented below to binding and weak crossover, see Shan and Barker's 2006 paper in *Linguistics and Philosophy* ('Explaining crossover and superiority as left-to-right evaluation').

Simple combination. We'll stack information about each expression like this:

$$\begin{array}{cc} \text{DP} & \text{syntactic category} \\ \textit{John} & \text{expression} \\ \mathsf{j} & \text{semantic value} \end{array}$$

In the simplest case, syntactic combination reduces to standard combinatory categorial grammar:

$$\left(\begin{array}{cc} \text{DP} & \text{DP}\backslash\text{S} \\ \textit{John} & \textit{left} \\ \mathsf{j} & \mathbf{left} \end{array}\right) = \begin{array}{c} \text{S} \\ \textit{John left} \\ \mathbf{left}\,\mathsf{j} \end{array}$$

As usual, the category under the slash (here, "DP") cancels with the category of the argument expression; the semantics is functional application.

Quantification: Quantificational expressions have an extra layer on top of their syntactic category (likewise on top of their semantic value):

$$\begin{array}{c} \dfrac{\text{S} \mid \text{S}}{\text{DP}} \\ \textit{everyone} \\ \dfrac{\forall y[\,]}{y} \end{array}$$

Read the syntactic category counterclockwise:

$$\dfrac{\text{S} \mid \text{S}}{\text{DP}} \quad \text{means} \quad \dfrac{\text{...to form an S.} \mid \text{and takes scope over an S...}}{\text{Replaces a DP,}}$$

For example:

$$\left(\begin{array}{cc} \dfrac{\text{S} \mid \text{S}}{\text{DP}} & \dfrac{\text{S} \mid \text{S}}{\text{DP}\backslash\text{S}} \\ \textit{everyone} & \textit{left} \\ \dfrac{\forall y[\,]}{y} & \dfrac{[\,]}{\mathbf{left}} \end{array}\right) = \begin{array}{c} \dfrac{\text{S} \mid \text{S}}{\text{S}} \\ \textit{everyone left} \\ \dfrac{\forall y[\,]}{\mathbf{left}\,y} \end{array}$$

In the general case, we have:

$$\left(\begin{array}{cc} \dfrac{\text{C} \mid \mathbf{D}}{\text{A}/\mathbf{B}} & \dfrac{\mathbf{D} \mid \text{E}}{\mathbf{B}} \\ \textit{left} & \textit{right} \\ \dfrac{g[\,]}{f} & \dfrac{h[\,]}{x} \end{array}\right) = \begin{array}{c} \dfrac{\text{C} \mid \text{E}}{\text{A}} \\ \textit{left right} \\ \dfrac{g[h[\,]]}{f(x)} \end{array}$$

Below the horizontal line, combination is simple combinatory categorial grammar (i.e., below the line, we have A/B combining with B to form an A; in the semantics, f combines with x to form $f(x)$).
Above the line is likewise a cancellation operation: here, the D on the left cancels with the D on the right. Semantically, we insert the rightmost meaning element inside the brackets of the leftmost meaning element (i.e., $g[\,]$ combines with $h[\,]$ to form $g[h[\,]]$). The fact that the meaning of the leftmost element surrounds the meaning of the rightmost element expresses the generalization that the default order of semantic evaluation is left-to-right.

Type shifter 1 of 3: Lift: Comparing the analysis above of *John left* with *Everyone left* reveals two distinct analyses of *left*. The simpler one is the basic lexical entry, and the more complex one that participates in a quantificational sentence is derived through a standard type-shifter called Lift.

$$\begin{array}{c} \mathrm{DP\backslash S} \\ \textit{left} \\ \mathbf{left} \end{array} \quad \overset{\text{Lift}}{\Rightarrow} \quad \begin{array}{c} \dfrac{\mathrm{S} \mid \mathrm{S}}{\mathrm{DP\backslash S}} \\ \textit{left} \\ \dfrac{[\,]}{\mathbf{left}} \end{array}$$

In general:

$$\boxed{\begin{array}{c} \mathrm{A} \\ \textit{expression} \\ x \end{array} \quad \overset{\text{Lift}}{\Rightarrow} \quad \begin{array}{c} \dfrac{\alpha \mid \alpha}{\mathrm{A}} \\ \textit{expression} \\ \dfrac{[\,]}{x} \end{array}}$$

Syntactically, Lift adds a layer with arbitrary (but matching!) syntactic categories. Semantically, it adds empty brackets, which amount to a null semantic operation (i.e., application of the identity function); we often will omit writing "[ ]" in the semantic values.

Type shifter 2 of 3: Lower: The final semantic value given above for *Everyone left* was $\dfrac{\forall x.[\,]}{\mathbf{left}\,x}$. In order to arrive at a traditional representation (i.e., one that is not split across two levels), we introduce the type-shifter Lower:

$$\frac{\alpha \mid S}{S} \quad \overset{\text{Lower}}{\Rightarrow} \quad \alpha$$

expression $\Rightarrow$ *expression*

$$\frac{f[\,]}{x} \quad \overset{\text{Lower}}{\Rightarrow} \quad f[x]$$

It is syntactically important that Lower is more specific than Lift: Lower applies only when the two matching categories are S. Applying Lower to the analysis above for *Everyone left* gives

$$\frac{S \mid S}{S} \quad \overset{\text{Lower}}{\Rightarrow} \quad S$$

everyone left $\Rightarrow$ *everyone left*

$$\frac{\forall y[\,]}{\mathbf{left}\,y} \quad \overset{\text{Lower}}{\Rightarrow} \quad \forall y.\mathbf{left}\,y$$

Lower collapses at two-level meaning into a single level by inserting the semantic material below the line into the brackets above the line. Note that this substitution is variable-capturing! Further technical details are given in my manuscript 'Reconstruction as delayed evaluation' (available at semanticsarchive.net).

Pop quiz: Compute the syntactic category and the semantic value of the following sentence:

$$\begin{array}{c} \frac{S \mid S}{DP} \\ \textit{someone} \\ \frac{\exists x[\,]}{x} \end{array} \left(\begin{array}{c} \frac{S \mid S}{(DP\backslash S)/DP} \\ \textit{loves} \\ \frac{[\,]}{\mathbf{loves}} \end{array} \quad \begin{array}{c} \frac{S \mid S}{DP} \\ \textit{everyone} \\ \frac{\forall y[\,]}{y} \end{array} \right) = \textit{S.o. loves e.o.} \overset{\text{Lower}}{\Rightarrow} \textit{S.o. loves e.o.}$$

Show your cancellations. What is the relative scoping of the two quantifiers? Hint: proceed stepwise, following the syntax (do the VP first).

Pronouns: Pronouns create functional dependendies in the way argued for by Jacobson in recent work.

$$\begin{array}{c} \frac{\alpha \mid \alpha}{DP} \\ \textit{he} \\ \frac{\lambda y[\,]}{y} \end{array}$$

For instance, we have

$$\begin{array}{c}\mathrm{DP}\rhd\mathrm{S}\,|\,\mathrm{S}\\ \hline \mathrm{DP}\\ \textit{he}\\ \lambda y[\,]\\ \hline y\end{array}\quad \begin{array}{c}\mathrm{S}\,|\,\mathrm{S}\\ \hline \mathrm{DP}\backslash\mathrm{S}\\ \textit{left}\\ [\,]\\ \hline \mathbf{left}\end{array}\quad = \quad \begin{array}{c}\mathrm{DP}\rhd\mathrm{S}\,|\,\mathrm{S}\\ \hline \mathrm{S}\\ \textit{He left}\\ \lambda y[\,]\\ \hline \mathbf{left}\,y\end{array}\quad \overset{\text{Lower}}{\Rightarrow}\quad \begin{array}{c}\mathrm{DP}\rhd\mathrm{S}\\ \textit{He left}\\ \lambda y.\mathbf{left}\,y\end{array}$$

Type shifter 3 of 3: Binding: Binding allows an arbitrary DP to control the value of a pronoun—provided (only very roughly!) that the DP precedes the pronoun.

$$\boxed{\begin{array}{c}\alpha\,|\,\beta\\ \hline \mathrm{DP}\\ \textit{expression}\\ f[\,]\\ \hline x\end{array}\quad \overset{\text{Bind}}{\Rightarrow}\quad \begin{array}{c}\alpha\,|\,\mathrm{DP}\rhd\beta\\ \hline \mathrm{DP}\\ \textit{expression}\\ f([\,]x)\\ \hline x\end{array}}$$

Syntactically, the binding rule annotates the top right syntactic category with “DP▷”, announcing the presence of a binder for any downstream pronouns. Semantically, it makes a copy of the value of the DP and makes it available to the downstream pronoun. For instance, applying Bind to *everyone*, we have

$$\begin{array}{c}\mathrm{S}\,|\,\mathrm{S}\\ \hline \mathrm{DP}\\ \textit{everyone}\\ \forall x.[\,]\\ \hline x\end{array}\quad \overset{\text{Bind}}{\Rightarrow}\quad \begin{array}{c}\mathrm{S}\,|\,\mathrm{DP}\rhd\mathrm{S}\\ \hline \mathrm{DP}\\ \textit{everyone}\\ \forall x.([\,]x)\\ \hline x\end{array}$$

This gives us

$$\begin{array}{c}\mathrm{S}\,|\,\mathrm{DP}\rhd\mathrm{S}\\ \hline \mathrm{DP}\\ \textit{everyone}\\ \forall x.([\,]x)\\ \hline x\end{array}\left(\begin{array}{c}\mathrm{DP}\rhd\mathrm{S}\,|\,\mathrm{DP}\rhd\mathrm{S}\\ \hline (\mathrm{DP}\backslash\mathrm{S})/\mathrm{DP}\\ \textit{loves}\\ \\ \hline \mathbf{loves}\end{array}\left(\begin{array}{c}\mathrm{DP}\rhd\mathrm{S}\,|\,\mathrm{S}\\ \hline \mathrm{DP}\\ \textit{his}\\ \lambda y[\,]\\ \hline y\end{array}\quad \begin{array}{c}\mathrm{S}\,|\,\mathrm{S}\\ \hline \mathrm{DP}\backslash\mathrm{DP}\\ \textit{mother}\\ \\ \hline \mathbf{mom}\end{array}\right)\right)$$

$$= \begin{array}{c} \dfrac{\mathrm{S} \mid \mathrm{S}}{\mathrm{S}} \\ \forall\ \textit{loves his mom} \\ \dfrac{\forall x.([\lambda y.[\,]]x)}{\mathbf{loves}(\mathbf{mom}\,y)\,x} \end{array} \overset{\text{Lower}}{\Rightarrow} \begin{array}{c} \mathrm{S} \\ \forall \textit{lhm} \\ \forall x.([\lambda y.[\mathbf{loves}(\mathbf{mom}\,y)\,x]]x) \end{array}$$

This lowered value reduces to $\forall x.\mathbf{loves}(\mathbf{mom}\,x)\,x$.

Binding out of DP:

$$\left(\begin{array}{c} \dfrac{\mathrm{S} \mid \mathrm{DP} \triangleright \mathrm{S}}{\mathrm{DP}} \\ \text{everyone's} \end{array} \quad \begin{array}{c} \dfrac{}{\mathrm{DP} \backslash \mathrm{DP}} \\ \text{mother} \end{array} \right) \left(\begin{array}{c} \dfrac{}{(\mathrm{DP} \backslash \mathrm{S})/\mathrm{DP}} \\ \text{loves} \end{array} \quad \begin{array}{c} \dfrac{\mathrm{DP} \triangleright \mathrm{S} \mid \mathrm{S}}{\mathrm{DP}} \\ \text{him} \end{array} \right)$$

Interpretation: $\forall y.\mathbf{loves}\,y(\mathbf{mom}\,y)$. Nothing special need be said. Thus in this system, c-command is not required for binding.

Weak crossover:

$$\left(\begin{array}{c} \dfrac{\mathrm{DP} \triangleright \mathrm{S} \mid \mathrm{S}}{\mathrm{DP}} \\ \text{his} \end{array} \quad \begin{array}{c} \dfrac{}{\mathrm{DP} \backslash \mathrm{DP}} \\ \text{mother} \end{array} \right) \left(\begin{array}{c} \dfrac{}{(\mathrm{DP} \backslash \mathrm{S})/\mathrm{DP}} \\ \text{loves} \end{array} \quad \begin{array}{c} \dfrac{\mathrm{S} \mid \mathrm{DP} \triangleright \mathrm{S}}{\mathrm{DP}} \\ \text{everyone} \end{array} \right) = \begin{array}{c} \dfrac{\mathrm{DP} \triangleright \mathrm{S} \mid \mathrm{DP} \triangleright \mathrm{S}}{\mathrm{S}} \\ \text{Hml}\forall \end{array}$$

The problem with this final category is that it cannot be lowered into a single-level analysis. The reason is that the trigger for the Lower rule isn't met, since lowering requires the top right category to be S. The pronoun is looking leftward for an antecedent, the binder is looking rightward for a pronoun to bind, and they never find each other.

Donkey Conditionals:

$$\begin{array}{c} \\ \hline \\ \hline (S/S)/S \\ \textit{if} \\ \hline \\ \hline \mathbf{if} \end{array}
\left(\begin{array}{c} S \mid DP \triangleright S \\ \hline \\ \hline DP \\ \textit{a farmer} \\ \exists x.(\mathbf{far}\,x) \wedge ([\,]x) \\ \hline \\ \hline x \end{array}
\begin{array}{c} \\ \hline \\ \hline (DP\backslash S)/DP \\ \textit{owns} \\ \hline \\ \hline \mathbf{owns} \end{array}
\begin{array}{c} \\ \hline S \mid DP \triangleright S \\ \hline DP \\ \textit{a donkey} \\ \\ \hline \exists y.(\mathbf{don}\,y) \wedge ([\,]y) \\ \hline y \end{array} \right)
\left(\begin{array}{c} DP \triangleright S \mid S \\ \hline \\ \hline DP \\ \textit{he} \\ \lambda z.[\,] \\ \hline \\ \hline z \end{array}
\begin{array}{c} \\ \hline \\ \hline (DP\backslash S)/DP \\ \textit{beats} \\ \hline \\ \hline \mathbf{beats} \end{array}
\begin{array}{c} \\ \hline DP \triangleright S \mid S \\ \hline DP \\ \textit{it} \\ \hline \lambda w.[\,] \\ \hline w \end{array} \right)$$

$$= \begin{array}{c} S \mid S \\ \hline S \mid S \\ \hline S \\ \textit{If a farmer owns a donkey he beats it} \\ \exists x.(\mathbf{far}\,x) \wedge ([\lambda z.[\,]]x) \\ \hline \exists y.(\mathbf{don}\,y) \wedge ([\lambda w.[\,]]y) \\ \hline \mathbf{if}(\mathbf{owns}\,y\,x)(\mathbf{beats}\,w\,z) \end{array}$$

$$\underset{\Rightarrow}{\text{Lower}} \begin{array}{c} S \mid S \\ \hline S \\ \textit{If a farmer owns a donkey he beats it} \\ \exists x.(\mathbf{far}\,x) \wedge ([\lambda z.[\,]]x) \\ \hline \exists y.(\mathbf{don}\,y) \wedge ([\lambda w.[\mathbf{if}(\mathbf{owns}\,y\,x)(\mathbf{beats}\,w\,z)]]y) \end{array} = \begin{array}{c} S \mid S \\ \hline S \\ \textit{If a farmer owns a donkey he beats it} \\ \exists x.(\mathbf{far}\,x) \wedge ([\lambda z.[\,]]x) \\ \hline \exists y.(\mathbf{don}\,y) \wedge [\mathbf{if}(\mathbf{owns}\,y\,x)(\mathbf{beats}\,y\,z)] \end{array}$$

$$\underset{\Rightarrow}{\text{Lower}} \begin{array}{c} S \\ \textit{If a farmer owns a donkey he beats it} \\ \exists x.(\mathbf{far}\,x) \wedge ([\lambda z.[\exists y.(\mathbf{don}\,y) \wedge [\mathbf{if}(\mathbf{owns}\,y\,x)(\mathbf{beats}\,y\,z)]]]x) \end{array}$$

$= \exists x.(\mathbf{far}\,x) \wedge [\exists y.(\mathbf{don}\,y) \wedge [\mathbf{if}(\mathbf{owns}\,y\,x)(\mathbf{beats}\,y\,x)]]$

Bishop sentences work exactly the same way (*If a bishop meets a bishop, he blesses him*)

What is missing: inverse scope (inner application of Lift). Scope bounding and scope islands. Binding of multiple pronouns. Generalized disjunction, existential disjunction.