Donkeys in DyCG without Discourse Referents

Carl Pollard

Linguistics 812
April 4, 2011

(1) **DyCG Overview**

Dynamic Categorial Grammar (DyCG) is a framework for linguistic analysis that integrates and builds on results from three distinct research traditions:

- the *curryesque* tradition in categorial grammar that distinguishes **phenogrammar** (concrete syntax) from **tectogrammar** (abstract syntax).
- the *hyperintensional* tradition in (static) semantics that takes propositions rather than worlds as basic
- the *dynamic* tradition in discourse semantics, where utterance interpretation depends on and transforms context

(2) **Salient Features of DyCG**

- The underlying static meanings are fine-grained **hyperintensions**, rather than the usual Montague-style intensions.
- Worlds, intensions, and extensions can be defined within the semantic theory, but grammars needn't refer to them.
- There is no extensional version of the theory.
- Heim's Logical Forms are replaced by DyCG sentences (theorems of the grammar of tectogrammatical type S).
- The semantic theory is written in a type theory (like Muskens or de Groote), not in the metalanguage (like Heim's file change semantics).
- Sentence meanings specified by the grammar are (adapting Roberts' terminology) **proffered contents**, what is to be added to the input context in case of acceptance.
- In case of acceptance, the input context is transformed by the context change obtained by applying the context change function cc to the proffered content.

(3) **This Fragment**

- This fragment uses a new, simplified version of DyCG in which anchors (essentially, Heim's assignments) are replaced with tuples of entities.
- Thus the role of discourse referents (DR's) is taken over by the linear positions in the tuples in the domains of contexts.
- Unlike previous versions of the grammar, the subject of a verb is taken to be the first rather than the last argument. This simplifies most derivations, since all NPs have raised types and usually their relative scope coincides with their linear order.
- The fragment analyzes the classic donkey sentence *every farmer that owns a donkey beats it* with the strong reading (every farmer beats every donkey s/he owns).
- Weak readings and the proportion problem with *most farmers that own a donkey beat it* will be dealt with another day.
- Since we are focusing on the semantics, we finesse the pragmatic issue of anaphora resolution using the 'cheap trick' (*à la* Montague, Heim, and Muskens) of treating pronouns as ambiguous with respect to which DR is the antecedent.
- Anaphora resolution too will be dealt with another day.
- However, we do handle the *semantic* aspect of pronominal presupposition, namely that the antecedent be practically entailed by the context to satisfy the descriptive content of the pronoun (in the case of *it*, being nonhuman).
- New Notation
  - We write the natural number type as n, and reserve ω for the metalanguage name of the set of natural numbers in the ambient set theory.
  - We use vector notation for tuples of entities:

    $$\mathbf{x}^n = (x_0, \ldots, x_{n-1}) \text{ for } n > 0$$

    with the superscript indicating the length of the tuple only on the leftmost occurrence of the tuple in a formula.

(4) **Types from HOL**

t (truth values, a basic type)

n (naturals, a basic type)

Variables: i, j, n

The n-th cartesian power of A is written A^n, thus $A^0 = 1$, $A^1 = A$, and $A^{n+1} = A^n \times A$ $(n > 0)$

(5) **Pheno Types**

s (strings, a basic type, axiomatized as a monoid)

Variables: s, t, u

N.B. Constants of type s include the null string **e** and prime strings, e.g. DONKEY.

$\mathrm{s} \to \mathrm{s}$ (string functions)

Variables: f, g

$\mathrm{s} \to \mathrm{s} \to \mathrm{s}$ (the type of the constant $\cdot$ (concatenation), written infix)

(6) **Types from (Static) Hyperintensional Semantics**

e (entities, a basic type)

Variables: x, y, z

p (propositions, a basic type)

Variables: p, q

$\mathrm{p}_0 =_{\mathrm{def}} \mathrm{p}$ (0-ary properties)

$\mathrm{p}_{n+1} =_{\mathrm{def}} \mathrm{e} \to \mathrm{p}_n$ (n-ary properties, $n > 0$)

Variables of type p_1: P, Q

$\mathrm{p}_1 \to \mathrm{p}$ (static generalized quantifiers)

$\mathrm{p}_1 \to \mathrm{p}_1 \to \mathrm{p}$ (static generalized determiners)

(7) **Types for Contexts**

$\mathrm{c}_0 =_{\mathrm{def}} \mathrm{p}$ (0-ary contexts)

$\mathrm{c}_n =_{\mathrm{def}} \mathrm{e}^n \to \mathrm{p}$ (n-ary contexts, $n > 0$)

$\mathrm{c} =_{\mathrm{def}} \coprod_{n \in \omega} .\mathrm{c}_n$ (contexts)

Variables: c, d

(8) **Dynamic Semantic Types**

$\mathrm{k} =_{\mathrm{def}} \mathrm{c} \rightharpoonup \mathrm{c}$ (proffered contents and context changes)
Variables: k, h
These correspond to what other dynamic semantic theories variously call file change potentials, boxes, updates, or dynamic propositions.

$\mathrm{d}_0 =_{\mathrm{def}} \mathrm{k}$ (0-ary dynamic properties)

$\mathrm{d}_{n+1} =_{\mathrm{def}} \mathrm{n} \rightarrow \mathrm{d}_n$ (n-ary dynamic properties, $n > 0$)

$\mathrm{d} =_{\mathrm{def}} \mathrm{d}_1$ ((unary) dynamic properties)
Variables: D, E

$\mathrm{d} \rightarrow \mathrm{k}$ (dynamic generalized quantifiers)

$\mathrm{d} \rightarrow \mathrm{d} \rightarrow \mathrm{k}$ (dynamic generalized determiners)

(9) **The Arity of a Context**

- For an n-ary context c, the **arity** of c, written $|c|$, is n.
- Intuitively, $|c|$ is 'the number of DRs that c knows about'.
- *Example.* $|c| = 2$, where c is the output context from an accepted utterance of *A farmer owns a donkey* in the trivial input context true. Here c is equivalent to

$$\lambda_{x,y}.(\mathsf{farmer}\ x)\ \mathsf{and}\ (\mathsf{donkey}\ y)\ \mathsf{and}\ (\mathsf{own}\ x\ y)$$

(10) **The Degree of a Context Change**

- Context changes will be defined in such a way that, for any context change k, there is a unique natural number called the **degree** of h, also written $|k|$, such that, for any context c in the domain of k, $|k\ c| = |c| + |k|$.
- Intuitively, the degree of a context change is the number of new DRs that it introduces.
- *Example.* $|\textsc{cold}| = 0$, where $\textsc{cold} =_{\mathrm{def}} \lambda_c.\lambda_{\mathbf{x}^{|c|}}.\mathsf{cold}$ is the translation of *it is cold*.
  Example. $|\textsc{exists donkey}| = 1$, where (as shown later)

$$\textsc{exists donkey} = \lambda_c.\lambda_{\mathbf{x}^{|\mathbf{c}|},y}.\mathsf{donkey}\ y$$

  is the translation of Chinese *you lü* 'there is a donkey'.

(11) **Composition of Partial Functions**

- For any three types A, B, C, the composition $f; g : A \rightharpoonup B$ of any two partial functions $f : A \rightharpoonup B$ and $g : B \rightharpoonup C$ is

$$f; g =_{\text{def}} \lambda_{a|(f\downarrow a)\wedge(g\downarrow(f\ a))}.g\ (f\ a)$$

- And so the composition of two context changes h and k is

$$k; h =_{\text{def}} \lambda_{c|(k\downarrow c)\wedge(h\downarrow(k\ c))}.h\ (k\ c)$$

(12) **The Context Change of a Proffered Content**

- The **context change** function $\mathsf{cc} : \mathrm{k} \rightarrow \mathrm{k}$, which maps every proferred content to a context change of the same degree, is defined as follows:

$$\mathsf{cc}\ k =_{\text{def}} \lambda_{c|k\downarrow c}.\lambda_{\mathbf{x}^{|c|},\mathbf{y}^{|k|}}.(c\ \mathbf{x})\ \mathsf{and}\ \ (k\ c\ \mathbf{x}, \mathbf{y}))$$

- First conjunct is the 'carryover' from the input context c.
- Second conjunct is determined by the proffered content k.
- The motivation for this will become clearer when we look at examples with indefinites in them.
- *Example: It is cold* $\rightsquigarrow$ $\textsc{cold} =_{\text{def}} \lambda_c.\lambda_{\mathbf{x}^{|c|}}.\mathsf{cold}$. Application of cc to this proffered content produces the context change

$$\mathsf{cc}\ \textsc{cold} =_{\text{def}} \lambda_c.\lambda_{\mathbf{x}^{|c|}}.(c\ \mathbf{x})\ \ \mathsf{and}\ \mathsf{cold}.$$

(13) **The Dynamic Conjunction of Proffered Contents (1/2)**

- The **dynamic conjunction** function $\textsc{and} : \mathrm{k} \rightarrow \mathrm{k} \rightarrow \mathrm{k}$ is defined as follows:

$$k\ \textsc{and}\ h =_{\text{def}} \lambda_{c|(k\downarrow c)\wedge(h\downarrow(\mathsf{cc}\ k\ c))}.\lambda_{\mathbf{x}^{|c|},\mathbf{y}^{|k|},\mathbf{z}^{|h|}}.$$
$$(k\ c\ \mathbf{x}, \mathbf{y})\ \mathsf{and}\ (h\ (\mathsf{cc}\ k\ c)\ \mathbf{x}, \mathbf{y}, \mathbf{z})$$

- The usual definitions of dynamic conjunction as composition of context changes makes no reference to static conjunction. Distinguishing proffered contents from their induced context changes makes it easier to see the connection between dynamic and static conjunction.

(14) **The Dynamic Conjunction of Proffered Contents (1/2)**

Theorem: The dynamic conjunction of proffered contents is the composition of their context changes.

$$\vdash \forall_{kh}.\mathsf{cc}\ (k \text{ AND } h) = (\mathsf{cc}\ k);(\mathsf{cc}\ h)$$

Proof hint: show that both sides of the equality are equal to:

$$\lambda_{c|(k\downarrow c)\wedge(h\downarrow(\mathsf{cc}\ k\ c))}.\lambda_{\mathbf{x}^{|c|},\mathbf{y}^{|k|},\mathbf{z}^{|h|}}.$$
$$(c\ \mathbf{x})\ \mathsf{and}\ (k\ c\ \mathbf{x},\mathbf{y})\ \mathsf{and}\ (h\ (\mathsf{cc}\ k\ c)\ \mathbf{x},\mathbf{y},\mathbf{z})$$

(15) **Dynamicization of Properties**

- Dynamic properties that are not presupposition triggers can be defined by applying the **dynamicization** function **dyn** to their static counterparts.
- For each n, $\mathbf{dyn}_n : \mathrm{p}_n \rightarrow \mathrm{d}_n$. For $n = 0, 1, 2$ these are defined as follows:

$$\mathbf{dyn}_0\ p =_{\text{def}} \lambda_c.\lambda_{\mathbf{x}^{|c|}}.p$$
$$\mathbf{dyn}_1\ P =_{\text{def}} \lambda_m.\lambda_{c||c|>m}.\lambda_{\mathbf{x}^{|c|}}.P\ x_m$$
$$\mathbf{dyn}_2\ R =_{\text{def}} \lambda_{mn}.\lambda_{c||c|>m,n}.\lambda_{\mathbf{x}^{|c|}}.R\ x_m\ x_n$$

- *Examples:*

$$\text{COLD} =_{\text{def}} \mathbf{dyn}_0\ \mathsf{cold} = \lambda_c.\lambda_{\mathbf{x}^{|c|}}.\mathsf{cold}$$
$$\text{DONKEY} =_{\text{def}} \mathbf{dyn}_1\ \mathsf{donkey} = \lambda_m.\lambda_{c||c|>m}.\lambda_{\mathbf{x}^{|c|}}.\mathsf{donkey}\ \ x_m$$
$$\text{BEAT} =_{\text{def}} \mathbf{dyn}_2\ \mathsf{beat} = \lambda_{mn}.\lambda_{c||c|>m,n}.\lambda_{\mathbf{x}^{|c|}}.\mathsf{beat}\ x_m\ x_n$$

(16) **Dynamic Property Conjunction**

- In static semantics, property conjunction (involved in relativization and VP-coordination) is defined as follows:

$$P\ \mathsf{that}\ Q =_{\text{def}} \lambda_x.(P\ x)\ \mathsf{and}\ (Q\ x)$$

- Dynamic property conjunction is defined analogously:

$$D\ \text{THAT}\ E =_{\text{def}} \lambda_n.(D\ n)\ \text{AND}\ (E\ n)$$

(17) **Dynamic Negation of Proffered Contents**

- **Dynamic negation** NOT : k → k is defined by NOT $k =_{\text{def}}$

$$\lambda_{c|k\downarrow c}.\lambda_{\mathbf{x}^{|c|}}.\mathsf{not}\ (k\ c\ \mathbf{x})\ (\text{for}\ |k|=0)$$
$$\lambda_{c|k\downarrow c}.\lambda_{\mathbf{x}^{|c|}}.\mathsf{not}\ (\mathsf{exists}_{\mathbf{y}^{|k|}}.k\ c\ \mathbf{xy})\ (\text{for}\ |k|>0)$$

- *Example. No way it's cold* ⇝

$$\lambda_c.\lambda_{\mathbf{x}^{|c|}}.\mathsf{not}\ \mathsf{cold}$$

- *Example.* Chinese *meiyou lü* 'there isn't a donkey' ⇝

$$\lambda_c.\lambda_{\mathbf{x}^{|c|}}.\mathsf{not}\ (\mathsf{exists}_y.\mathsf{donkey}\ y)$$

  cf. above, *you lü* 'there's a donkey' ⇝ $\lambda_c.\lambda_{\mathbf{x}^{|c|},y}.\mathsf{donkey}\ y$.

- As we'll see, indefinites don't have existential force. But they sometimes appear to, because certain operators (such as NOT) introduce an existential that delimits the lifespans of the DRs introduced by the operator's argument.

(18) **Dynamic Double Negation**

Theorem. $\vdash \forall_k.\,|\text{NOT}\ k| = 0$

Corollary. For proffered contents of degree 0, the dynamic Double Negation Law is satisfied up to equivalence.

$$\vdash \forall_{k\in \mathrm{k}_0}.\text{NOT}\ (\text{NOT}\ k)) \equiv k$$

Corollary. For a proffered content k of positive degree m, dynamic double negation is equivalent to simultaneous existential binding of all m DRs that k introduces.

$$\vdash \forall_{k\in \mathrm{k}_n}.\text{NOT}\ (\text{NOT}\ k) \equiv \lambda_{c|k\downarrow c}.\lambda_{\mathbf{x}^{|c|}}.\mathsf{exists}_{\mathbf{y}^m}.k\ c\ \mathbf{x},\mathbf{y}$$

(19) **Dynamic Property Negation**

- In static semantics, property negation is defined as follows:

$$\mathsf{non}\ P =_{\text{def}} \lambda_x.\mathsf{not}\ (P\ x)$$

- Dynamic property negation is defined analogously:

$$\text{NON}\ D =_{\text{def}} \lambda_n.\text{NOT}\ (D\ n)$$

(20) **(So-Called) Dynamic Existential Quantification**

- First, we define a function $^+ : \mathrm{c} \to \mathrm{c}$ that adds a new DR to an arbitrary context:

$$c^+ =_{\text{def}} \lambda_{\mathbf{x}|c|,y}.c\ \mathbf{x}$$

  Obviously $|c^+| = |c| + 1$.

- Then we define the dynamic generalized quantifier EXISTS:

$$\textsc{exists}\ D =_{\text{def}} \lambda_{c|(D\ |c|)\downarrow c^+}.D\ \ |c|\ \ c^+$$

- Note that in this definition, the new DR $|c|$ depends on c, which is λ-bound, but *not* existentially bound.
- *Exercise.* Recall that

$$\textsc{donkey} = \lambda_n.\lambda_{c||c|>n}.\lambda_{\mathbf{x}|c|}.\mathsf{donkey}\ x_n$$

  and show that

$$\textsc{exists donkey} = \lambda_c.\lambda_{\mathbf{x}|c|,y}.\mathsf{donkey}\ y$$

(21) **The Dynamic Indefinite Determiner**

- In static semantics, the indefinite determiner is defined as follows:

$$\mathsf{a}\ P\ Q =_{\text{def}} \mathsf{exists}\ (P\ \mathsf{that}\ Q)$$

- The dynamic indefinite determiner A is defined analogously:

$$\textsc{a}\ D\ E =_{\text{def}} \textsc{exists}\ (D\ \textsc{that}\ E)$$

- *Exercise:* Show that

$$\textsc{a donkey bray} = \lambda_c.\lambda_{\mathbf{x}|c|,y}.(\mathsf{donkey}\ y)\ \mathsf{and}\ (\mathsf{bray}\ y)$$

- The fact that the restriction and the scope of an indefinite are dynamically conjoined has the consequence that presuppositions of the scope can be satisfied from the restriction, e.g. *A farmer that owns [a donkey]$_i$ beats it$_i$.*

  This will become clearer once we see how a definite pronoun like *it* gives rise to an anaphoric presupposition.

(22) **Toward a Dynamic Universal Determiner**

- Disregarding generic readings, sentence (2) seems ambiguous in a way that sentence (1) isn't:
  (1) A farmer that owns a donkey beats it.
  (2) Every farmer that owns a donkey beats it.
- For (2) to be true, does each farmer have to beat *every* donkey s/he owns, or just one of them?
- Early dynamic approaches to donkey anaphora in terms of **unselective binding** only predicted the first known as the **strong**) reading, not the second (**weak**) reading.
- One way to characterize the difference is to say that in strong readings, the determiner in question (here, *every*, treats (a) the DR introduced by the head noun of the restriction (here, *farmer*) on a par with (b) the DRs introduced by the indefinites in the relative clause (here, *donkey*), whereas in weak readings, (a) and (b) are treated asymmetrically.
- Sometimes the weak reading is the obvious or only one:
  (3) Every man that had a quarter put it in the parking meter.
  (4) Most farmers that own a donkey beat it.
- The usual story says that (4) is false if there are exactly 100 farmers, 99 of them have only one donkey and don't beat it, while the other has 1000 donkeys and beats them all.
- That is, most donkey-owning farmers beat at least one of their donkeys; it doesn't matter that 1000 out of the 1009 farmer-donkey pairs involve human-asinine battery.
- Explaining this asymmetry is known as the **proportion** problem, or alternatively, the problem of **farmer-donkey asymmetry**.
- To get started, we'll propose a dynamic meaning for *every* that yields strong readings.
- But we will revisit this set of issues soon.

(23) **A Strong Dynamic Universal Determiner**

- Recall that in HOL (or FOL):

$$\vdash (\forall_x.\phi) \leftrightarrow \neg(\exists_x.\neg\phi)$$

- Analogously, in static semantics,

$$\vdash \mathsf{forall} \equiv \lambda_P.\mathsf{not}\ (\mathsf{exists}_x.\mathsf{not}\ (P\ x))$$

and

$$\vdash \mathsf{every} \equiv \lambda_{PQ}.\mathsf{forall}_x\ .(P\ x)\ \mathsf{implies}\ (Q\ x)$$

where

$$\vdash\ \mathsf{implies}\ \equiv \lambda_{pq}.\mathsf{not}\ (p\ \mathsf{and}\ (\mathsf{not}\ q)$$

from which we can derive (using the fact that static not obeys the Double Negation Law up to equivalence)

$$\vdash \mathsf{every} \equiv \lambda_{PQ}.\mathsf{not}\ (\mathsf{exists}_x.(P\ x)\ \mathsf{and}\ (\mathsf{not}\ (Q\ x)))$$

or equivalently

$$\vdash \mathsf{every} \equiv \lambda_{PQ}.\mathsf{not}\ (\mathsf{exists}\ (P\ \ \mathsf{that}\ \ (\mathsf{non}\ Q)))$$

- Analogizing again at the dynamic level, we define

$$\textsc{every}\ D\ E =_{\text{def}} \textsc{not exists}_n\ ((D\ n)\ \textsc{and}\ (\textsc{not}\ (E\ n)))$$

or equivalently

$$\textsc{every}\ D\ E =_{\text{def}} \textsc{not exists}\ (D\ \textsc{that}\ (\textsc{non}\ Q))$$

- *Exercise.* Once our grammar is in place, we will have *every donkey brays* $\rightsquigarrow$ EVERY DONKEY BRAY. Show that this can be reduced to

$$\lambda_c.\lambda_{\mathbf{x}^{|c|}}.\mathsf{not}(\mathsf{exists}\ (\mathsf{donkey}\ \mathsf{that}\ (\mathsf{non}\ \mathsf{bray}))) =$$
$$\lambda_c.\lambda_{\mathbf{x}^{|c|}}.\mathsf{not}(\mathsf{exists}_y.(\mathsf{donkey}\ y)\ \mathsf{and}\ (\mathsf{not}\ (\mathsf{bray}\ y))$$

(24) **The Definite Pronoun *it*, Intuitively**

- Here's the intuition:
  (a) The definite pronoun *it* 'picks up' a DR, the 'antecedent', already in the input context (or inferrable from it).
  (b) The antecedent is practically entailed by the context to satisfy the 'descriptive content' of the pronoun (in this case, the property of being nonhuman).
  (c) The speaker believes the context provides enough information for the addressee to resolve *which* DR is the antecedent.
- Here we handle (a) and (b), which are presuppositional in nature, in the lexical semantics of the pronoun.
- For now, we finesse the pragmatic issues posed by (c) by the familiar cheap trick of treating the pronoun as ambiguous with respect to which DR is its antecedent.

(25) **The Definite Pronoun *it*, Formally**

- As a point of departure, suppose we give *it* this lexical entry schema (for $i \in \mathrm{n}$):

$$\vdash \mathrm{IT}; \mathrm{NP}; i$$

  This is simplicity itself, but it doesn't impose on the context the presuppositons corresponding to (a) and (b) above.
- So instead we type-raise the preceding to a dynamic generalized quantifier:

$$\vdash \lambda_f.f\ \mathrm{IT}; (\mathrm{NP} \multimap \mathrm{S}) \multimap \mathrm{S}; \lambda_D.D\ i$$

- And then, finally, we refine the preceding by restricting the domain to satisfy the two presuppositions in question:

$$\vdash \lambda_f.f\ \mathrm{IT}; (\mathrm{NP} \multimap \mathrm{S}) \multimap \mathrm{S};$$
$$\lambda_{c|(|c|>i)\wedge(c\ \mathsf{pentails}\ \lambda_{\mathbf{x}^{|c|}}.\mathsf{nonhuman}\ x_i)}.\lambda_D.D\ i\ c$$

- *Exercise.* Show that IT_i BRAY equals

$$\lambda_{c|(|c|>i)\wedge(c\ \mathsf{pentails}\ \lambda_{\mathbf{x}^{|c|}}.\mathsf{nonhuman}\ x_i)}.\lambda_{\mathbf{x}^{|c|}}.\mathsf{bray}\ x_i$$

(26) **Tectogrammatical Types (Syntactic Categories)**

S; NP (noun phrase); N; D (discourse).
Also, QP abbreviates $(NP \multimap \mathrm{S}) \multimap \mathrm{S}$

(27) **Rules of the Grammar Logic**

Hypothesize/Variable/Trace:

$$\frac{}{x : A; B; y : C \vdash x : A; B; y : C}\ \mathrm{T}$$

Modus Ponens/Application/Merge:

$$\frac{\Gamma \vdash \mathrm{f} : A \to B; C \multimap D; \mathsf{g} : E \to F \qquad \Delta \vdash \mathrm{a} : A; C; \mathsf{b} : E}{\Gamma, \Delta \vdash (\mathrm{f}\ \mathrm{a}) : B; D; (\mathsf{g}\ \mathsf{b}) : F}\ \mathrm{M}$$

Hypothetical Proof/Abstraction/Move:

$$\frac{\Gamma, x : A; B; y : C \vdash \mathrm{d} : D; E; \mathsf{f} : F}{\vdash \lambda_x; \mathrm{d} : A \to D; B \multimap E; \lambda_y.\mathsf{f} : C \to F}\ \mathrm{H}$$

Initiate Discourse:

$$\frac{\vdash \mathrm{s}; \mathrm{S}; \mathsf{k}}{\vdash \mathrm{s}; \mathrm{D}; \mathsf{k}}\ \mathrm{I}$$

Continue Discourse:

$$\frac{\vdash \mathrm{s}; \mathrm{D}; \mathsf{k} \qquad \vdash \mathrm{t}; \mathrm{S}; \mathsf{h}}{\vdash \mathrm{s} \cdot \mathrm{t}; \mathrm{D}; \mathsf{k}\ \textsc{and}\ \mathsf{h}}\ \mathrm{C}$$

(28) **Summary of Lexical Meanings Employed**

$\textsc{every} =_{\mathrm{def}} \lambda_{DE}.\textsc{not exists}\ (D\ \textsc{that}\ (\textsc{non}\ Q))$
$\textsc{farmer} =_{\mathrm{def}} \mathbf{dyn}_1\ \mathsf{farmer} = \lambda_n.\lambda_{c||c|>n}.\lambda_{\mathbf{x}^{|c|}}.\mathsf{farmer}\ x_n$
$\textsc{that} =_{\mathrm{def}} \lambda_{DE}.\lambda_n.(D\ n)\ \textsc{and}\ (E\ n)$
$\textsc{own} =_{\mathrm{def}} \mathbf{dyn}_2\ \mathsf{own} = \lambda_{mn}.\lambda_{c||c|>m,n}.\lambda_{\mathbf{x}^{|c|}}.\mathsf{own}\ x_m\ x_n$
$\textsc{a} =_{\mathrm{def}} \lambda_{DE}.\textsc{exists}\ (D\ \textsc{that}\ E)$
$\textsc{donkey} =_{\mathrm{def}} \mathbf{dyn}_1\ \mathsf{donkey} = \lambda_n.\lambda_{c||c|>n}.\lambda_{\mathbf{x}^{|c|}}.\mathsf{donkey}\ x_n$
$\textsc{beat} =_{\mathrm{def}} \mathbf{dyn}_2\ \mathsf{beat} = \lambda_{mn}.\lambda_{c||c|>m,n}.\lambda_{\mathbf{x}^{|c|}}.\mathsf{beat}\ x_m\ x_n$
$\textsc{it}_i =_{\mathrm{def}} \lambda_{c|(|c|>i)\wedge(c\ \mathsf{pentails}\ \lambda_{\mathbf{x}^{|c|}}.\mathsf{nonhuman}\ x_i)}.\lambda_D.D\ i\ c$

(29) **The Lexical Entries**

a $\vdash \lambda_{sf}.f\ \text{EVERY} \cdot s; \text{N} \multimap \text{QP};$ EVERY
b $\vdash \lambda_{sf}.s \cdot \text{THAT} \cdot (f\ \mathbf{e}); \text{N} \multimap (\text{NP} \multimap \text{S}) \multimap \text{N};$ THAT
c $\vdash \text{FARMER}; \text{N};$ FARMER
d $\vdash \lambda_{sf}.f\ \text{A} \cdot s; \text{N} \multimap \text{QP};$ A
e $\vdash \text{DONKEY}; \text{N};$ DONKEY
f $\vdash \lambda_{st}.s \cdot \text{OWNS} \cdot t; \text{NP} \multimap \text{NP} \multimap \text{S};$ OWN
g $\vdash \lambda_{f}.f\ \text{IT}; \text{QP};$ IT$_i$
h $\vdash \lambda_{st}.s \cdot \text{BEATS} \cdot t; \text{NP}v \multimap \text{NP} \multimap \text{S};$ BEAT

(30) **Notational Conventions for Grammar Proofs**

1. Leaves (other than traces) are keyed to the lexical entries.
2. Only the tectotype of non-leaves are shown.
3. Unlabelled binary rule applications are modus ponens.
4. Unlabelled unary rule applications are hypothetical proof.
5. (t) abbreviates $s; \text{NP}; x \vdash s; \text{NP}; x$ (trace of NP)
6. Withdrawal of traces is indicated (Prawitz-style) by indices.

(31) **Grammar Proof for the Donkey Sentence**

$$\dfrac{\text{a}\quad \dfrac{\dfrac{\text{b}\quad \text{c}}{(\text{NP} \multimap \text{S}) \multimap \text{N}} \quad \dfrac{\dfrac{\text{d}\quad \text{e}}{\text{QP}} \quad \dfrac{\text{f}\quad (\text{t}_1)}{\text{NP} \multimap \text{S}}}{\text{S}}}{\text{NP} \multimap \text{S}}\,{}^{1}}{\text{QP}} \qquad \dfrac{\text{g}\quad \dfrac{\text{h}\quad (\text{t}_2)}{\text{NP} \multimap \text{S}}}{\text{NP} \multimap \text{S}}\,{}^{2}}{\text{S}}$$

(32) **Semantics Produced by the Preceding Proof**

EVERY (FARMER THAT $(\lambda_m$.A DONKEY (OWN $m)))$ $(\lambda_m$.IT$_i$ (BEAT $m)) =$
$\lambda_{c\mathbf{x}^{|c|}}$.not (exists$_y$.(farmer y) and exists$_z$.(donkey z) and (own $y\ z$) and (not (beat $y\ w$)

where w is the i-th component of the tuple $\mathbf{x}^{|c|}, y, z$. When $i = |c| + 1$, then $w = z$ and we obtain donkey anaphora with the strong reading:

$\lambda_{c\mathbf{x}^{|c|}}$.not (exists$_y$.(farmer y) and exists$_z$.(donkey z) and (own $y\ z$) and (not (beat $y\ z$)