Introduction to Hyperintensional Dynamic Semantics

Scott Martin and Carl Pollard

Department of Linguistics
Ohio State University

June 27, 2012

Review: Types of (Static) Hyperintensional Semantics

- Basic types from the logic:
  t (truth values)
  n (natural numbers)
- Basic static semantic types:
  e (entities)
  p (propositions)
- some nonbasic static semantic types:
  $\mathrm{p}_{\mathit{1}} =_{\mathrm{def}} \mathrm{e} \rightarrow \mathrm{p}$ (unary static properties)
  $\mathrm{p}_{\mathit{n+1}} =_{\mathrm{def}} \mathrm{e} \rightarrow \mathrm{p}_{\mathit{n}}$ (n-ary static properties, $n > 1$)
  $\mathrm{p}_{\mathit{1}} \rightarrow \mathrm{p}$ (static generalized quantifiers)
  $\mathrm{p}_{\mathit{1}} \rightarrow \mathrm{p}_{\mathit{1}} \rightarrow \mathrm{p}$ (static determiners)

Review: Static Propositional Connectives and Quantifiers

a. $\vdash \textsf{truth} : \mathrm{p}$

b. $\vdash \textsf{falsity} : \mathrm{p}$

c. $\vdash \textsf{not} : \mathrm{p} \rightarrow \mathrm{p}$ (translates *it is not the case that*)

d. $\vdash \textsf{and} : \mathrm{p} \rightarrow \mathrm{p} \rightarrow \mathrm{p}$ (translates *and*)

e. $\vdash \textsf{or} : \mathrm{p} \rightarrow \mathrm{p} \rightarrow \mathrm{p}$ (translates *or*)

f. $\vdash \textsf{implies} : \mathrm{p} \rightarrow \mathrm{p} \rightarrow \mathrm{p}$ (translates episodic *if* ... *then*)

g. $\vdash \textsf{exists}_A : (A \rightarrow \mathrm{p}) \rightarrow \mathrm{p}$

h. $\vdash \textsf{forall}_A : (A \rightarrow \mathrm{p}) \rightarrow \mathrm{p}$

i. $\vdash \textsf{entails} : \mathrm{p} \rightarrow \mathrm{p} \rightarrow \mathrm{t}$

j. $\vdash \equiv : \mathrm{p} \rightarrow \mathrm{p} \rightarrow \mathrm{t}$ (mutual entailment)

Axioms for Static Propositional Connectives

These axioms say that the type p is interpreted as a preboolean algebra relative to the entailment preorder. A is a type metavariable; p, q, and r variables of type p, x is of type A, and P of type $A \rightarrow p$.

a. $\vdash \forall_p . p$ entails truth
b. $\vdash \forall_p .$falsity entails p
c. $\vdash \forall_{p,q} .(p$ and $q)$ entails p
d. $\vdash \forall_{p,q} .(p$ and $q)$ entails q
e. $\vdash \forall_{p,q,r} .((p$ entails $q) \wedge (p$ entails $r)) \rightarrow (p$ entails $(q$ and $r))$
f. $\vdash \forall_{p,q} . p$ entails $(p$ or $q)$
g. $\vdash \forall_{p,q} . q$ entails $(p$ or $q)$
h. $\vdash \forall_{p,q,r} .((p$ entails $r) \wedge (q$ entails $r)) \rightarrow ((p$ or $q)$ entails $r)$
i. $\vdash \forall_{p,q} .(p$ implies $q)$ and $p)$ entails q
j. $\vdash \forall_{p,q,r} .((r$ and $p)$ entails $q) \rightarrow (r$ entails $(p$ implies $q))$
k. $\vdash \forall_p .($not $p) \equiv (p$ implies falsity$)$
l. $\vdash \forall_p .($not $($not $p))$ entails p

Axioms for Static Propositional Quantifiers

m. $\vdash \forall_{xP}.(P\ x)$ entails (exists P)

n. $\vdash \forall_{pP}.(\forall_x.(P\ x)$ entails $p) \rightarrow (($exists $P)$ entails $p)$

o. $\vdash \forall_{xP}.($forall $P)$ entails $(P\ x)$

p. $\vdash \forall_{pP}.(\forall_x.p$ entails $(P\ x)) \rightarrow (p$ entails (forall P))

Review: Tonicity (1/2)

- Recall that if $\langle S, \sqsubseteq \rangle$ and $\langle P, \leq \rangle$ are two preordered sets, then $f : S \to P$ is called **monotonic** (resp. **antitonic**) iff for all $s, s' \in S$, if $s \sqsubseteq s'$ then $f(s) \leq f(s')$ (resp. $f(s') \leq f(s)$).
- f is called **tonic** if it is either monotonic or antitonic, and **atonic** otherwise.
- Linguists often say 'upward monotonic' for monotonic, and 'downward monotonic' for antitonic.

Review: Tonicity (2/2)

- If $f : S \to S \to P$ is a (curried) function, it is called **monotonic** (resp. **antitonic**) **in its first** (resp. **second**) **argument** iff, for each $r \in S$, the function $\lambda_s.f(s)(r)$ (resp. $\lambda_s.f(r)(s)$) is monotonic (resp. antitonic).
- The case we're interested in is where P is the set of (static) propositions (type p), $\leq$ is entailment, S is the set of (static) properties (type $e \to p$), and $\sqsubseteq$ is defined by

$$P \sqsubseteq Q =_{\text{def}} \forall_x.(P\ x) \text{ entails } (Q\ x)$$

So the functions we are concerned with have type

$$(\mathrm{e} \to \mathrm{p}) \to (\mathrm{e} \to \mathrm{p}) \to p$$

namely, the type of (static) determiners.

Static Property Conjunction

- $\textsf{that} =_{\text{def}} \lambda_{PQx}.(P\ x)\ \textsf{and}\ (Q\ x) : (\text{e} \to \text{p}) \to (\text{e} \to \text{p}) \to (\text{e} \to \text{p})$
- Like propositional conjunction, static property conjunction is associative, commutative, and idempotent up to equivalence, i.e. for any three static properties P, Q, R:

$$(P\ \textsf{that}\ Q)\ \textsf{that}\ R \equiv P\ \textsf{that}\ (Q\ \textsf{that}\ R)$$

$$(P\ \textsf{that}\ Q) \equiv Q\ \textsf{that}\ P$$

$$(P\ \textsf{that}\ P) \equiv P$$

Static Conservativity

- A static determiner d is called (**statically**) **conservative** iff, for all P, Q,

$$(d\ P\ Q) \equiv (d\ P\ (P\ \ \mathsf{that}\ Q))$$

- Natural language static determiners are conservative, e.g.

  A donkey brays iff a donkey is a donkey that brays.

  Every donkey brays iff every donkey is a donkey that brays.

  No donkey brays iff no donkey is a donkey that brays, etc.

Tonicity of Static Determiners

A static determiner is called:

$\uparrow\uparrow$ iff it is monotonic in both arguments

$\uparrow\downarrow$ iff it is monotonic in the first argument and antitonic in the second

$\downarrow\uparrow$ iff it is antitonic in the first argument and monotonic in the second

$\downarrow\downarrow$ iff it is antitonic in both arguments

$\not\uparrow\!\not\downarrow\uparrow$ iff it is atonic in the first argument and monotonic in the second.

Examples

$\uparrow\uparrow$ determiners: *a, some, several, many, at least n*

$\uparrow\downarrow$ determiners: *not every, not all*

$\downarrow\uparrow$ determiners: *every, all*

$\downarrow\downarrow$ determiners: *no, few, at most n*

$\not\uparrow\not\downarrow\uparrow$ determiner: *most*

Testing Tonicity of Determiners

The following (dis-)entailments are characteristic of a $\not\uparrow\not\downarrow\uparrow$ determiner:

Most donkeys bray and snort entails *Most donkeys bray*.

Most donkeys bray does not entail *Most brown donkeys bray*.

Most brown donkeys bray does not entail *Most donkeys bray*.

Readings of *d farmer that owns a donkey beats it*

- Weak: d farmer that owns a donkey owns a donkey and beats it *or*
  d farmer that owns a donkey beats a donkey that he owns
- Strong: d farmer that owns a donkey beats every donkey that he owns
- E-type: d farmer that owns a donkey beats the donkey that he owns
- Pair-quantification: For d pairs $\langle x, y \rangle$ where x is a farmer, y is a donkey, and x owns y, x beats y

Note: as we'll see, for some determiners, two or more of these readings might have identical truth conditions.

Which determiners allow which readings?

Why?

Conservativization

- We define

$$\mathbf{conserv} : ((\mathrm{e} \to \mathrm{p}) \to (\mathrm{e} \to \mathrm{p}) \to \mathrm{p}) \to (\mathrm{e} \to \mathrm{p}) \to (\mathrm{e} \to \mathrm{p}) \to \mathrm{p}$$

  as follows:

$$\mathbf{conserv} =_{\mathrm{def}} \lambda_{dPQ}.d\ P\ (P\ \textsf{that}\ Q)$$

- Observations:
  - For all d, **conserv** d is conservative.
  - If d is conservative, then **conserv** $d \equiv d$
  - **conserv** is idempotent u.t.e..
  - The notion of conservativity does *not* carry over straightforwardly to the dynamic setting.

The Uniqueness Condition

- The **uniqueness** condition is that no farmer own more than one donkey.
- If the uniqueness condition is satisfied, then the e-type reading gives the right truth conditions.
- But it's not clear what to say about the e-type reading when the uniqueness condition is *not* satisfied.
- Also, usually there is not much reason to think the uniquess condition *is* satisfied.

The Consistency Condition

- The **consistency** condition is that every donkey-owning farmer treats all his donkeys the same way.
- Obviously the uniqueness condition is a special case of the consistency condition.
- When the consistency condition is satisfied, the weak and strong reading coincide and give the intuitively correct truth conditions.
- In the special case of uniqueness, the e-type reading coincides with these also.

The Pair-Quantification Reading

- This is the reading predicted by classical DRT/FCS.
- It coincides with the strong reading for d = *every*.
- It coincides with the weak reading when d = *no*, *some*, or *a*.
- It doesn't work at all for d = *most*.
- Also, it doesn't work at all for d = *at least two*, so the failure of the pair-quantification reading is not merely an issue of 'proportion'.
- In fact, the pair-quantification reading fails for nearly all determiners.
- Based on these considerations and the ones above about the e-type reading, Kanazawa suggests abandoning both the pair-quantification reading and the e-type reading as playing any role in the interpretation of donkey sentences.

Summary of Kanazawa's Empirical Claims (1/2)

- The interpretation of a donkey sentence is given by either the weak or the strong reading.
- The choice of determiner is the main factor that affects which readings are possible.
- More specifically, the key factor is the tonicity of the determiner.

Summary of Kanazawa's Empirical Claims (2/2)

- $\uparrow\uparrow$ determiners (*a, some, several, many, at least n*): weak reading only.
- $\downarrow\downarrow$ determiners (*no, few, at most n*): weak reading only.
- $\downarrow\uparrow$ determiners (*every, all*): strong reading preferred.
- $\uparrow\downarrow$ determiners (*not every, not all*): strong reading preferred?
- $\not\uparrow\not\downarrow\uparrow$ determiner (*most*): both readings possible:

  Most people that owned a slave also owned his offspring. (strong reading preferred)

  Most men that have a quarter put it in the parking meter. (weak reading preferred).

Where we are Going with This

- We accept Kanazawa's arguments for ignoring the e-type and pair-quantification readings.
- We agree with Kanazawa that for *most*, both weak and strong readings are available.
- But we reject his claim that the strong reading is 'preferred' for *every*.
- It seems that the strong reading for *every* and *most* is favored in cases where there is good reason to assume the consistency condition is satisfied.
- And so, we conclude that the only reading generated by the grammar should be the *weak* reading.
- Apparent strong 'readings' arise via pragmatic inference (e.g. based on consistency assumptions.
- We'll implement a weak-reading-only analysis of donkey sentences within **hyperintensional dynamic semantics (HDS)**.

Types for Contexts in HDS

- $\mathrm{c}_{0} =_{\mathrm{def}} \mathrm{p}$ (nullary contexts)
- $\mathrm{c}_n =_{\mathrm{def}} \mathrm{e}^n \rightarrow \mathrm{p}$ (n-ary contexts, $n > 0$
  Example: the output context from an utterance in the (unrealistic!) null input context truth of *a farmer beats a donkey* is the binary context

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

- $\mathrm{c} =_{\mathrm{def}} \coprod_{n \in \mathrm{n}}.\mathrm{c}_n$ (contexts)
- For $c \in \mathrm{c}_n$, $|c| =_{\mathrm{def}} n$ is the number of active discourse referents (DRs) in c.

About HDS Contexts

- Our contexts correspond roughly to Lewis/Stalnaker/Heim common grounds (CGs).
- The abstraction represents *indeterminacy* about the identity of the entities that the CG is about.
- Using abstraction rather than existential quantification obviates the need for scope extension or continuations to render DRs accessible for subsequent anaphora.
- It is also in the spirit of DRT and FCS that indefinites are fundamentally *nonquantificational*—a property shared with definites—though they have the same *type* as quantificational NPs.

Proffered Contents and Context Changes

- The type for proffered contents of decarative sentences and their associated context changes is:
  $\mathrm{k} =_{\mathrm{def}} \mathrm{c} \rightharpoonup \mathrm{c}$
- For each proffered content $k \in \mathrm{k}$, there is a natural number $|k|$ such that for every c in the domain of k,

  $$|k\ c| = |c| + |k|$$

  Intuitively, $|k|$ is the number of discourse referents (DRs) that k introduces.

Dynamic Conjunction of Proffered Contents

- AND : k → k → k is defined as follows:

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

- the function cc : k → k mapping proffered contents to their associated context changes is defined as follows:

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

The first conjunct is the carryover from the input context, and the second is the contribution from the proffered content itself.

- We can use cc to relate dynamic conjunction of proffered contents to composition of context changes:

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

Other Dynamic Semantic Types

- unary dynamic properties: $\mathrm{d}_{\mathit{1}} =_{\mathrm{def}} \mathrm{n} \to \mathrm{k}$
- n-ary dynamic properties ($n > 1$) $\mathrm{d}_{n+\mathit{1}} =_{\mathrm{def}} \mathrm{n} \to \mathrm{d}_n$
- dynamic generalized quantifiers: $\mathrm{d}_{\mathit{1}} \to \mathrm{k}$
- dynamic determiners: $\mathrm{d}_{\mathit{1}} \to \mathrm{d}_{\mathit{1}} \to \mathrm{k}$

Dynamicization of Properties (1/2)

Dynamic properties can be defined by applying a **dynamicization** function $\mathbf{dyn}_n : \mathrm{p}_n \rightarrow \mathrm{d}_n$ to the static counterpart. For $n < 3$ these are:

$$\mathbf{dyn}_{\mathit{0}}\ p =_{\mathrm{def}} \lambda_c \lambda_{\mathbf{x}^{|c|}}.p$$
$$\mathbf{dyn}_{\mathit{1}}\ P =_{\mathrm{def}} \lambda_m.\lambda_{c||c|>m}.\lambda_{\mathbf{x}^{|c|}}.P\ x_m$$
$$\mathbf{dyn}_{\mathit{2}}\ R =_{\mathrm{def}} \lambda_{mn}.\lambda_{c||c|>m,n}.\lambda_{\mathbf{x}^{|c|}}.R\ x_m\ x_n$$

Dynamicization of Properties (2/2)

Examples:

$$\text{COLD} =_{\text{def}} \mathbf{dyn}_{\mathit{0}}\ \mathsf{cold} = \lambda_c.\lambda_{\mathbf{x}^{|c|}}.\mathsf{cold}$$
$$\text{DONKEY} =_{\text{def}} \mathbf{dyn}_{\mathit{1}}\ \mathsf{donkey} = \lambda_m.\lambda_{c||c|>m}.\lambda_{\mathbf{x}^{|c|}}.\mathsf{donkey}\ x_m$$
$$\text{BEAT} =_{\text{def}} \mathbf{dyn}_{\mathit{2}}\ \mathsf{beat} = \lambda_{mn}.\lambda_{c||c|>m,n}.\lambda_{\mathbf{x}^{|c|}}.\mathsf{beat}\ x_m\ x_n$$

Dynamic Conjunction of Dynamic Properties

THAT $: \mathrm{d}_1 \rightarrow \mathrm{d}_1 \rightarrow \mathrm{d}_1$ is defined as follows:

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

Unlike their static counterparts, dynamic conjunction (of both proffered contents and dynamic properties) are *not* commutative or idempotent u.t.e. (though they *are* associative u.t.e.).

That's because the two conjuncts are evaluated in different contexts.

As a consequence, the notion of conservativity does not transfer straigh tforwardly to the dynamic setting.

But maybe we won't need it?

Review of Dynamic Negation

Dynamic negation of proffered contents NOT : k → k is defined by NOT $k =_{\text{def}}$:

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

And dynamic negation of dynamic properties NON : $\mathrm{d}_1 \to \mathrm{d}_1$ is defined as follows:

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

Dynamic Double Negation

- Dynamically negated proffered contents do not introduce any DRs:

$$\vdash \forall_k . \, |\textsc{not}\ k| = 0$$

- If $|k| = 0$, then

$$\vdash \textsc{not}\ (\textsc{not}\ k) \equiv k$$

- If $|k| = m > 0$, then

$$\vdash \textsc{not}\ (\textsc{not}\ k) \not\equiv k$$

- More specifically:

$$\vdash \textsc{not}\ (\textsc{not}\ k) \equiv \lambda_{c|k\downarrow c} . \lambda_{\mathbf{x}^{|c|}} . \mathsf{exists}_{\mathbf{y}^m} . k\ c\ \mathbf{x}, \mathbf{y}$$

That is, dynamic double negation of a proffered content has the effect of (statically) existentially binding all the DRs that it introduces.

And another Thing ...

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

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

The Dynamic 'Existential' Quantifier

- The dynamic generalized quantifier EXISTS is defined as follows:

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

- Crucially, the new DR $|c|$ depends on c, which is λ-bound but *not* existentially bound, just as in DRT and FCS.
- So there is no need for any kind of scope extension mechanism, or for continuations.

  Example:

$$\text{EXISTS DONKEY} = \lambda_c.\lambda_{\mathbf{x}^{|c|},y}.\mathsf{donkey}\ y$$

Two Dynamic Determiners

- the dynamic indefinite determiner:

$$\text{A } D \ E =_{\text{def}} \text{EXISTS } (D \text{ THAT } E)$$

- the dynamic negative determiner:

$$\text{NO } D \ E =_{\text{def}} \text{NOT } (\text{A } D \ E)$$

- *Examples:*

$$\textit{A donkey brays} \rightsquigarrow$$
$$\text{A DONKEY BRAY} = \lambda_c.\lambda_{\mathbf{x}^{|c|},y}.(\mathsf{donkey}\ y)\ \mathsf{and}\ (\mathsf{bray}\ y)$$

$$\textit{No donkey brays} \rightsquigarrow$$
$$\text{NO DONKEY BRAY} =$$
$$\lambda_c.\lambda_{\mathbf{x}^{|c|}}.\mathsf{not}(\mathsf{exists}_y.(\mathsf{donkey}\ y)\ \mathsf{and}\ (\mathsf{bray}\ y))$$

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.
  b. The antecedent is practically entailed by the context to satisfy the 'descriptive content' of the pronoun (in this case, roughly speaking, being nonhuman).
  c. In using the pronoun, the speaker publicly certifies that the context provides sufficient information for the addressee to resolve *which* DR is the antecedent.

- We handle (a) and (b) in the semantics of the pronoun.
- We dodge the pragmatic issues posed by (c) (the anaphoricity, or retrievability, or contextual felicity) by treating the pronoun as *ambiguous* with respect to which DR is its antecedent.

The Definite Pronoun *It*, Formally

- For each i, *it* $\rightsquigarrow$

$$\lambda_D.\lambda_{c|(|c|>i)\wedge(c \text{ pentails } \lambda_{\mathbf{x}|c|}.\text{nonhuman } x_i)}.D\ i\ c$$

- These meanings are dynamic generalized quantifiers.
- Thus pronouns (like other definite NPs, and like indefinite NPs), have the same semantic type as quanticational NPs, even though there is nothing quantificational about them.
- *Example: It brays* $\rightsquigarrow$ IT_i BRAY =

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

Two Unambiguous Donkey Sentences

- *A farmer that owns a donkey beats it* $\rightsquigarrow$
  $\textsc{a}\ (\textsc{farmer that}(\lambda_m.\textsc{a donkey}\ (\textsc{own}\ m)))(\lambda_m.\textsc{it}_i\ (\textsc{beat}\ m)) =$
  $\lambda_c.\lambda_{\mathbf{x}^{|c|},y,z}.(\mathsf{farmer}\ y)\ \mathsf{and}\ (\mathsf{donkey}\ z)\ \mathsf{and}\ (\mathsf{own}\ y\ z)\ \mathsf{and}\ (\mathsf{beat}\ y\ w)))$
  where w is the i-th component of the tuple $\mathbf{x}, y, z$. When $i = |c| + 1$, then $w = z$ and we get:
  $\lambda_c.\lambda_{\mathbf{x}^{|c|},y,z}.(\mathsf{farmer}\ y)\ \mathsf{and}\ (\mathsf{donkey}\ z)\ \mathsf{and}\ (\mathsf{own}\ y\ z)\ \mathsf{and}\ (\mathsf{beat}\ y\ z)))$
- *No farmer that owns a donkey beats it* $\rightsquigarrow$
  $\lambda_c.\lambda_{\mathbf{x}^{|c|}}.\mathsf{not}(\mathsf{exists}_{y,z}.(\mathsf{farmer}\ y)\ \mathsf{and}\ (\mathsf{donkey}\ z)\ \mathsf{and}\ (\mathsf{own}\ y\ z)\ \mathsf{and}\ (\mathsf{b}$
- These are the desired readings.

Two Donkey Sentences with Weak and Strong Readings

a. *Every farmer that owns a donkey beats it.*
b. *Most farmers that own a donkey beat it.*

How should we analyze these?

A First Attempt (1/2)

- An obvious analysis of *every* is to use a traditional semantics of which yields the strong reading:

$$\text{EVERY}_s =_{\text{def}} \lambda_{DE}.\text{NOT}\ (\text{A}\ D\ (\text{NON}\ E)) =$$

and then apply (the dynamic counterpart of) the conservativization operator to it to get another semantics for *every* that yields the weak reading:

$$\text{EVERY}_w =_{\text{def}} \lambda_{DE}.\text{NOT}\ (\text{A}\ D\ (\text{NON}\ (D\ \text{ THAT }\ E))) =$$
$$\lambda_{DE}.\text{NOT}\ (\text{EXISTS}\ (D\ \text{ THAT}\ (\text{NON}\ (D\ \text{ THAT }\ E)))$$

- But this strategy doesn't generalize to the case of *most*.

A First Attempt (2/2)

- Additionally, there's a technical problem (pointed out by Chierchia) of 'donkey doubling' arising from the fact that the weak reading of (a) is essentially analyzed as *every farmer that owns a donkey is a farmer that owns a donkey and beats it*: we end up with two different donkey DRs, and the pronoun can resolve to either one of them!
- Instead, we posit meanings for *every* and *most* that yield weak readings, and assume that the strong readings arise via pragmatic inference in the presence of background assumptions of consistency.

Weak *Every*

- we start with the previous definition of weak *every* and eliminate the ‘donkey doubling’ problem by doubly negating the restriction to prevent any DRs introduced in the relative clause from getting passed into the scope:

$$\text{EVERY}_w =_{\text{def}}$$

$$\lambda_{DE}.\text{NOT}\ (\text{A}\ (\text{NON}\ (\text{NON}\ D))\ (\text{NON}\ (D\ \ \text{THAT}\ \ E))) =$$

$$\lambda_{DE}.\text{NOT}\ (\text{EXISTS}\ ((\text{NON}\ (\text{NON}\ D))\ \text{THAT}(\text{NON}\ (D\ \text{THAT}\ E)))$$

- This yields the right reading and solves the donkey doubling problem, but it has a kind of hokey, arbitrary look to it.
- How can we justify this theoretically?

Justifying Weak *Every* (1/2)

- We use as our model the natural hyperintensional generalization of Montague's definition for static *every*:

$$\mathsf{every} =_{\text{def}} \lambda_{PQ}.\mathsf{forall}_x.((P\ x)\ \mathsf{implies}\ (Q\ x))$$

- Thus, we define:

$$\text{EVERY} =_{\text{def}} \lambda_{DE}.\text{FORALL}_n.((D\ n)\ \text{IMPLIES}\ (E\ n))$$

- Here the dynamic universal quantifier is defined as expected:

$$\text{FORALL} =_{\text{def}} \lambda_D.\text{NOT}\ (\text{EXISTS}\ (\text{NON}\ D))$$

- And dynamic implication, also as expected, is defined so that presuppositions of the consequent can be satisfied from the antecedent:

$$\text{IMPLIES} =_{\text{def}} \lambda_{kh}.(\text{NOT}\ k)\ \text{OR}\ (k\ \text{AND}\ h)$$

- Here dynamic disjunction is defined as in DMG:

$$\text{OR} =_{\text{def}} \lambda_{kh}.\text{NOT}\ ((\text{NOT}\ k)\ \text{AND}\ (\text{NOT}\ h))$$

Justifying Weak *Every* (2/2)

- From these definitions, we establish:

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

- But because the outer double negation is eliminable (why?), this is equivalent to

$$\lambda_{DE}.\textsc{not}\ (\textsc{exists}_n.((\textsc{not}\ (\textsc{not}\ (D\ n)))\textsc{and}(\textsc{not}\ ((D\ n)\textsc{and}(E\ n))$$
$$\lambda_{DE}.\textsc{not}\ (\textsc{exists}\ ((\textsc{non}\ (\textsc{non}\ D))\ \textsc{that}(\textsc{non}\ (D\ \textsc{that}\ E)))$$

which coincides with the amended semantics of weak *every*.