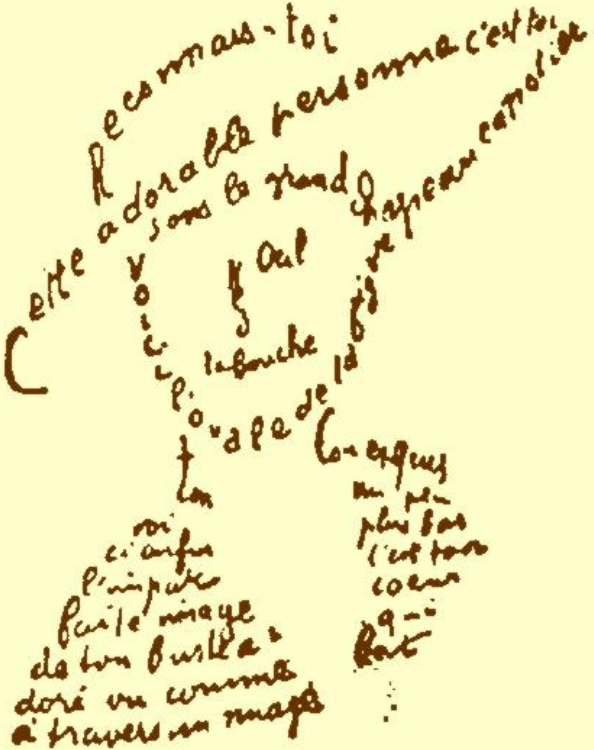

Reconnais-toi
Cette adorable personne c'est toi
sous la grand chapeau carolingien
Voici
Oui
Voici la bouche
Voici le pistolet de la
Ton
Quelques
un pli-
plus bas
c'est ton
coeur
qui
a enfin
l'impact
Suite image
de ton Buste
doré en comme
à travers un nuage

Yet Another Dynamic Logic

Philippe de Groote
LORIA & Inria-Lorraine

A Type-theoretic reconstruction of DRT

A Type-theoretic reconstruction of DRT

Motivation:

- to formalize DRT within Church's simple theory of type (aka, Higher-Order Logic), which will allow DRT and Montague semantics to rest on the same logical foundations.

A Type-theoretic reconstruction of DRT

Motivation:

- to formalize DRT within Church's simple theory of type (aka, Higher-Order Logic), which will allow DRT and Montague semantics to rest on the same logical foundations.

Challenge:

- to express dynamics using "static" primitives (in particular, to avoid the "destructive assignment" problem, wich necessitates a LISP-like $gensym$ operator).

A Type-theoretic reconstruction of DRT

Motivation:

- to formalize DRT within Church's simple theory of type (aka, Higher-Order Logic), which will allow DRT and Montague semantics to rest on the same logical foundations.

Challenge:

- to express dynamics using "static" primitives (in particular, to avoid the "destructive assignment" problem, wich necessitates a LISP-like $gensym$ operator).

Proposed solution:

- to interpret a sentence according to both its left and right contexts;
- to abstract these two kinds of contexts over the meaning of the sentences.

Typing the left and the right contexts

Typing the left and the right contexts

Montague semantics is based on Church's simple type theory, which provides a full hierarchy of functional types built upon two atomic types:

- ι, the type of individuals (a.k.a. entities).
- o, the type of propositions (a.k.a. truth values).

Typing the left and the right contexts

Montague semantics is based on Church's simple type theory, which provides a full hierarchy of functional types built upon two atomic types:

- ι, the type of individuals (a.k.a. entities).
- o, the type of propositions (a.k.a. truth values).

We add a third atomic type, γ, which stands for the type of the left contexts.

Typing the left and the right contexts

Montague semantics is based on Church's simple type theory, which provides a full hierarchy of functional types built upon two atomic types:

- ι, the type of individuals (a.k.a. entities).
- o, the type of propositions (a.k.a. truth values).

We add a third atomic type, γ, which stands for the type of the left contexts.

What about the type of the right contexts?

Typing the left and the right contexts

Montague semantics is based on Church's simple type theory, which provides a full hierarchy of functional types built upon two atomic types:

- ι, the type of individuals (a.k.a. entities).
- o, the type of propositions (a.k.a. truth values).

We add a third atomic type, γ, which stands for the type of the left contexts.

What about the type of the right contexts?

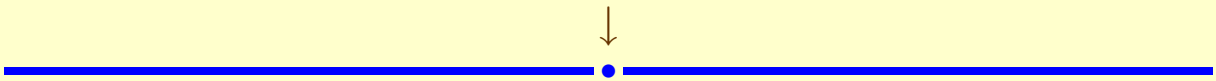
Typing the left and the right contexts

Montague semantics is based on Church's simple type theory, which provides a full hierarchy of functional types built upon two atomic types:

- ι, the type of individuals (a.k.a. entities).
- o, the type of propositions (a.k.a. truth values).

We add a third atomic type, γ, which stands for the type of the left contexts.

What about the type of the right contexts?



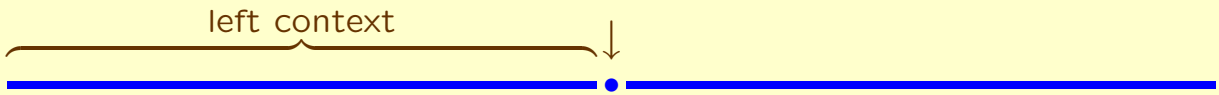
Typing the left and the right contexts

Montague semantics is based on Church's simple type theory, which provides a full hierarchy of functional types built upon two atomic types:

- ι, the type of individuals (a.k.a. entities).
- o, the type of propositions (a.k.a. truth values).

We add a third atomic type, γ, which stands for the type of the left contexts.

What about the type of the right contexts?



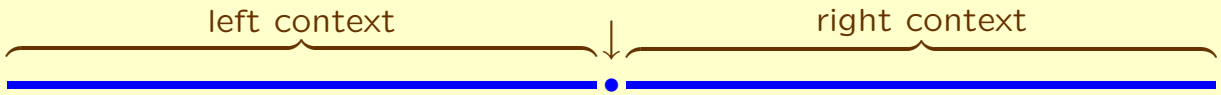
Typing the left and the right contexts

Montague semantics is based on Church's simple type theory, which provides a full hierarchy of functional types built upon two atomic types:

- ι, the type of individuals (a.k.a. entities).
- o, the type of propositions (a.k.a. truth values).

We add a third atomic type, γ, which stands for the type of the left contexts.

What about the type of the right contexts?



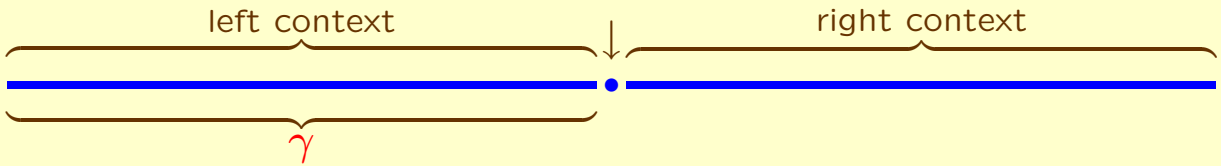
Typing the left and the right contexts

Montague semantics is based on Church's simple type theory, which provides a full hierarchy of functional types built upon two atomic types:

- ι, the type of individuals (a.k.a. entities).
- o, the type of propositions (a.k.a. truth values).

We add a third atomic type, γ, which stands for the type of the left contexts.

What about the type of the right contexts?



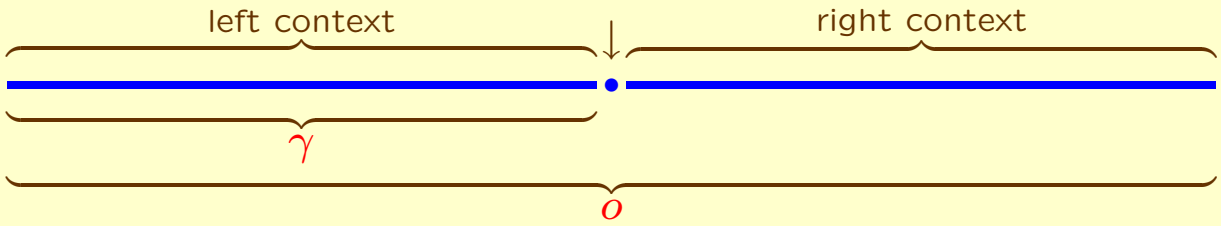
Typing the left and the right contexts

Montague semantics is based on Church's simple type theory, which provides a full hierarchy of functional types built upon two atomic types:

- ι, the type of individuals (a.k.a. entities).
- o, the type of propositions (a.k.a. truth values).

We add a third atomic type, γ, which stands for the type of the left contexts.

What about the type of the right contexts?



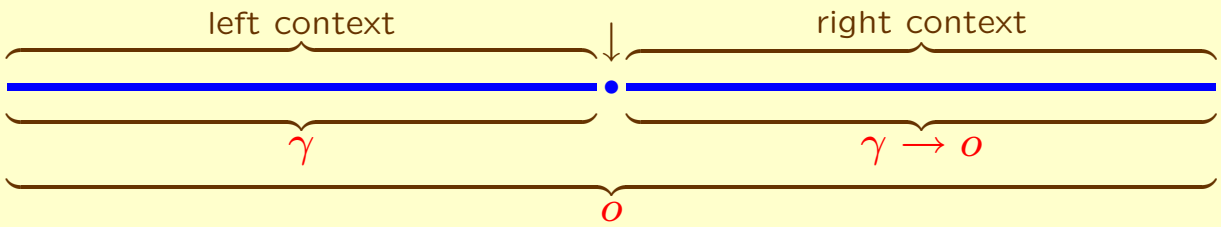
Typing the left and the right contexts

Montague semantics is based on Church's simple type theory, which provides a full hierarchy of functional types built upon two atomic types:

- ι, the type of individuals (a.k.a. entities).
- o, the type of propositions (a.k.a. truth values).

We add a third atomic type, γ, which stands for the type of the left contexts.

What about the type of the right contexts?



Semantic interpretation of the sentences

Semantic interpretation of the sentences

Let s be the syntactic category of sentences. Remember that we intend to abstract our notions of left and right contexts over the meaning of the sentences.

Semantic interpretation of the sentences

Let s be the syntactic category of sentences. Remember that we intend to abstract our notions of left and right contexts over the meaning of the sentences.

$$[\![s]\!] = \gamma \to (\gamma \to o) \to o$$

Semantic interpretation of the sentences

Let s be the syntactic category of sentences. Remember that we intend to abstract our notions of left and right contexts over the meaning of the sentences.

$$[\![s]\!] = \gamma \to (\gamma \to o) \to o$$

Composition of two sentence interpretations

Semantic interpretation of the sentences

Let s be the syntactic category of sentences. Remember that we intend to abstract our notions of left and right contexts over the meaning of the sentences.

$$[\![s]\!] = \gamma \to (\gamma \to o) \to o$$

Composition of two sentence interpretations

$$[\![S_1 . S_2]\!] = \lambda e \phi . [\![S_1]\!]\, e\, (\lambda e' . [\![S_2]\!]\, e'\, \phi)$$

Semantic interpretation of the syntactic categories

Semantic interpretation of the syntactic categories

Montague's interpretation

$$\begin{aligned} [\![s]\!] &= o \\ [\![n]\!] &= \iota \to o \\ [\![np]\!] &= (\iota \to o) \to o \end{aligned}$$

Semantic interpretation of the syntactic categories

Montague's interpretation

$$\begin{array}{rcl} [\![s]\!] & = & o \\ [\![n]\!] & = & \iota \to o \\ [\![np]\!] & = & (\iota \to o) \to o \end{array}$$

may be rephrased as follows:

$$\begin{array}{rcll} [\![s]\!] & = & o & (1) \\ [\![n]\!] & = & \iota \to [\![s]\!] & (2) \\ [\![np]\!] & = & (\iota \to [\![s]\!]) \to [\![s]\!] & (3) \end{array}$$

Semantic interpretation of the syntactic categories

Montague's interpretation

$$\begin{array}{rcl} [\![s]\!] & = & o \\ [\![n]\!] & = & \iota \to o \\ [\![np]\!] & = & (\iota \to o) \to o \end{array}$$

may be rephrased as follows:

$$\begin{array}{rcll} [\![s]\!] & = & o & (1) \\ [\![n]\!] & = & \iota \to [\![s]\!] & (2) \\ [\![np]\!] & = & (\iota \to [\![s]\!]) \to [\![s]\!] & (3) \end{array}$$

Replacing (1) with:

$$[\![s]\!] = \gamma \to (\gamma \to o) \to o$$

Semantic interpretation of the syntactic categories

Montague's interpretation

$$\begin{array}{rcl} [\![s]\!] & = & o \\ [\![n]\!] & = & \iota \to o \\ [\![np]\!] & = & (\iota \to o) \to o \end{array}$$

may be rephrased as follows:

$$\begin{array}{rcll} [\![s]\!] & = & o & (1) \\ [\![n]\!] & = & \iota \to [\![s]\!] & (2) \\ [\![np]\!] & = & (\iota \to [\![s]\!]) \to [\![s]\!] & (3) \end{array}$$

Replacing (1) with:

$$[\![s]\!] = \gamma \to (\gamma \to o) \to o$$

we obtain:

$$\begin{array}{rcl} [\![n]\!] & = & \iota \to \gamma \to (\gamma \to o) \to o \\ [\![np]\!] & = & (\iota \to \gamma \to (\gamma \to o) \to o) \to \gamma \to (\gamma \to o) \to o \end{array}$$

This interpretation results in handcrafted lexical semantics such as the following:

$$[\![\text{every}]\!] = \lambda n \psi e \phi.\, (\forall x.\, \neg(n\, x\, e\, (\lambda e.\, \neg(\psi\, x\, (x{::}e)\, (\lambda e.\, \top))))) \wedge \phi\, e$$

This interpretation results in handcrafted lexical semantics such as the following:

$$[\![\text{every}]\!] = \lambda n \psi e \phi.\, (\forall x.\, \neg(n\, x\, e\, (\lambda e.\, \neg(\psi\, x\, (x{::}e)\, (\lambda e.\, \top))))) \wedge \phi\, e$$

which might seem a little bit involved.

This interpretation results in handcrafted lexical semantics such as the following:

$$[\![\text{every}]\!] = \lambda n \psi e \phi. \, (\forall x. \neg (n \, x \, e \, (\lambda e. \neg (\psi \, x \, (x{::}e) \, (\lambda e. \top))))) \wedge \phi \, e$$

which might seem a little bit involved.

Questions:

- is there a systematic way of obtaining the new lexical semantics from Montague's ?
- can we find any "modular" presentation of the approach ?
- is there some dynamic logic hidden in the approach ?

A Dynamic Logic

A Dynamic Logic

Let $\Omega \triangleq \gamma \to (\gamma \to o) \to o$. We intend to design a logic acting on propositions of type Ω

A Dynamic Logic

Let $\Omega \triangleq \gamma \to (\gamma \to o) \to o$. We intend to design a logic acting on propositions of type Ω

We share with DRT the two following assumptions:

- discourse composition is mainly conjunctive (roughly speaking, a discourse consists in the conjunction of its sentences);
- the main form of quantification is existential (it introduces referential markers).

A Dynamic Logic

Let $\Omega \triangleq \gamma \to (\gamma \to o) \to o$. We intend to design a logic acting on propositions of type Ω

We share with DRT the two following assumptions:

- discourse composition is mainly conjunctive (roughly speaking, a discourse consists in the conjunction of its sentences);
- the main form of quantification is existential (it introduces referential markers).

Consequently, our logic will be based on conjunction and existential quantification (defined as primitives). The other connectives will be obtained using negation (a third primitive) and de Morgan's laws.

Conjunction

Conjunction

Conjunction is nothing but sentence composition. We therefore define:

$$A \sqcap B \triangleq \lambda e \phi.\, A\, e\, (\lambda e.\, B\, e\, \phi)$$

Conjunction

Conjunction is nothing but sentence composition. We therefore define:

$$A \sqcap B \triangleq \lambda e \phi.\, A\, e\, (\lambda e.\, B\, e\, \phi)$$

Existential quantification

Conjunction

Conjunction is nothing but sentence composition. We therefore define:

$$A \sqcap B \triangleq \lambda e \phi.\, A\, e\, (\lambda e.\, B\, e\, \phi)$$

Existential quantification

Existential quantification is canonically defined:

$$\Sigma x.\, P\, x \triangleq \lambda e \phi.\, \exists x.\, P\, x\, e\, \phi$$

Conjunction

Conjunction is nothing but sentence composition. We therefore define:

$$A \sqcap B \triangleq \lambda e \phi.\, A\, e\, (\lambda e.\, B\, e\, \phi)$$

Existential quantification

Existential quantification is canonically defined:

$$\Sigma x.\, P\, x \triangleq \lambda e \phi.\, \exists x.\, P\, x\, e\, \phi$$

Negation

Conjunction

Conjunction is nothing but sentence composition. We therefore define:

$$A \sqcap B \triangleq \lambda e \phi.\, A\, e\, (\lambda e.\, B\, e\, \phi)$$

Existential quantification

Existential quantification is canonically defined:

$$\Sigma x.\, P\, x \triangleq \lambda e \phi.\, \exists x.\, P\, x\, e\, \phi$$

Negation

Negation cannot be canonically defined because we do not want the continuation of the discourse to fall into the scope of the negation:

$$\sim A \triangleq \lambda e \phi.\, \neg\, (A\, e\, (\lambda e.\, \top)) \wedge \phi\, e$$

Implication and Universal Quantification

Implication and Universal Quantification

These are defined using de Morgan's laws:

$$A \sqsupset B \triangleq \sim(A \sqcap \sim B)$$
$$\Pi x.\, P\, x \triangleq \sim\Sigma x.\, \sim(P\, x)$$

Implication and Universal Quantification

These are defined using de Morgan's laws:

$$A \sqsupset B \triangleq \sim(A \sqcap \sim B)$$
$$\Pi x.\, P\, x \triangleq \sim\Sigma x.\, \sim(P\, x)$$

Embedding of first-order logic into dynamic logic

Implication and Universal Quantification

These are defined using de Morgan's laws:

$$\begin{aligned} A \sqsupset B &\triangleq \sim(A \sqcap \sim B) \\ \Pi x.\, P\, x &\triangleq \sim \Sigma x.\, \sim (P\, x) \end{aligned}$$

Embedding of first-order logic into dynamic logic

$$\begin{aligned} \overline{R_i^n\, t_1\, \ldots\, t_n} &= \lambda e \phi.\, R_i^n\, t_1\, \ldots\, t_n \wedge \phi\, e \\ \overline{\neg A} &= \sim \overline{A} \\ \overline{A \wedge B} &= \overline{A} \sqcap \overline{B} \\ \overline{\exists x.\, A} &= \Sigma x.\, \overline{A} \end{aligned}$$

Implication and Universal Quantification

These are defined using de Morgan's laws:

$$\begin{aligned} A \sqsupset B &\triangleq \sim(A \sqcap \sim B) \\ \Pi x.\, P\, x &\triangleq \sim \Sigma x.\, \sim (P\, x) \end{aligned}$$

Embedding of first-order logic into dynamic logic

$$\begin{aligned} \overline{R_i^n\, t_1\, \ldots\, t_n} &= \lambda e \phi.\, R_i^n\, t_1\, \ldots\, t_n \wedge \phi\, e \\ \overline{\neg A} &= \sim \overline{A} \\ \overline{A \wedge B} &= \overline{A} \sqcap \overline{B} \\ \overline{\exists x.\, A} &= \Sigma x.\, \overline{A} \end{aligned}$$

This embedding is such that, for every term e of type γ:

$$A \equiv \overline{A}\, e\, (\lambda e.\, \top)$$

What about dynamics?

What about dynamics?

No dynamics up to this point! In order to get a dynamic interpretation, we must put contexts (i.e., terms of type γ) at work.

What about dynamics?

No dynamics up to this point! In order to get a dynamic interpretation, we must put contexts (i.e., terms of type γ) at work.

We need operations to update and access the contexts. As a first approximation, consider the contexts to be sets of individuals and add the following primitives to the system:

$$\_ :: \_ : \iota \to \gamma \to \gamma$$
$$\mathsf{sel} : \gamma \to \iota$$

where expressions such as $t :: e$ may be interpreted as $\{t\} \cup e$, and where sel is a choice operator.

What about dynamics?

No dynamics up to this point! In order to get a dynamic interpretation, we must put contexts (i.e., terms of type γ) at work.

We need operations to update and access the contexts. As a first approximation, consider the contexts to be sets of individuals and add the following primitives to the system:

$$\_ :: \_ : \iota \to \gamma \to \gamma$$
$$\mathsf{sel} : \gamma \to \iota$$

where expressions such as $t :: e$ may be interpreted as $\{t\} \cup e$, and where sel is a choice operator.

We then define two translations of the (first-order) relational symbols:

$$\overline{R_i^n} \triangleq \lambda x_1 \ldots x_n e \phi.\, R_i^n\, x_1\, \ldots\, x_n \wedge \phi\, e$$
$$\overline{\overline{R_i^n}} \triangleq \lambda x_1 \ldots x_n e \phi.\, R_i^n\, x_1\, \ldots\, x_n \wedge \phi\, (x_1 :: \cdots :: x_n :: e)$$

Donkey sentence revisited

Donkey sentence revisited

Montague-like semantic interpretation:

$$
\begin{aligned}
[\![\text{farmer}]\!] &= \mathbf{farmer} \\
[\![\text{donkey}]\!] &= \mathbf{donkey} \\
[\![\text{owns}]\!] &= \lambda OS.\, S\,(\lambda x.\, O\,(\lambda y.\, \mathbf{own}\, x\, y)) \\
[\![\text{beats}]\!] &= \lambda OS.\, S\,(\lambda x.\, O\,(\lambda y.\, \mathbf{beat}\, x\, y)) \\
[\![\text{who}]\!] &= \lambda RQx.\, Q\, x \wedge R\,(\lambda P.\, P\, x) \\
[\![\text{a}]\!] &= \lambda PQ.\, \exists x.\, P\, x \wedge Q\, x \\
[\![\text{every}]\!] &= \lambda PQ.\, \forall x.\, P\, x \supset Q\, x \\
[\![\text{it}]\!] &= ???
\end{aligned}
$$

Donkey sentence revisited

Montague-like semantic interpretation:

$$
\begin{aligned}
[\![\text{farmer}]\!] &= \mathbf{farmer} \\
[\![\text{donkey}]\!] &= \mathbf{donkey} \\
[\![\text{owns}]\!] &= \lambda OS.\, S\,(\lambda x.\, O\,(\lambda y.\, \mathbf{own}\, x\, y)) \\
[\![\text{beats}]\!] &= \lambda OS.\, S\,(\lambda x.\, O\,(\lambda y.\, \mathbf{beat}\, x\, y)) \\
[\![\text{who}]\!] &= \lambda RQx.\, Q\, x \wedge R\,(\lambda P.\, P\, x) \\
[\![\text{a}]\!] &= \lambda PQ.\, \exists x.\, P\, x \wedge Q\, x \\
[\![\text{every}]\!] &= \lambda PQ.\, \forall x.\, P\, x \supset Q\, x \\
[\![\text{it}]\!] &= ???
\end{aligned}
$$

Dynamic interpretation:

$$
\begin{aligned}
[\![\text{farmer}]\!] &= \overline{\mathbf{farmer}} \\
[\![\text{donkey}]\!] &= \overline{\mathbf{donkey}} \\
[\![\text{owns}]\!] &= \lambda OS.\, S\,(\lambda x.\, O\,(\lambda y.\, \overline{\overline{\mathbf{own}}}\, x\, y)) \\
[\![\text{beats}]\!] &= \lambda OS.\, S\,(\lambda x.\, O\,(\lambda y.\, \overline{\overline{\mathbf{beat}}}\, x\, y)) \\
[\![\text{who}]\!] &= \lambda RQx.\, Q\, x \sqcap R\,(\lambda P.\, P\, x) \\
[\![\text{a}]\!] &= \lambda PQ.\, \Sigma x.\, P\, x \sqcap Q\, x \\
[\![\text{every}]\!] &= \lambda PQ.\, \Pi x.\, P\, x \sqsupset Q\, x \\
[\![\text{it}]\!] &= \lambda Pe\phi.\, P\,(\mathtt{sel}\, e)\, e\, \phi
\end{aligned}
$$

With the dynamic interpretation we have that:

$$[\![\text{beats}]\!]\,[\![\text{it}]\!]\,([\![\text{every}]\!]\,([\![\text{who}]\!]\,([\![\text{owns}]\!]\,([\![\text{a}]\!]\,[\![\text{donkey}]\!]))\,[\![\text{farmer}]\!]))$$

With the dynamic interpretation we have that:

$$[\![\text{beats}]\!]\,[\![\text{it}]\!]\,([\![\text{every}]\!]\,([\![\text{who}]\!]\,([\![\text{owns}]\!]\,([\![\text{a}]\!]\,[\![\text{donkey}]\!]))\,[\![\text{farmer}]\!]))$$

β-reduces to the following term (modulo de Morgan's laws):

$$\lambda e\phi.\,(\forall x.\,\mathbf{farmer}\,x\;\supset\;(\forall y.\,\mathbf{donkey}\,y\;\supset\;(\mathbf{own}\,x\,y\;\supset\;\mathbf{beat}\,x\,(\mathsf{sel}\,(x{::}y{::}e)))))\;\wedge\;\phi\,e$$