The Semantics of Intensionalization

Gilad Ben-Avi
Technion

Yoad Winter
Technion/Utrecht University

New Directions in Type-Theoretical Grammar, ESSLLI
Dublin, 10 August 2007

Sensitivity to intensions

- Some words are intension-sensitive (IS):
  seek a lawyer, *fake* diamond, *believe* that...
- Other words are intension-insensitive (INS):
  kiss a lawyer, *shiny* diamond
- IS words and expressions lead to intensional phenomena: propositional attitudes, privative modification.
- INS words and expressions only require extensional semantics.

Syntactic compatibility of IS and INS expressions

IS and INS expressions may share syntactic categories and appear in the same constructions.

Especially – in the case of transitive verbs:
John needed and inherited a house.
Mary sought, found and ate a fish.
Sue ordered and got a new PC.

“Modularity” of intensional phenomena

Intensional phenomena may appear due to mechanisms that are also relevant for purely extensional effects.

The Quine-Montague Hypothesis: *De dicto/de re* ambiguity as scope ambiguity:
A queen kissed every king.
A queen looked for a king.

“Modularity” of intensional phenomena

Intensional phenomena may appear due to mechanisms that are also relevant for purely extensional effects.

The Quine-Montague Hypothesis: *De dicto*/*de re* ambiguity as scope ambiguity:
A queen kissed every king.
A queen looked for a king. (vs. *A queen kissed a king*)

A voice from the 80s

Keenan and Faltz 1985, p. 274:

“Our general task will be to create a system of model-theoretic semantic interpretation for our logical language which will preserve the advantages and insights revealed by our extensional system while allowing properly intensional facts to be represented.”

Our aim – a modular architecture of intensional semantics

- Start with an extensional grammar – only INS items in the lexicon.

Our aim – a modular architecture of intensional semantics

- Start with an extensional grammar – only INS items in the lexicon.
- Intensionalize – shift meanings of INS items so to allow:

Our aim – a modular architecture of intensional semantics

- Start with an extensional grammar – only INS items in the lexicon.
- Intensionalize – shift meanings of INS items so to allow:
  - adding IS items,

Our aim – a modular architecture of intensional semantics

- Start with an extensional grammar – only INS items in the lexicon.
- Intensionalize – shift meanings of INS items so to allow:
  - adding IS items,
  - without changing anything else in the extensional grammar,

Our aim – a modular architecture of intensional semantics

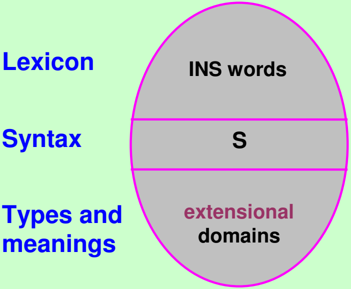
- Start with an extensional grammar – only INS items in the lexicon.
- Intensionalize – shift meanings of INS items so to allow:
    - adding IS items,
    - without changing anything else in the extensional grammar,
    - and especially – preserving truth-conditional behavior.

Our aim – a modular architecture of intensional semantics

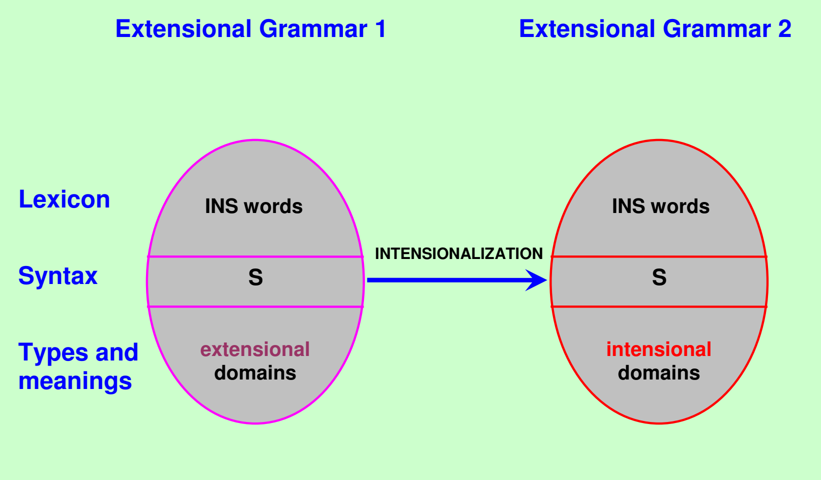
- Start with an extensional grammar – only INS items in the lexicon.
- Intensionalize – shift meanings of INS items so to allow:
  - adding IS items,
  - without changing anything else in the extensional grammar,
  - and especially – preserving truth-conditional behavior.
- Add IS items to the lexicon.

Graphically

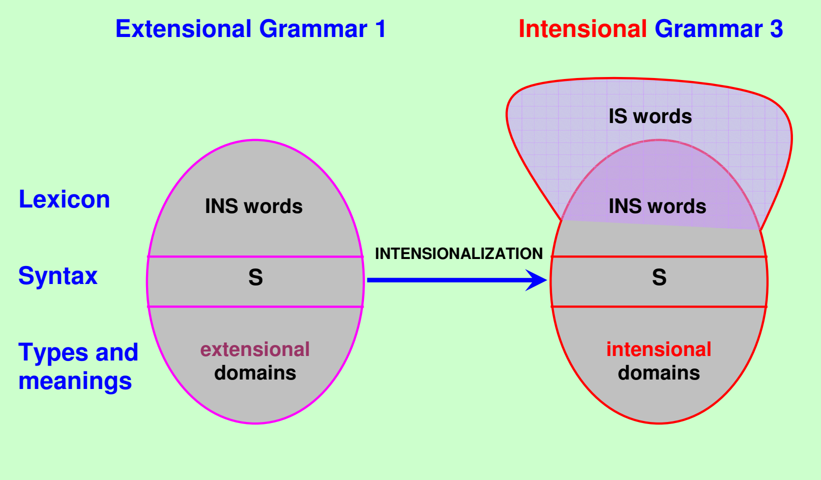
Extensional Grammar 1



Graphically



Graphically



Benefits

Architectural benefits:

- Extensional treatments have exact parallels in intensional systems.
- Treatments of intensional phenomena are fully lexicalized.
- No *ad hoc* type shifts for intensionality as in Partee & Rooth (1983).

Benefits for treating concrete phenomena:

- *De dicto*/*de re* – manifestations of extensional scope shifting principles.
- Avoiding PTQ-style meaning postulates for high types of INS words.
- IS TVs (e.g. *seek*) and INS TVs (e.g. *kiss*) are naturally coordinated.

Pedagogical benefit:

- We don't have to teach intensional systems "from scratch" – we can rely to a large extent on the understanding of extensional systems.

What is intensionalization?

An intensionalization procedure $\mathcal{I}$ is a mapping of an extensional grammar G to a grammar $\mathcal{I}(G)$ that satisfies:

What is intensionalization?

An intensionalization procedure $\mathcal{I}$ is a mapping of an extensional grammar G to a grammar $\mathcal{I}(G)$ that satisfies:

- Strong syntactic equivalence between G and $\mathcal{I}(G)$.

What is intensionalization?

An intensionalization procedure $\mathcal{I}$ is a mapping of an extensional grammar G to a grammar $\mathcal{I}(G)$ that satisfies:

- Strong syntactic equivalence between G and $\mathcal{I}(G)$.
- Truth-conditional soundness: G and $\mathcal{I}(G)$ support the same entailments.

What is intensionalization?

An intensionalization procedure $\mathcal{I}$ is a mapping of an extensional grammar G to a grammar $\mathcal{I}(G)$ that satisfies:

- Strong syntactic equivalence between G and $\mathcal{I}(G)$.
- Truth-conditional soundness: G and $\mathcal{I}(G)$ support the same entailments.
- Extendability: by only adding IS lexical items, $\mathcal{I}(G)$ can be extended to an adequate intensional grammar.

Precursors on intensionalization

- An explicit intensionalization procedure in Keenan and Faltz (1985).
- Insights on type-theoretical frameworks in Van Benthem (1988).
- A limited intensionalization procedure can be inferred from Heim and Kratzer (1998,ch.12).
- Shan (2001) formalizes a similar intensionalization procedure using *monads*, following Barker's (2002) modular treatment of quantification using *continuations*.

Possible-world intensionalization

Modify components of a grammar G as follows:

- Types – add a basic type s.
- Frame – add a nonempty domain D_s of possible worlds.
- Typing – modify types of lexical items.
- Meanings – modify meanings of lexical items.

Sentences in $\mathcal{I}(G)$ are of type st.

Soundness of possible-world intensionalization

The following should be equivalent, given two derivations $\mathcal{D}(S_1)$ and $\mathcal{D}(S_2)$ in G of sentences S_1 and S_2:

- For every model $\mathcal{M}$ of G:

$$[\![\mathcal{D}(S_1)]\!]^{\mathcal{M}} = 1 \Rightarrow [\![\mathcal{D}(S_2)]\!]^{\mathcal{M}} = 1$$

- For every model $\mathcal{M}$ of $\mathcal{I}(G)$ and for every $w \in D_s$:

$$[\![\mathcal{D}(S_1)]\!]^{\mathcal{M}}(w) = 1 \Rightarrow [\![\mathcal{D}(S_2)]\!]^{\mathcal{M}}(w) = 1$$

Principles

- Following Van Benthem (1988):
  - For all extensional types: type t (of truth-values) is replaced by type st (of propositions).
  - Only relational types are used (inessential).
- Heim and Kratzer's distinction between logical constants and non-logical constants is preserved.
- A sophisticated mapping is now needed for logical constants, which may have many s's in their intensionalized type.
- Words for boolean operators are treated syncategorematically.

Relational types

A proposal by Van Benthem (1988), attributed to Reinhard Muskens:

> *"a relational rather than a functional type theory may prove the proper setting for this investigation…"*

We restrict the set of possible types to $\mathcal{T}_{\text{ext}}$, which is the least set s.t.

- $t \in \mathcal{T}_{\text{ext}}$.
- If $\sigma_1 \in \{e\} \cup \mathcal{T}_{\text{ext}}$ and $\sigma_2 \in \mathcal{T}_{\text{ext}}$ then $(\sigma_1\sigma_2) \in \mathcal{T}_{\text{ext}}$.

Note that if $\sigma \in \mathcal{T}_{\text{ext}}$ then:

- $\sigma = (\sigma_1 \ldots (\sigma_n t) \ldots)$ for some $n \geq 0$ and $\sigma_1, \ldots \sigma_n \in \{e\} \cup \mathcal{T}_{\text{ext}}$.
- D_σ is isomorphic to $\wp(D_{\sigma_1} \times \cdots \times D_{\sigma_n})$.

Intensionalizing types

- For every $\sigma \in \mathcal{T}_{\text{ext}}$, let $\ulcorner\sigma\urcorner$ the type that results from replacing each occurrence of t within σ by st.
  (This is adopted from Van Benthem (1988).)
- Let $\mathcal{T}_{\text{int}} \overset{def}{=} \{\ulcorner\sigma\urcorner | \sigma \in \mathcal{T}_{\text{ext}}\}$.
- Denote by $\llcorner\cdot\lrcorner$ the inverse of $\ulcorner\cdot\urcorner$.

Examples:

- $\ulcorner t\urcorner = st$ (propositions).
- $\ulcorner et\urcorner = e(st)$ (properties).
  Note: $D_{e(st)}$ is isomporphic to $D_{s(et)}$ – the standard domain of properties.
- $\ulcorner e(et)\urcorner = e(e(st))$

- **Logical constants**: If a word α has a constant denotation $f \in D_\sigma$ (e.g., *every*), then in the intensionalized grammar α denotes a constant $L(f) \in D_{\ulcorner\sigma\urcorner}$.
- **Non-logical constants**: If a word α can denote any $f \in D_\sigma$, then in the intensionalized grammar α can denote any $f \in D_{\ulcorner\sigma\urcorner}$. For instance: words that denote arbitrary one-place predicates in D_{et} are mapped to words that denote arbitrary elements in $D_{e(st)}$.

But how to define $L(\cdot)$?

Intensionalizing *every*

- **every** $= \lambda A_{et} \lambda B_{et} . \forall x_e [A(x) \rightarrow B(x)]$.

Intensionalizing *every*

- **every** $= \lambda A_{et} \lambda B_{et}.\forall x_e[A(x) \rightarrow B(x)]$.
- We would like to arrive at the PTQ-like denotation:
  $L(\textbf{every}) = \lambda A_{e(st)} \lambda B_{e(st)} \lambda w_s.\forall x_e[A(x)(w) \rightarrow B(x)(w)]$

Intensionalizing *every*

- **every** $= \lambda A_{et} \lambda B_{et}.\forall x_e[A(x) \rightarrow B(x)]$.
- We would like to arrive at the PTQ-like denotation:
$$L(\textbf{every}) = \lambda A_{e(st)} \lambda B_{e(st)} \lambda w_s.\forall x_e[A(x)(w) \rightarrow B(x)(w)]$$
- Note that if we define the extension of $A \in D_{e(st)}$ in $w \in D_s$ as $A^w \stackrel{def}{=} \lambda x_e.A(w)(x)$ then:
$$L(\textbf{every})(A)(B)(w) = \textbf{every}(A^w)(B^w)$$

Intensionalizing *every*

- **every** $= \lambda A_{et} \lambda B_{et}.\forall x_e[A(x) \rightarrow B(x)]$.
- We would like to arrive at the PTQ-like denotation:
  $$L(\textbf{every}) = \lambda A_{e(st)} \lambda B_{e(st)} \lambda w_s.\forall x_e[A(x)(w) \rightarrow B(x)(w)]$$
- Note that if we define the extension of $A \in D_{e(st)}$ in $w \in D_s$ as $A^w \stackrel{def}{=} \lambda x_e.A(w)(x)$ then:
  $$L(\textbf{every})(A)(B)(w) = \textbf{every}(A^w)(B^w)$$
- But, standardly: the extension of an intensional denotation φ in $w \in D_s$ is assumed to be $\varphi(w)$.

Intensionalizing *every*

- **every** $= \lambda A_{et} \lambda B_{et}.\forall x_e[A(x) \rightarrow B(x)]$.
- We would like to arrive at the PTQ-like denotation:

$$L(\textbf{every}) = \lambda A_{e(st)} \lambda B_{e(st)} \lambda w_s.\forall x_e[A(x)(w) \rightarrow B(x)(w)]$$

- Note that if we define the extension of $A \in D_{e(st)}$ in $w \in D_s$ as $A^w \stackrel{def}{=} \lambda x_e.A(w)(x)$ then:

$$L(\textbf{every})(A)(B)(w) = \textbf{every}(A^w)(B^w)$$

- But, standardly: the extension of an intensional denotation φ in $w \in D_s$ is assumed to be $\varphi(w)$.
- Thus, standardly: it is implicitly assumed that φ has to have a type $s\tau$ in order to have an extension.

Intensionalizing *every*

- **every** $= \lambda A_{et} \lambda B_{et}. \forall x_e [A(x) \rightarrow B(x)]$.
- We would like to arrive at the PTQ-like denotation:
$$L(\textbf{every}) = \lambda A_{e(st)} \lambda B_{e(st)} \lambda w_s. \forall x_e [A(x)(w) \rightarrow B(x)(w)]$$
- Note that if we define the extension of $A \in D_{e(st)}$ in $w \in D_s$ as $A^w \stackrel{def}{=} \lambda x_e. A(w)(x)$ then:
$$L(\textbf{every})(A)(B)(w) = \textbf{every}(A^w)(B^w)$$
- But, standardly: the extension of an intensional denotation φ in $w \in D_s$ is assumed to be $\varphi(w)$.
- Thus, standardly: it is implicitly assumed that φ has to have a type $s\tau$ in order to have an extension.
And furthermore – τ has to be extensional.

Intensionalizing *every*

- **every** $= \lambda A_{et} \lambda B_{et}. \forall x_e [A(x) \to B(x)]$.
- We would like to arrive at the PTQ-like denotation:
$$L(\mathbf{every}) = \lambda A_{e(st)} \lambda B_{e(st)} \lambda w_s. \forall x_e [A(x)(w) \to B(x)(w)]$$
- Note that if we define the extension of $A \in D_{e(st)}$ in $w \in D_s$ as $A^w \stackrel{def}{=} \lambda x_e. A(w)(x)$ then:
$$L(\mathbf{every})(A)(B)(w) = \mathbf{every}(A^w)(B^w)$$
- But, standardly: the extension of an intensional denotation φ in $w \in D_s$ is assumed to be $\varphi(w)$.
- Thus, standardly: it is implicitly assumed that φ has to have a type $s\tau$ in order to have an extension.
And furthermore – τ has to be extensional.
- This is a too strong assumption for obtaining a general extensionalization procedure.

We therefore generalize our observation about intensional determiners in PTQ.

Generalized extension

Definition (extension F^w of F in a world w)

Let $\sigma \in \mathcal{T}_{\text{int}} \cup \{e\}$, $F \in D_\sigma$ and $w \in D_s$.

1. if $\sigma = e$ then $F^w = F$;
2. if $\sigma = (\sigma_1 \cdots (\sigma_n(st)) \cdots)$, $n \geq 0$, then for all $x_1 \in D_{\llcorner \sigma_1 \lrcorner}, \ldots, x_n \in D_{\llcorner \sigma_n \lrcorner}$:

$$F^w(x_1) \cdots (x_n) = 1 \Leftrightarrow$$
$$\exists Y_1 \cdots Y_n [\bigwedge_{i=1}^{n} (Y_i^w = x_i) \wedge f(Y_1) \cdots (Y_n)(w) = 1]$$

In words: A tuple $x_1, \ldots, x_n$ is in the w-extension of a relation F iff there is a tuple $Y_1, \ldots, Y_n$ in F whose w-extensions are $x_1, \ldots, x_n$.

Intensionalizing logical constants

The L operator: Let $\sigma \in \mathcal{T}_{\text{ext}} \cup \{e\}$ and $f \in D_\sigma$.

1. if $\sigma = e$ then $L(f) = f$;
2. if $\sigma = (\sigma_1 \cdots (\sigma_n t) \cdots)$, $n \geq 0$, then for every $w \in D_s$ and for all $X_1 \in D_{\ulcorner \sigma_1 \urcorner}, \ldots, X_n \in D_{\ulcorner \sigma_n \urcorner}$:
   $(L(f))(X_1) \cdots (X_n)(w) = f(X_1^w) \cdots (X_n^w)$.

In words: A tuple $X_1, \ldots, X_n$ and w are in the intension of a relation f iff the w-extensions of $X_1, \ldots, X_n$ are in f.

If a word α has a constant denotation $f \in D_\sigma$, then in the intensionalized grammar α denotes the constant $L(f) \in D_{\ulcorner \sigma \urcorner}$.

Note: Applying L to extensional det's gives intensional PTQ-style det's.

Soundness

Theorem

The intensionalization procedure described above is sound for any extensional grammar with:

- *Logical constants – of constant denotation.*
- *Non-logical constants (only n-ary predicates over e-type entities) – of arbitrary denotation.*
- *Syncategorematic boolean operators.*

Improvement (De Groote, Kanazawa and Muskens – p.c.)

$$F^{w}(x_1)\cdots(x_n) \;=\; F(L(x_1))\cdots(L(x_n))(w)$$

Expected benefits:

- No restriction to relational types.
- No restriction over the types of non-logical constants.
- Simpler soundness proof (De Groote, Kanazawa and Muskens).
- Similar – or wider – linguistic coverage (work in progress).

A simple extensional lexicon

word α	**type**	**denotes in**	λ-term
Mary, *John*,…	e	D_e	
king, *queen*,…	et	D_{et}	
smile,…	et	D_{et}	
kiss,…	$e(et)$	$D_{e(et)}$	
every	$(et)((et)t)$	$\{\textbf{every}\}$	$\lambda A_{et}\lambda B_{et}.\forall x_e[A(x) \rightarrow B(x)]$
a	$(et)((et)t)$	$\{\textbf{some}\}$	$\lambda A_{et}\lambda B_{et}.\exists x_e[A(x) \wedge B(x)]$

Zero morphology items and syncategorematic operations

word α	**type**	**denotes in**	λ-term
ϵ_{ONS}	$(e(et))(((et)t)(et))$	$\{$**ons**$\}$	$\lambda R_{e(et)} \lambda F_{(et)t} \lambda x_e.F(\lambda y_e.R(y)(x))$
ϵ_{OWS}	$(((et)t)(et))$ $(((et)t)(((et)t)t))$	$\{$**ows**$\}$	$\lambda R_{(((et)t)(et))} \lambda F_{(et)t} \lambda Q_{(et)t}.$ $F(\lambda y_e.Q(\lambda x_e.R(\lambda A_{et}.A(y))(x)))$
ϵ_{lift}	$e((et)t)$	$\{$**lift**$\}$	$\lambda x_e \lambda A_{et}.A(x)$

In addition: boolean conjunction, disjunction and negation are treated syncategorematically.

Examples of derivations

Example

The sentence *Every king kissed a queen* has (at least) the two derivations in (1) and (2).

(1) [[Every king] [[ϵ_{ONS} kissed] [a queen]]]

(2) [[Every king] [[ϵ_{OWS} [ϵ_{ONS} kissed]] [a queen]]]

- $[\![(1)]\!]^{\mathcal{M}} = 1$ iff

  $$\forall x \in D_e[[\![king]\!]^{\mathcal{M}}(x) \rightarrow \exists y \in D_e[[\![queen]\!]^{\mathcal{M}}(y) \wedge [\![kiss]\!]^{\mathcal{M}}(y)(x)]]$$

- $[\![(2)]\!]^{\mathcal{M}} = 1$ iff

  $$\exists y \in D_e[[\![queen]\!]^{\mathcal{M}}(y) \wedge \forall x \in D_e[[\![king]\!]^{\mathcal{M}}(x) \rightarrow [\![kiss]\!]^{\mathcal{M}}(y)(x)]]$$

- Standardly, (2) (extensionally) entails (1).

After intensionalization

word α	**type**	**denotes in**	λ-term
Mary, John,...	e	D_e	
king, queen,...	$e(st)$	$D_{e(st)}$	
smile,...	$e(st)$	$D_{e(st)}$	
kiss,...	$e(e(st))$	$D_{e(e(st))}$	
every	$(e(st))((e(st))(st))$	$\{L(\textbf{every})\}$	(3)
a	$(e(st))((e(st))(st))$	$\{L(\textbf{some})\}$	(4)

(3) $L(\textbf{every}) = \lambda \mathcal{A}_{e(st)} \lambda \mathcal{B}_{e(st)} \lambda w_s . \forall x_e [\mathcal{A}(x)(w) \rightarrow \mathcal{B}(x)(w)]$

(4) $L(\textbf{some}) = \lambda \mathcal{A}_{e(st)} \lambda \mathcal{B}_{e(st)} \lambda w_s . \exists x_e [\mathcal{A}(x)(w) \wedge \mathcal{B}(x)(w)]$

After intensionalization (cont.)

word α	type	denotes in	λ-term
ϵ_{ONS}	$(e(e(st)))(((e(st))(st))(e(st)))$	$\{L(\mathbf{ons})\}$	(5)
ϵ_{OWS}	$(((e(st))(st))(e(st)))$ $(((e(st))(st))(((e(st))(st))(st)))$	$\{L(\mathbf{ows})\}$	(6)
ϵ_{lift}	$e((e(st))(st))$	$\{L(\mathbf{lift})\}$	(7)

(5) $L(\mathbf{ons}) = \lambda\mathcal{R}_{e(e(st))}\lambda\mathcal{F}_{(e(st))(st)}\lambda x_e \lambda w_s.\mathcal{F}^w(\lambda y_e.\mathcal{R}(y)(x)(w))$

(6) $L(\mathbf{ows}) = \lambda\mathcal{R}_{(((e(st))(st))(e(st)))}\lambda\mathcal{F}_{(e(st))(st)}\lambda\mathcal{Q}_{(e(st))(st)}\lambda w_s.$
$\mathcal{F}^w(\lambda y_e.\mathcal{Q}^w(\lambda x_e.\mathcal{R}^w(\lambda A_{et}.A(y))(x)))$

(7) $L(\mathbf{lift}) = \lambda x_e \lambda\mathcal{A}_{e(st)}.\mathcal{A}(x)$

Intensionalized derivations

Example

The sentence *Every king kissed a queen* has (at least) the two derivations:

(8) [[Every king] [[ϵ_{ONS} kissed] [a queen]]]

(9) [[Every king] [[ϵ_{OWS} [ϵ_{ONS} kissed]] [a queen]]]

Given an intensional model $\mathcal{M}$ and $w \in D_s$:

- $[\![(8)]\!]^{\mathcal{M}}(w) = 1$ iff

  $$\forall x_e[[\![king]\!]^{\mathcal{M}}(x)(w) \rightarrow \exists y_e[[\![queen]\!]^{\mathcal{M}}(y)(w) \wedge [\![kiss]\!]^{\mathcal{M}}(y)(x)(w)]]$$

- $[\![(9)]\!]^{\mathcal{M}}(w) = 1$ iff

  $$\exists y_e[[\![queen]\!]^{\mathcal{M}}(y)(w) \wedge \forall x_e[[\![king]\!]^{\mathcal{M}}(x)(w) \rightarrow [\![kiss]\!]^{\mathcal{M}}(y)(x)(w)]]$$

- Now, the extensional entailment is preserved after intensionalization: (9) (intensionally) entails (8).

Extending the intensionalized grammar

- We can add a transitive verb like *seek* as a nonlogical constant of type $((e(st))(st))(e(st))$.
- Thus, the object of *seek* is an intensional quantifier like in PTQ.

Example

In a model $\mathcal{M}$, the derivation (10) is interpreted as the proposition in (11).

(10) [Mary [sought [a king]]]

(11) $[\![seek]\!]^{\mathcal{M}}(\lambda \mathcal{B}_{e(st)} \lambda w_s.\exists y_e[[\![king]\!]^{\mathcal{M}}(y)(w) \wedge \mathcal{B}(y)(w)])([\![Mary]\!]^{\mathcal{M}})$

Deriving the *de re* interpretation extensionally

- The interpretation in (11) is the *de dicto* interpretation of (10).
- We can also derive the *de re* interpretation, using the *intensionalized* version of the *extensional* scope mechanism.

Example

In a model $\mathcal{M}$, the derivation (12) is interpreted as the proposition in (13).

(12) [[ϵ_{lift} Mary] [[ϵ_{OWS} sought] [a king]]]

(13) $\lambda w_s \exists y_e [[\![king]\!]^{\mathcal{M}}(y)(w) \wedge ([\![seek]\!]^{\mathcal{M}})^w(\lambda A_{et}.A(y))([\![Mary]\!]^{\mathcal{M}})]$

Coordination of mixed transitive verbs

Example

(14) [Mary [[sought [and [ϵ_{ONS} kissed]]] [a king]]]

The denotation of (14) in a model $\mathcal{M}$ is:

$$\lambda w_s.\exists y_e[\mathbf{king}(y)(w) \wedge \mathbf{kiss}(y)(\mathbf{m})(w)] \wedge \\ \mathbf{seek}(\lambda \mathcal{B}_{e(st)} \lambda w_s.\exists y_e[\mathbf{king}(y)(w) \wedge \mathcal{B}(y)(w)])(\mathbf{m})$$

= "Mary sought a king and kissed a king"

- A *de dicto* reading of *a king* relative to *seek*.
- The existential import of *Mary kissed a king* is preserved thanks to the intensionalization technique.

Conclusion

- Intensionalization glues an *extensional grammar* to *intensional lexical entries*.
- Implication 1: Extensional scope mechanisms allow to derive intensional *de dicto*/*de re* ambiguities – the Quine-Montague Hypothesis.
- Implication 2: Extensional composition of objects with transitive verbs allows to derive conjunctions of extensional TVs with intensional TVs.
- *Hope*: Such modular architectures will be found useful for different grammatical frameworks and various linguistic phenomena, and especially for teaching them.