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.
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.*
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.
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*)
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.
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

Lexicon

Syntax

Types and meanings

INS words

S

extensional domains
Graphically

**Lexicon**

**Syntax**
- **Types and meanings**
  - Extensional domains

**Extensional Grammar 1**
- INS words
- S

**INTENSIONALIZATION**

**Extensional Grammar 2**
- INS words
- S
  - Intensional domains
Graphically

Lexicon

Syntax

Types and meanings

INS words

S

extensional domains

INTENSIONALIZATION

IS words

INS words

S

intensional domains
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.
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.
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.


A limited intensionalization procedure can be inferred from Heim and Kratzer (1998, ch. 12).

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 $\mathcal{L}(G)$ are of type $st$. 
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 $G$:
  \[
  \llbracket \mathcal{D}(S_1) \rrbracket^\mathcal{M} = 1 \Rightarrow \llbracket \mathcal{D}(S_2) \rrbracket^\mathcal{M} = 1
  \]

- For every model $\mathcal{M}$ of $\mathcal{I}(G)$ and for every $w \in D_s$:
  \[
  \llbracket \mathcal{D}(S_1) \rrbracket^\mathcal{M}(w) = 1 \Rightarrow \llbracket \mathcal{D}(S_2) \rrbracket^\mathcal{M}(w) = 1
  \]
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.
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 $\sigma_1, \ldots, \sigma_n \in \{e\} \cup \mathcal{T}_{\text{ext}}$.
- $D_\sigma$ is isomorphic to $\varphi(D_{\sigma_1} \times \cdots \times D_{\sigma_n})$. 
For every $\sigma \in T_{\text{ext}}$, let $\lceil \sigma \rceil$ the type that results from replacing each occurrence of $t$ within $\sigma$ by $st$.
(This is adopted from Van Benthem (1988).)

Let $T_{\text{int}} \overset{\text{def}}{=} \{ \lceil \sigma \rceil | \sigma \in T_{\text{ext}} \}$.

Denote by $\lceil \cdot \rceil$ the inverse of $\lceil \cdot \rceil$.

Examples:

- $\lceil t \rceil = st$ (propositions).
- $\lceil et \rceil = e(st)$ (properties).
  Note: $D_{e(st)}$ is isomorphic to $D_{s(et)}$ – the standard domain of properties.
- $\lceil e(et) \rceil = e(e(st))$
**Logical constants**: If a word $\alpha$ has a constant denotation $f \in D_\sigma$ (e.g., *every*), then in the intensionalized grammar $\alpha$ denotes a constant $L(f) \in D_{\Gamma\sigma^{-1}}$.

**Non-logical constants**: If a word $\alpha$ can denote any $f \in D_\sigma$, then in the intensionalized grammar $\alpha$ can denote any $f \in D_{\Gamma\sigma^{-1}}$. 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*

\[
\text{every} = \lambda A_{et} \lambda B_{et}. \forall x_e [A(x) \rightarrow B(x)].
\]
Intensionalizing \textit{every}

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

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

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

- Note that if we define the extension of \( A \in D_{e(st)} \) in \( w \in D_s \) as
  \( A^w \overset{\text{def}}{=} \lambda x_e. A(w)(x) \) then:
  \[
  L(\text{every})(A)(B)(w) = \text{every}(A^w)(B^w)
  \]
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:
  \[
  L(\text{every}) = \lambda A_{e(st)} \lambda B_{e(st)} \lambda w_s. \forall x_e [A(x)(w) \rightarrow B(x)(w)]
  \]

- Note that if we define the **extension** of $A \in D_{e(st)}$ in $w \in D_s$ as $A^w \overset{\text{def}}{=} \lambda x_e. A(w)(x)$ then:
  \[
  L(\text{every})(A)(B)(w) = \text{every}(A^w)(B^w)
  \]

- But, standardly: the extension of an intensional denotation $\varphi$ in $w \in D_s$ is assumed to be $\varphi(w)$.
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:

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

Note that if we define the extension of \( A \in D_{e(st)} \) in \( w \in D_s \) as

\[ A^w \overset{\text{def}}{=} \lambda x_e A(w)(x) \]

then:

\[ L(\text{every})(A)(B)(w) = \text{every}(A^w)(B^w) \]

But, standardly: the extension of an intensional denotation \( \varphi \) in \( w \in D_s \) is assumed to be \( \varphi(w) \).

Thus, standardly: it is implicitly assumed that \( \varphi \) 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:
  $$L(\text{every}) = \lambda A_{e(st)} \lambda B_{e(st)} \lambda w_s. \forall x_e [A(x)(w) \rightarrow B(x)(w)]$$

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

- But, standardly: the extension of an intensional denotation $\varphi$ in $w \in D_s$ is assumed to be $\varphi(w)$.

- Thus, standardly: it is implicitly assumed that $\varphi$ has to have a type $s\tau$ in order to have an extension.
  And furthermore – $\tau$ has to be extensional.
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:
  \[L(\text{every}) = \lambda A_{e(st)} \lambda B_{e(st)} \lambda w_s . \forall x_e [A(x)(w) \rightarrow B(x)(w)]\]

- Note that if we define the extension of \(A \in D_{e(st)}\) in \(w \in D_s\) as 
  \(A^w \overset{\text{def}}{=} \lambda x_e . A(w)(x)\) then:
  \[L(\text{every})(A)(B)(w) = \text{every}(A^w)(B^w)\]

- But, standardly: the extension of an intensional denotation \(\varphi\) in \(w \in D_s\) is assumed to be \(\varphi(w)\).

- Thus, standardly: it is implicitly assumed that \(\varphi\) has to have a type \(s\, \tau\) in order to have an extension.
  And furthermore – \(\tau\) 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.
Definition (extension $F^w$ of $F$ in a world $w$)

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

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

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

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 $X_1 \in D_{\neg \sigma_1 \neg}, \ldots, X_n \in D_{\neg \sigma_n \neg}$:
   $$(L(f))(X_1) \cdots (X_n)(w) = f(X_1^w) \cdots (X_n^w).$$

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 $\alpha$ has a constant denotation $f \in D_\sigma$, then in the intensionalized grammar $\alpha$ denotes the constant $L(f) \in D_{\neg \sigma \neg}$.

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**.
\[ F^w(x_1) \cdots (x_n) = F(L(x_1)) \cdots (L(x_n))(w) \]

**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

<table>
<thead>
<tr>
<th>word</th>
<th>type</th>
<th>denotes in</th>
<th>(\lambda)-term</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary, John, ...</td>
<td>(e)</td>
<td>(D_e)</td>
<td>(\lambda A_{et} \lambda B_{et}. \forall x [A(x) \rightarrow B(x)])</td>
</tr>
<tr>
<td>king, queen, ...</td>
<td>(et)</td>
<td>(D_{et})</td>
<td>(\lambda A_{et} \lambda B_{et}. \forall x [A(x) \rightarrow B(x)])</td>
</tr>
<tr>
<td>smile, ...</td>
<td>(et)</td>
<td>(D_{et})</td>
<td>(\lambda A_{et} \lambda B_{et}. \forall x [A(x) \rightarrow B(x)])</td>
</tr>
<tr>
<td>kiss, ...</td>
<td>(e(et))</td>
<td>(D_{e(et)})</td>
<td>(\lambda A_{et} \lambda B_{et}. \forall x [A(x) \rightarrow B(x)])</td>
</tr>
<tr>
<td>every</td>
<td>((et)((et)t))</td>
<td>{every}</td>
<td>(\lambda A_{et} \lambda B_{et}. \forall x [A(x) \rightarrow B(x)])</td>
</tr>
<tr>
<td>a</td>
<td>((et)((et)t))</td>
<td>{some}</td>
<td>(\lambda A_{et} \lambda B_{et}. \exists x [A(x) \land B(x)])</td>
</tr>
<tr>
<td>word $\alpha$</td>
<td>type</td>
<td>denotes in</td>
<td>$\lambda$-term</td>
</tr>
<tr>
<td>----------------</td>
<td>------</td>
<td>------------</td>
<td>----------------</td>
</tr>
<tr>
<td>$\epsilon_{\text{ONS}}$</td>
<td>$(e(et))(((et)t)(et))$</td>
<td>${\text{ons}}$</td>
<td>$\lambda R_{e(et)}\lambda F_{(et)t}\lambda x_e.F(\lambda y_e.R(y)(x))$</td>
</tr>
<tr>
<td>$\epsilon_{\text{OWS}}$</td>
<td>$(((et)t)(et))$</td>
<td>${\text{ows}}$</td>
<td>$\lambda R_{(((et)t)(et))}\lambda F_{(et)t}\lambda Q_{(et)t}$. $F(\lambda y_e.Q(\lambda x_e.R(\lambda A_{et}.A(y))(x)))(x))$</td>
</tr>
<tr>
<td>$\epsilon_{\text{lift}}$</td>
<td>$e((et)t)$</td>
<td>${\text{lift}}$</td>
<td>$\lambda x_e\lambda A_{et}.A(x)$</td>
</tr>
</tbody>
</table>

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) \[ [[\text{Every king}] [[\epsilon_{\text{ONS}} \text{ kissed}] [\text{a queen}]]] \]

(2) \[ [[\text{Every king}] [[\epsilon_{\text{OWS}} [\epsilon_{\text{ONS}} \text{ kissed}]] [\text{a queen}]]] \]

- \[ [(1)]^\mathcal{M} = 1 \text{ iff} \]
  \[ \forall x \in D_e[[\text{king}]^\mathcal{M}(x) \rightarrow \exists y \in D_e[[\text{queen}]^\mathcal{M}(y) \land [\text{kiss}]^\mathcal{M}(y)(x)]] \]

- \[ [(2)]^\mathcal{M} = 1 \text{ iff} \]
  \[ \exists y \in D_e[[\text{queen}]^\mathcal{M}(y) \land \forall x \in D_e[[\text{king}]^\mathcal{M}(x) \rightarrow [\text{kiss}]^\mathcal{M}(y)(x)]] \]

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

<table>
<thead>
<tr>
<th>word</th>
<th>type</th>
<th>denotes in</th>
<th>λ-term</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary, John, . . .</td>
<td>e</td>
<td>De</td>
<td></td>
</tr>
<tr>
<td>king, queen, . . .</td>
<td>e(st)</td>
<td>De(st)</td>
<td></td>
</tr>
<tr>
<td>smile, . . .</td>
<td>e(st)</td>
<td>De(st)</td>
<td></td>
</tr>
<tr>
<td>kiss, . . .</td>
<td>e(e(st))</td>
<td>De(e(st))</td>
<td></td>
</tr>
<tr>
<td>every</td>
<td>(e(st))(e(st))(st))</td>
<td>{L(every)}</td>
<td>(3)</td>
</tr>
<tr>
<td>a</td>
<td>(e(st))(e(st))(st))</td>
<td>{L(some)}</td>
<td>(4)</td>
</tr>
</tbody>
</table>

(3) \(L(\text{every}) = \lambda A_{e(st)} \lambda B_{e(st)} \lambda w.s. \forall x e[ A(x)(w) \rightarrow B(x)(w)]\)

(4) \(L(\text{some}) = \lambda A_{e(st)} \lambda B_{e(st)} \lambda w.s. \exists x e[ A(x)(w) \wedge B(x)(w)]\)
After intensionalization (cont.)

<table>
<thead>
<tr>
<th>word</th>
<th>type</th>
<th>denotes in</th>
<th>λ-term</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon_{\text{ONS}}$</td>
<td>$(e(e(st))(((e(st))(st))(e(st))))$</td>
<td>${L(\text{ons})}$</td>
<td>(5)</td>
</tr>
<tr>
<td>$\epsilon_{\text{OWS}}$</td>
<td>$(((e(st))(st))(e(st)))$</td>
<td>${L(\text{ows})}$</td>
<td>(6)</td>
</tr>
<tr>
<td>$\epsilon_{\text{lift}}$</td>
<td>$e((e(st))(st))$</td>
<td>${L(\text{lift})}$</td>
<td>(7)</td>
</tr>
</tbody>
</table>

(5) $L(\text{ons}) = \lambda R_{e(e(st))} \lambda F_{(e(st))(st)} \lambda x e \lambda w_s. F^w(\lambda y e. R(y)(x)(w))$

(6) $L(\text{ows}) = \lambda R_{((e(st))(st))(e(st))} \lambda F_{(e(st))(st)} \lambda Q_{(e(st))(st)} \lambda w_s. F^w(\lambda y e. Q^w(\lambda x e. R^w(\lambda A e_t. A(y))(x))))$

(7) $L(\text{lift}) = \lambda x e \lambda A_{e(st)}. A(x)$
Example

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

(8) \[ [[[\text{Every king}]] \, [[[\epsilon_{\text{ONS}} \, \text{kissed}]]] \, [[\text{a queen}]]] \]

(9) \[ [[[\text{Every king}]] \, [[[\epsilon_{\text{OWS}} \, [\epsilon_{\text{ONS}} \, \text{kissed}]]]]] \, [[\text{a queen}]]] \]

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

- \[ \llbracket (8) \rrbracket^\mathcal{M}(w) = 1 \text{ iff } \forall x \llbracket \text{king} \rrbracket^\mathcal{M}(x)(w) \rightarrow \exists y \llbracket \text{queen} \rrbracket^\mathcal{M}(y)(w) \land \llbracket \text{kiss} \rrbracket^\mathcal{M}(y)(x)(w) \rrbracket \]

- \[ \llbracket (9) \rrbracket^\mathcal{M}(w) = 1 \text{ iff } \exists y \llbracket \text{queen} \rrbracket^\mathcal{M}(y)(w) \land \forall x \llbracket \text{king} \rrbracket^\mathcal{M}(x)(w) \rightarrow \llbracket \text{kiss} \rrbracket^\mathcal{M}(y)(x)(w) \rrbracket \]

Now, the extensional entailment is preserved after intensionalization: (9) (intensionally) entails (8).
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) \[\text{\texttt{seek}}\,^\mathcal{M}(\lambda B_{e(st)} \lambda w_s \cdot \exists y_e [[\text{king}]^\mathcal{M}(y)(w) \land B(y)(w)])([[\text{Mary}]^\mathcal{M}])]\]
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 $M$, the derivation (12) is interpreted as the proposition in (13).

(12) $[[\epsilon_{\text{lift}} \text{ Mary}]] [[\epsilon_{\text{OWS sought}} [a \text{ king}] ]]$

(13) $\lambda w_s \exists y e [[[\text{king}]^M(y)(w) \land ([\text{seek}]^M)^w(\lambda A_{et}.A(y))([[\text{Mary}]^M])]]$
Example

(14) [Mary [[sought [and [$\epsilon_{ONS}$ kissed]]] [a king]]]

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

$$\lambda w_s. \exists y e [\text{king}(y)(w) \land \text{kiss}(y)(m)(w)] \land$$

$$\text{seek}(\lambda B_{e(st)} \lambda w_s. \exists y e [\text{king}(y)(w) \land B(y)(w)])(m)$$

= “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.
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.