Noname manuscript No.
(will be inserted by the editor)

A Note on Intensionalization

Philippe de Groote · Makoto Kanazawa

the date of receipt and acceptance should be inserted later

Abstract Building on Ben-Avi and Winter's (2007) work, this paper provides a general "intensionalization" procedure that turns an extensional semantics for a language into an intensionalized one that is capable of accommodating "truly intensional" lexical items without changing the compositional semantic rules. We prove some formal properties of this procedure and clarify its relation to the procedure implicit in Montague's (1973) PTQ.

Keywords intensionalization · Montague semantics

1 Introduction

Ben-Avi and Winter (2007) proposed a procedure for *intensionalization*, a method for mapping an object of an "extensional" type (i.e., type based on atomic types *e* and *t*) into an object of a corresponding "intensional" type (based on *e*, *t*, and *s*). They used this mapping to convert an extensional semantics for a fragment of natural language, where all lexical items have extensional denotations, into an intensional semantics which assigns "intensionalized" denotations to the same lexical items and which is capable of accommodating new lexical items with "truly intensional" denotations without any change in the grammar. This is supposed to allow a "modular"

This note has its origin in the discussions we had with Reinhard Muskens in Tokyo in early 2007, after all three of us had heard Yoad Winter present his joint work with Gilad Ben-Avi on intensionalization (see Ben-Avi and Winter 2007). A previous version of this note, dated December 2009, was cited in Jan van Eijck and Christina Unger's textbook (van Eijck and Unger 2011). The section on Montague's typing (Section 5) is new to the present version.

P. de Groote
INRIA Nancy - Grand-Est, 615, rue du Jardin Botanique, 54600 Villers-lès-Nancy, France
E-mail: Philippe.deGroote@loria.fr

M. Kanazawa
National Institute of Informatics, 2–1–2 Hitotsubashi, Chiyoda-ku, Tokyo, 101–8430, Japan
E-mail: kanazawa@nii.ac.jp

development of natural language semantics, where the purely extensional fragment is first presented in the simplest possible terms, without the machinery of possible worlds. Intensionalization can also be discerned, in a particularly simple form, in the semantics of various intensional logics, which are usually built on top of the standard, extensional language of propositional or first-order logic. The higher-order character of natural language semantics makes it a less trivial task to find a suitable definition of intensionalization.

This paper presents an alternative intensionalization procedure that is more general than Ben-Avi and Winter's (2007) in two respects. First, Ben-Avi and Winter's procedure is only applicable to objects of types that are either *quasi-relational* or *e-based*—in other words, types that contain no subtype of the form $(\alpha_1 \to \cdots \to \alpha_n \to t) \to \beta_1 \to \cdots \to \beta_m \to e$. This limitation stems from the fact that their type change scheme replaces t by $s \to t$, but leaves e unchanged. In contrast, the present scheme uniformly replaces each atomic type a in an extensional type α by $s \to a$ to produce an intensional type $\overline{\alpha}$; this uniformity allows the intensionalization procedure to be defined at arbitrary extensional types. Second, unlike Ben-Avi and Winter's (2007) method, where the input is a single extensional object, the method defined here takes a set of extensional objects of type α *parameterized* by objects of type s, and returns an object of type $\overline{\alpha}$. This allows a construction of an intensional model from a class of extensional models viewed as possible worlds; as a result of this, there is no need to stipulate a sharp distinction between lexical items that are "logical constants" and others whose denotations are unconstrained, as in Ben-Avi and Winter's construction. We give the formal definition of intensionalization in Section 2.

Naturally, the present intensionalization procedure generalizes the way an intensional language is usually built on top of an extensional one in formal logic. We take the extensional language whose sentences are simply typed λ-terms (of type t) containing constants, and interpret it in an intensional model constructed from a class of extensional models, assigning each constant the denotation obtained from its denotations in the extensional models by intensionalization. For the intended application to natural language semantics, constants are to stand for extensional lexical items, and a closed λ-term of type t is to represent a possible sentence meaning, expressing how the denotations of words may be combined to give a truth value. After intensionalization, a constant of type α will now denote an object of type $\overline{\alpha}$; a closed λ-term of type t will now express a recipe for combining intensionalized denotations of constants to yield an object of type $\overline{t} = s \to t$, or a set of possible worlds.

To give a concrete example, in giving an extensional semantics for a certain fragment of English, we may use the λ-terms

$$\textsc{every man}\ (\textsc{find}\ (\textsc{a unicorn})) \tag{1}$$

$$\textsc{a unicorn}\ (\boldsymbol{\lambda v}^{e}.(\textsc{every man}\ (\textsc{find}\ (\boldsymbol{\lambda u}^{e\to t}.\boldsymbol{u}\ \boldsymbol{v})))) \tag{2}$$

to represent the subject wide scope and object wide scope readings of the sentence *every man finds a unicorn*. Here, each word whose (extensional) denotation is of type α is represented by the corresponding constant (in small capitals) of type α. We assume that determiners have type $(e \to t) \to (e \to t) \to t$, commoun nouns have type $e \to t$, and transitive verbs have type $((e \to t) \to t) \to e \to t$. (Note that a transitive verb can directly combine with its object by functional application in the case of subject wide scope reading.) Given an extensional model M assigning each constant c of type α its denotation $M(c)$, the truth value of the sentence under the two readings is given by the denotation of the above λ-terms in M. Now, suppose that we systematically intensionalize the semantic types for all syntactic categories, so that determiners, common nouns, and transitive verbs will now denote objects of type $\overline{(e \to t) \to (e \to t) \to t}$, $\overline{e \to t}$, and $\overline{((e \to t) \to t) \to e \to t}$, respectively. Given an intensional denotation for each constant, the same λ-terms (1) and (2) will still be meaningful, provided that the variables $\boldsymbol{v}^e$ and $\boldsymbol{u}^{e \to t}$ are now understood to range over objects of type $\overline{e}$ and $\overline{e \to t}$, respectively; the denotations of (1) and (2) will then be objects of type $\overline{t} = s \to t$. The resulting compositional semantics will be able to accommodate intensional transitive verbs like *seeks*, treating them as belonging to the same syntactic category—and hence having the same semantic type—as extensional transitive verbs like *finds*. Whereas the denotation of *finds* will be an object that is constructed from a collection of objects of type $((e \to t) \to t) \to e \to t$, the denotation of *seeks* will not reduce in the same way to objects of type $((e \to t) \to t) \to e \to t$. The compositional semantics, however, will be agnostic to the distinction, and meaning recipes of the exact same form as (1) and (2) will account for two readings of *every man seeks a unicorn*.

An important desideratum for intensionalization is that the resulting intensional semantics is "conservative" over the original extensional semantics. This desideratum is somewhat misstated in Ben-Avi and Winter's (2007) paper for a reason related to their treatment of "non-logical" lexical items. They demand that validity and consequence be preserved in moving from extensional models to intensional models. Since their treatment of non-logical lexical items in effect amounts to treating them as free variables, this is an unreasonably strong requirement, one that easily fails in the "intensionalization" of as simple a language as the language of propositional logic augmented with a non-logical constant of type $t \to t$. In order to satisfy this desideratum, Ben-Avi and Winter (2007) had to severely restrict the admissible types of non-logical lexical items—specifically, they were limited to those types that contain at most one instance of t, and only at the tail position.

In the present construction of intensional models from classes of extensional models, the desideratum is, simply put, that the truth conditions of sentences (i.e., closed λ-terms of type t) in the extensional language be preserved. More precisely, this means that the truth value of a sentence φ in an extensional model M is to coincide with the truth value of φ in any intensional model constructed form extensional models in which M is the "actual world". The preservation of validity and consequence for sentences follows as a corollary. These results are proved in Section 3.

Even though the validity of an open formula in a class of extensional models should not be expected to be preserved in intensional models built from them, it may be of some technical interest to see to what limited extent this expectation may be satisfied. For an open formula φ in the extensional language to be valid in a class of intensional models, all instances of φ, including formulas in an expanded language containing some truly intensional items, must be valid as well. Particularly simple examples of such formulas are provided by propositional modal logic, where a propositional variable in a tautology may be replaced by any modal formula to produce a valid modal formula. This is a rather trivial case, but there are other more interesting cases. In Section 4, we define a class of *safe* open formulas, and show that for those formulas, validity in extensional models guarantees validity in intensional models based on them.

The present intensionalization procedure based on the mapping $\alpha \mapsto \overline{\alpha}$ is simple and natural, but not so familiar. In Section 5, we compare it to the procedure implicit in Montague (1973), which is closer to the practice of linguists. The two procedures can easily be seen to be equivalent. The chief difference is that in Montague's approach, not only the denotations of the lexical items but also the "meaning recipes" associated with the analysis trees (or "LFs") of natural language sentences must be modified. (Section 5 does not depend on Section 4 and can be read immediately after Section 3.)

It is not entirely clear to us whether a fully general intensionalization procedure such as the one given in this paper is called for in natural language semantics. It seems to us that the usual practice in linguistics is to introduce as few instances of s as is required for proper linguistic analysis, rather than adding as many instances of s as there are instances of e and t.[1] Assigning fully intensionalized types to extensional lexical items may be seen as a manifestation of Montague's strategy of "generalizing to the worst case". Instead, it may be more congenial to the current practice of linguists to adopt a type-shifting mechanism to intensionalize (and extensionalize) denotations of phrases and apply it only when it is needed, and only to the extent that it is needed.[2] Nevertheless, the very fact that a completely general intensionalization procedure exists, with a definition so natural as to look almost inevitable, and its equivalence to a mechanism behind Montague's PTQ, may interest some researchers concerned with foundational aspects of semantics.

2 Intensionalization

We begin with some necessary definitions. The set of *types* over a set A of *atomic types* is defined inductively as follows:[3]

- Every atomic type $a \in A$ is a type over A.
- If α_1 and α_2 are types over A, then $\alpha_1 \to \alpha_2$ is a type over A.

The type constructor $\to$ is assumed to be right-associative, so that $\alpha \to \beta \to \gamma$ stands for $\alpha \to (\beta \to \gamma)$. We let $\alpha^n \to \beta$ abbreviate $\alpha \to \cdots \to \alpha \to \beta$, with "$\alpha\to$" repeated n times.

Given a family $(D_a)_{a \in A}$ of base domains, the domain D_α of objects of type α is defined for each type α over A by recursion:

$$D_{\alpha_1 \to \alpha_2} = D_{\alpha_2}^{D_{\alpha_1}}.$$

In concrete examples, we often assume $A = \{e, t\}$ and $D_t = \{1, 0\}$. The elements of D_e are *individuals* and 1 and 0 are *truth values*.

[1] Cf. von Fintel and Heim (2011, section 1.3.1).

[2] The paper by Partee and Rooth (1983) contains two "type-lifting" principles to this effect.

[3] In Montague semantics, $\alpha \to \beta$ is usually written $\langle\alpha,\beta\rangle$, but we prefer the notation standard in λ-calculus.

Let s be a new atomic type not in A. Elements of D_s are called *indices* or *possible worlds*. The *intensionalization* of type α, written $\overline{\alpha}$, is defined by

$$\overline{\alpha} = \alpha[a := (s \to a)]_{a \in A}.$$

For example, if $\alpha = (e \to t) \to (e \to t) \to t$, then $\overline{\alpha} = ((s \to e) \to s \to t) \to ((s \to e) \to s \to t) \to s \to t$.

We use simply typed λ-calculus (à la Church) as our metalanguage to denote objects in D_α, for types α over $A \cup \{s\}$. The type of a variable is indicated by a superscript at its first occurrence. If ψ is a λ-term of type $\alpha \to \beta$ and χ is a λ-term of type α, then the application $\psi\chi$ is a λ-term of type β and denotes the object in D_β that is the value of ψ on argument χ. If ψ is a λ-term of type β, then the λ-abstract $\lambda x^\alpha.\psi$ is a λ-term of type $\alpha \to \beta$ and denotes the function from D_α to D_β which maps x to ψ. The application of λ-terms is assumed to be left-associative, so that $\varphi\psi\chi$ stands for $(\varphi\psi)\chi$. Application binds stronger than λ-abstraction, so that $\lambda x^\alpha.\psi\chi$ means $\lambda x^\alpha.(\psi\chi)$, rather than $(\lambda x^\alpha.\psi)\chi$. A sequence of λ's is collapsed into one, so that $\lambda x_1^{\alpha_1} \dots x_n^{\alpha_n}.\psi$ abbreviates $\lambda x_1^{\alpha_1}.\dots.\lambda x_n^{\alpha_n}.\psi$.

For each type α over A, we define the *intensionalization* and *extensionalization* combinators at type α by mutual recursion as follows:

$$\begin{aligned}
\mathrm{int}_a &= \mathrm{ext}_a = \lambda x^{s \to a}.x,\\
\mathrm{int}_{\alpha \to \beta} &= \lambda x^{s \to \alpha \to \beta} y^{\overline{\alpha}}.\,\mathrm{int}_\beta(\lambda i^s.xi(\mathrm{ext}_\alpha\, y\, i)),\\
\mathrm{ext}_{\alpha \to \beta} &= \lambda y^{\overline{\alpha \to \beta}} j^s x^\alpha.\,\mathrm{ext}_\beta(y(\mathrm{int}_\alpha(\lambda k^s.x)))j.
\end{aligned}$$

Note that int_α is of type $(s \to \alpha) \to \overline{\alpha}$ and ext_α is of type $\overline{\alpha} \to s \to \alpha$.

Remark 1 If $\alpha = \alpha_1 \to \cdots \to \alpha_n \to a$, we have

$$\begin{aligned}
\mathrm{int}_\alpha &= \lambda x^{s \to \alpha} y_1^{\overline{\alpha_1}} \dots y_n^{\overline{\alpha_n}} i^s.xi(\mathrm{ext}_{\alpha_1}\, y_1\, i) \dots (\mathrm{ext}_{\alpha_n}\, y_n\, i),\\
\mathrm{ext}_\alpha &= \lambda y^{\overline{\alpha}} j^s x_1^{\alpha_1} \dots x_n^{\alpha_n}.y(\mathrm{int}_{\alpha_1}(\lambda k^s.x_1)) \dots (\mathrm{int}_{\alpha_n}(\lambda k^s.x_n))j,
\end{aligned}$$

For example, if $q \in D_{s \to (e \to t) \to (e \to t) \to t}$, then

$$\mathrm{int}_{(e \to t) \to (e \to t) \to t}\, q =$$
$$\lambda y_1^{(s \to e) \to s \to t} y_2^{(s \to e) \to s \to t} i^s.qi(\lambda z^e.y_1(\lambda k^s.z)i)(\lambda z^e.y_2(\lambda k^s.z)i).$$

If qi is the quantifier *some* for all i, i.e.,

$$q = \lambda k^s x_1^{e \to t} x_2^{e \to t}.\exists_e(\lambda z^e.\wedge(x_1 z)(x_2 z)),$$

where $\exists_e \in D_{(e \to t) \to t}$ is the first-order existential quantifier over individuals and $\wedge \in D_{t \to t \to t}$ is conjunction, then $\mathrm{int}_{(e \to t) \to (e \to t) \to t}\, q$ equals

$$\lambda y_1^{(s \to e) \to s \to t} y_2^{(s \to e) \to s \to t} i^s.\exists_e(\lambda z^e.\wedge(y_1(\lambda k^s.z)i)(y_2(\lambda k^s.z)i)).$$

Lemma 2 *For any type α over A, $\lambda x^{s \to \alpha}.\,\mathrm{ext}_\alpha(\mathrm{int}_\alpha\, x) = \lambda x^{s \to \alpha}.x$.*

Proof By induction on α. The case of atomic types is obvious. For the non-atomic case, we have

$$\begin{aligned}
&\lambda x^{s\to\alpha\to\beta}.\,\mathrm{ext}_{\alpha\to\beta}(\mathrm{int}_{\alpha\to\beta}\,x)\\
&= \lambda x^{s\to\alpha\to\beta} j^s z^\alpha.\,\mathrm{ext}_\beta(\mathrm{int}_{\alpha\to\beta}\,x(\mathrm{int}_\alpha(\lambda k^s.z)))\,j\\
&= \lambda x^{s\to\alpha\to\beta} j^s z^\alpha.\,\mathrm{ext}_\beta(\mathrm{int}_\beta(\lambda i^s.xi(\mathrm{ext}_\alpha(\mathrm{int}_\alpha(\lambda k^s.z))i)))\,j\\
&= \lambda x^{s\to\alpha\to\beta} j^s z^\alpha.(\lambda i^s.xi((\lambda k^s.z)i))\,j \quad \text{by induction hypothesis}\\
&= \lambda x^{s\to\alpha\to\beta} j^s z^\alpha.xjz\\
&= \lambda x^{s\to\alpha\to\beta}.x
\end{aligned}$$

□

Clearly,

$$\lambda y^{\overline{\alpha}}.\,\mathrm{int}_\alpha(\mathrm{ext}_\alpha\,y) = \lambda y^{\overline{\alpha}}.y$$

does not hold in general. We call an object $y \in D_{\overline{\alpha}}$ *quasi-extensional* if $y = \mathrm{int}_\alpha(\mathrm{ext}_\alpha\,y)$. By Lemma 2, it is clear that $y \in D_{\overline{\alpha}}$ is quasi-extensional if and only if $y = \mathrm{int}_\alpha\,x$ for some $x \in D_{s\to\alpha}$. We call $y \in D_{\overline{\alpha}}$ *truly intensional* if it is not quasi-extensional.

Note that if $x \in D_{t^n\to t}$ (i.e., x is a truth function), then $\mathrm{int}_{t^n\to t}(\lambda k^s.x)$ coincides with its usual Boolean "generalization" to type $(s\to t)^n \to s \to t$. For instance, the intensionalization of conjunction $\wedge \in D_{t\to t\to t}$ is

$$\mathrm{int}_{t\to t\to t}(\lambda k^s.\wedge) = \lambda x^{s\to t} y^{s\to t} i^s.\wedge(xi)(yi),$$

that is, the intersection operation on subsets of D_s.[4]

In general, the intensionalization $\mathrm{int}_\alpha(\lambda k^s.x)$ corresponding to a "logical constant" $x \in D_\alpha$ does not necessarily agree with other existing ways of "lifting" x to an object in $D_{\overline{\alpha}}$. For instance, the first-order universal quantifier over individuals $\forall_e \in D_{(e\to t)\to t}$ can be naturally "lifted" to

$$\lambda y^{(s\to e)\to s\to t} i^s.\forall_{s\to e}(\lambda z^{s\to e}.yzi),$$

using the universal quantifier $\forall_{s\to e} \in D_{((s\to e)\to t)\to t}$ over *individual concepts* (i.e., functions from possible worlds to individuals), but this differs from its intensionalization,

$$\begin{aligned}
\mathrm{int}_{(e\to t)\to t}(\lambda k^s.\forall_e) &= \lambda y^{(s\to e)\to s\to t} i^s.\forall_e(\mathrm{ext}_{e\to t}\,y\;i)\\
&= \lambda y^{(s\to e)\to s\to t} i^s.\forall_e(\lambda z^e.y(\lambda k^s.z)i).
\end{aligned}$$

The point here is that the intensionalization of $\lambda k^s.x$ is λ-definable in terms of x, but the universal quantifier over objects of a higher type is not λ-definable in terms of the universal quantifier over objects of a lower type.[5]

[4] As is customary, we often identify sets with their characteristic functions.

[5] In fact, in the particular case of the universal quantifier, λ-definability goes the other way around: $\lambda k^s.\forall_\alpha$ is λ-definable in terms of $\forall_{\overline{\alpha}}$ by

$$\lambda i^s x^{\alpha\to t}.\forall_{\overline{\alpha}}(\lambda y^{\overline{\alpha}}.x(\mathrm{ext}_\alpha\,y\;i)).$$

(The same can be said of equality at different types.)

Another example is "generalized conjunction", defined for each type α of the form $\alpha_1 \to \cdots \to \alpha_n \to t$ by

$$\wedge_\alpha = \lambda y_1^\alpha y_2^\alpha z_1^{\alpha_1} \ldots z_n^{\alpha_n}.\wedge(y_1 z_1 \ldots z_n)(y_2 z_1 \ldots z_n).$$

For instance,

$$\begin{aligned}
&\mathrm{int}_{(e\to t)\to(e\to t)\to e\to t}(\lambda k^s.\wedge_{e\to t})\\
&= \lambda y_1^{(s\to e)\to s\to t} y_2^{(s\to e)\to s\to t} z_{s\to e} i^s.\wedge_{e\to t}(\mathrm{ext}_{e\to t}\, y_1\ i)(\mathrm{ext}_{e\to t}\, y_2\ i)(\mathrm{ext}_e\ z\ i)\\
&= \lambda y_1^{(s\to e)\to s\to t} y_2^{(s\to e)\to s\to t} z_{s\to e} i^s.\wedge_{e\to t}(\lambda x^e.y_1(\lambda k^s.x)i)(\lambda x^e.y_2(\lambda k^s.x)i)(zi)\\
&= \lambda y_1^{(s\to e)\to s\to t} y_2^{(s\to e)\to s\to t} z_{s\to e} i^s.\wedge(y_1(\lambda k^s.zi)i)(y_2(\lambda k^s.zi)i),
\end{aligned}$$

which does not equal

$$\wedge_{\overline{e\to t}} = \lambda y_1^{(s\to e)\to s\to t} y_2^{(s\to e)\to s\to t} z^{s\to e} i^s.\wedge(y_1 zi)(y_2 zi).$$

3 Preservation of Extensional Semantics Under Intensionalization

The object language of our study is that of typed λ-terms built up from basic expressions consisting of constants, each of some type α over A, and countably many variables for each type α over A. We use boldface variables $\boldsymbol{v}_1^\alpha, \boldsymbol{v}_2^\alpha, \ldots$ and boldface $\boldsymbol{\lambda}$ in the object language to avoid confusion with the metalanguage. Because of the presence of constants, whose interpretation we can pick at will, this choice of the object language is general enough to encompass most "extensional" languages of formal logic, and is also adequate as a language for representing meanings of expressions in extensional fragments of natural language in the style of Montague semantics. The intensionalization of the semantics of these languages serves as a foundation on which to build richer languages including intensional constructs within the usual framework of possible world semantics.

An *extensional model* M of our object language consists of base domains $(D_a)_{a\in A}$ and an assignment of a denotation $M(c) \in D_\alpha$ to each object language constant c of type α. An *intensional model* consists of base domains $(D_a)_{a\in A\cup\{s\}}$ together with an assignment of a denotation $M(c) \in D_{\overline{\alpha}}$ to each object language constant c of type α. We are interested in those intensional models that are built from extensional models by means of intensionalization.

An object language expression φ has the denotation $[\![\varphi]\!]_{M,g}$ in an (extensional or intensional) model M relative to an assignment g of values to variables. In an extensional model, $g(\boldsymbol{v}_l^\alpha) \in D_\alpha$, whereas in an intensional model, $g(\boldsymbol{v}_l^\alpha) \in D_{\overline{\alpha}}$. We let $g[x/\boldsymbol{v}_l^\alpha]$ denote the assignment that is like g except that it assigns x to $\boldsymbol{v}_l^\alpha$.

$$\begin{aligned}
[\![c]\!]_{M,g} &= M(c),\\
[\![\boldsymbol{v}_l^\alpha]\!]_{M,g} &= g(\boldsymbol{v}_l^\alpha),\\
[\![\varphi\psi]\!]_{M,g} &= [\![\varphi]\!]_{M,g}[\![\psi]\!]_{M,g},\\
[\![\boldsymbol{\lambda}\boldsymbol{v}_l^\alpha.\varphi]\!]_{M,g} &= \begin{cases} \lambda x^\alpha.[\![\varphi]\!]_{M,g[x/\boldsymbol{v}_l^\alpha]} & \text{if } M \text{ is an extensional model,}\\ \lambda x^{\overline{\alpha}}.[\![\varphi]\!]_{M,g[x/\boldsymbol{v}_l^\alpha]} & \text{if } M \text{ is an intensional model.}\end{cases}
\end{aligned}$$

Note that if φ is an object language expression of type α, the denotation of φ in an intensional model belongs to $D_{\overline{\alpha}}$.

If φ is a closed object language expression, $[\![\varphi]\!]_{M,g}$ does not depend on g, so we let $[\![\varphi]\!]_M = [\![\varphi]\!]_{M,g}$ for an arbitrarily chosen g.

Given an indexed collection $\mathcal{I} = (M_i)_{i \in I}$ of extensional models with the same base domains $(D_a)_{a \in A}$, we create an intensional model $M_{\mathcal{I}}$ based on $\mathcal{I}$, with $D_s = I$. For an object language constant c of type α, we let

$$M_{\mathcal{I}}(c) = \mathrm{int}_\alpha(\lambda i^s.M_i(c))$$

be its denotation in $M_{\mathcal{I}}$.

Let us consider a very simple example to illustrate the above definition. Let $A = \{t\}$ and let the vocabulary of the object language include constants c of type t and q of type $t \to t$. Fix $D_t = \{1, 0\}$. There are eight possible extensional models for this language: $M(c)$ is either 0 or 1, and $M(q)$ must be one of $\lambda x^t.0$, $\lambda x^t.1$, $\lambda x^t.x$, , and $\lambda x^t.\neg x$. Let $C = \{M_1, \dots, M_8\}$ be the set of these eight models and let I be a subset of $\{1, \dots, 8\}$. We can construct an intensional model $M_{\mathcal{I}}$ out of $\mathcal{I} = (M_i)_{i \in I}$ by the above definition. The denotation $M_{\mathcal{I}}(c)$ of c is (the characteristic function of) a subset of I, and $M_{\mathcal{I}}(q)$ is a function from the power set of I to the power set of I. We have

$$\begin{aligned} M_{\mathcal{I}}(q) &= \mathrm{int}_{t \to t}(\lambda i^s.M_i(q)) \\ &= \lambda y^{s \to t} i^s.M_i(q)(yi), \end{aligned}$$

so $M_{\mathcal{I}}(q)\ y\ i$ does not depend on the "global" property of y, but only on its value at i. In particular,

$$\begin{aligned} [\![qc]\!]_{M_{\mathcal{I}}} &= M_{\mathcal{I}}(q) M_{\mathcal{I}}(c) \\ &= (\lambda y^{s \to e} i^s.M_i(q)(yi))(\lambda i^s.M_i(c)) \\ &= \lambda i^s.M_i(q)M_i(c) \\ &= \lambda i^s.[\![qc]\!]_{M_i}. \end{aligned} \tag{3}$$

Having defined $M_{\mathcal{I}}$, one can then expand it to a model M' for a larger vocabulary including additional constants, such as the necessity operator $\Box$ (of type $t \to t$, with denotation in $D_{\overline{t \to t}}$), whose denotation is truly intensional.

We shall show that equalities exemplified by (3) are completely general and hold for all sentences in the extensional object language.

If g is an intensional assignment (suitable for $M_{\mathcal{I}}$), define an extensional assignment $g{\downarrow_i}$ by

$$g{\downarrow_i}(\boldsymbol{v}_l^\alpha) = \mathrm{ext}_\alpha\, g(\boldsymbol{v}_l^\alpha)\ i.$$

We call an intensional assignment g *quasi-extensional* if $g(\boldsymbol{v}_l^\alpha)$ is quasi-extensional for all $\boldsymbol{v}_l^\alpha$.

Lemma 3 *Let φ be an object language expression of type α, and let g be an intensional assignment suitable for $M_{\mathcal{I}}$ that is quasi-extensional.*

1. $\mathrm{ext}_\alpha [\![\varphi]\!]_{M_{\mathcal{I}},g}\ i = [\![\varphi]\!]_{M_i, g\downarrow_i}$.
2. If the β-normal form of φ is not a λ-abstract, $[\![\varphi]\!]_{M_{\mathcal{I}},g} = \mathrm{int}_\alpha(\lambda i^s.[\![\varphi]\!]_{M_i, g\downarrow_i})$.

Proof We prove both 1 and 2 simultaneously by induction on φ, assuming that φ is in β-normal form. Note that if φ is not a λ-abstract, 1 follows from 2 by Lemma 2.

If φ is a constant c, then 2 holds by the definition of $M_{\mathcal{I}}(c)$. If φ is a variable $\boldsymbol{v}_l^{\alpha}$, 2 holds since g is quasi-extensional.

Suppose $\varphi = \psi\chi$, where ψ is of type $\beta \to \alpha$ and χ is of type β. Since φ is in β-normal form, ψ is not a λ-abstract. Hence by induction hypothesis,

$$\begin{aligned}
[\![\psi]\!]_{M_{\mathcal{I}},g} &= \mathrm{int}_{\beta\to\alpha}(\lambda i^s.[\![\psi]\!]_{M_i,g\downarrow_i}) \\
&= \lambda y_0^{\overline{\alpha}}.\,\mathrm{int}_{\alpha}(\lambda i^s.[\![\psi]\!]_{M_i,g\downarrow_i}(\mathrm{ext}_{\beta}\, y_0\; i)).
\end{aligned}$$

The induction hypothesis about χ gives

$$\mathrm{ext}_{\beta}[\![\chi]\!]_{M_{\mathcal{I}},g}\; i = [\![\chi]\!]_{M_i,g\downarrow_i}.$$

Thus,

$$\begin{aligned}
[\![\psi\chi]\!]_{M_{\mathcal{I}},g} &= [\![\psi]\!]_{M_{\mathcal{I}},g}[\![\chi]\!]_{M_{\mathcal{I}},g} \\
&= \mathrm{int}_{\alpha}(\lambda i^s.[\![\psi]\!]_{M_i,g\downarrow_i}(\mathrm{ext}_{\beta}[\![\chi]\!]_{M_{\mathcal{I}},g}\; i)) \\
&= \mathrm{int}_{\alpha}(\lambda i^s.[\![\psi]\!]_{M_i,g\downarrow_i}[\![\chi]\!]_{M_i,g\downarrow_i}) \\
&= \mathrm{int}_{\alpha}(\lambda i^s.[\![\psi\chi]\!]_{M_i,g\downarrow_i})
\end{aligned}$$

and the condition in 2 holds of φ.

It remains to consider the case where φ is a λ-abstract $\boldsymbol{\lambda u}^{\beta}.\psi$, where $\alpha = \beta\to\gamma$ and ψ is of type γ. By induction hypothesis, $\mathrm{ext}_{\gamma}[\![\psi]\!]_{M_{\mathcal{I}},h}\; i = [\![\psi]\!]_{M_i,h\downarrow_i}$ for all assignments h that are quasi-extensional.

$$\begin{aligned}
\mathrm{ext}_{\beta\to\gamma}[\![\boldsymbol{\lambda u}^{\beta}.\psi]\!]_{M_{\mathcal{I}},g}\; i &= \lambda x^{\beta}.\,\mathrm{ext}_{\gamma}([\![\boldsymbol{\lambda u}^{\beta}.\psi]\!]_{M_{\mathcal{I}},g}(\mathrm{int}_{\beta}(\lambda k^s.x)))\; i \\
&= \lambda x^{\beta}.\,\mathrm{ext}_{\gamma}([\![\psi]\!]_{M_{\mathcal{I}},g[\mathrm{int}_{\beta}(\lambda k^s.x)/\boldsymbol{u}^{\beta}]})\; i \\
&= \lambda x^{\beta}.[\![\psi]\!]_{M_i,g[\mathrm{int}_{\beta}(\lambda k^s.x)/\boldsymbol{u}^{\beta}]\downarrow_i} \quad \text{by induction hypothesis} \\
&= \lambda x^{\beta}.[\![\psi]\!]_{M_i,g\downarrow_i[x/\boldsymbol{u}^{\beta}]} \\
&= [\![\boldsymbol{\lambda u}^{\beta}.\psi]\!]_{M_i,g\downarrow_i}.
\end{aligned}$$

(Note that $g[\mathrm{int}_{\beta}(\lambda k^s.x)/\boldsymbol{u}^{\beta}]$ is quasi-extensional.) Thus, 1 holds of φ. □

Note that the condition in part 2 of Lemma 3 does not hold of λ-abstracts. A simple counterexample is the **I** combinator $\boldsymbol{\lambda u}^{a\to b}.\boldsymbol{u}^{a\to b}$:

$$\begin{aligned}
&\mathrm{int}_{(a\to b)\to a\to b}(\lambda i^s.[\![\boldsymbol{\lambda u}^{a\to b}.\boldsymbol{u}^{a\to b}]\!]_{M_i}) \\
&= \mathrm{int}_{(a\to b)\to a\to b}(\lambda i^s x^{a\to b}.x) \\
&= \lambda y_1^{(s\to a)\to s\to b} y_2^{s\to a} i^s.\,\mathrm{ext}_{a\to b}\, y_1\; i\; (\mathrm{ext}_a\, y_2\; i) \\
&= \lambda y_1^{(s\to a)\to s\to b} y_2^{s\to a} i^s.(\lambda z^a.y_1(\mathrm{int}_a(\lambda k.z))i)(\mathrm{ext}_a\, y_2\; i) \\
&= \lambda y_1^{(s\to a)\to s\to b} y_2^{s\to a} i^s.(\lambda z^a.y_1(\lambda k.z)i)(y_2 i) \\
&= \lambda y_1^{(s\to a)\to s\to b} y_2^{s\to a} i^s.y_1(\lambda k.y_2 i)i \\
&\neq \lambda y_1^{(s\to a)\to s\to b} y_2^{s\to a} i^s.y_1 y_2 i
\end{aligned}$$

$$
\begin{aligned}
&= \lambda y_1^{(s\to a)\to s\to b}.y_1 \\
&= \llbracket \boldsymbol{\lambda u}^{a\to b}.\boldsymbol{u}^{a\to b} \rrbracket_{M_{\mathcal{I}}}.
\end{aligned}
$$

(The inequality assumes $|D_a| \geq 2$ and $|D_s| \geq 2$.)

Remark 4 A special case of Lemma 3 is when $g = h^*$ for some extensional assignment h, where h^* is defined by

$$h^*(\boldsymbol{v}_l^{\alpha}) = \mathrm{int}_{\alpha}(\lambda k^s.h(\boldsymbol{v}_l^{\alpha})).$$

In this case, we have $g{\downarrow_i} = h$ for all $i \in I$. This special case itself can be proved directly by induction.

Remark 5 The content of Lemma 3 can be stated entirely within simply typed λ-calculus, as follows. If φ is a λ-term of type α, with free variables $z_1^{\beta_1}, \dots, z_n^{\beta_n}$, let $\overline{\varphi}$ be the λ-term of type $\overline{\alpha}$ obtained from φ by replacing each occurrence of $a \in A$ by $s \to a$ in the type annotation of φ. Then we have

$$
\begin{aligned}
&\mathrm{ext}_{\alpha}\,\overline{\varphi}[(\mathrm{int}_{\beta_1}\,x_1^{s\to\beta_1})/z_1^{\overline{\beta_1}}, \dots, (\mathrm{int}_{\beta_n}\,x_n^{s\to\beta_n})/z_n^{\overline{\beta_n}}] \\
&\qquad\qquad =_{\beta\eta} \lambda i^s.\varphi[(x_1^{s\to\beta_1}\,i)/z_1^{\beta_1}, \dots, (x_n^{s\to\beta_n}\,i)/z_n^{\beta_n}],
\end{aligned}
$$

and if the β-normal form of φ is not a λ-abstract,

$$
\begin{aligned}
&\overline{\varphi}[(\mathrm{int}_{\beta_1}\,x_1^{s\to\beta_1})/z_1^{\overline{\beta_1}}, \dots, (\mathrm{int}_{\beta_n}\,x_n^{s\to\beta_n})/z_n^{\overline{\beta_n}}] \\
&\qquad\qquad =_{\beta\eta} \mathrm{int}_{\alpha}(\lambda i^s.\varphi[(x_1^{s\to\beta_1}\,i)/z_1^{\beta_1}, \dots, (x_n^{s\to\beta_n}\,i)/z_n^{\beta_n}]).
\end{aligned}
$$

Theorem 6 *For any closed object language expression φ of type $a \in A$, $\llbracket\varphi\rrbracket_{M_{\mathcal{I}}} = \lambda i^s.\llbracket\varphi\rrbracket_{M_i}$.*

Proof Immediate from Lemma 3. □

Now assume $t \in A$ and fix $D_t = \{1, 0\}$. We call an object language expression of type t a *formula*, and a closed formula a *sentence*. A *pointed possible world model* is a pair of the form $(M_{\mathcal{I}}, i)$ where $i \in I$. The extensional model M_i is the *actual world* of a pointed possible world model $(M_{\mathcal{I}}, i)$. We say that a sentence φ is true in a pointed possible world model $(M_{\mathcal{I}}, i)$ if $\llbracket\varphi\rrbracket_{M_{\mathcal{I}}}\, i = 1$.

Corollary 7 *For every sentence φ in the object language and every extensional model M, the following are equivalent:*

1. *φ is true in M.*
2. *φ is true in any pointed possible world model whose actual world is M.*

Let C be a class of extensional models. Call a sentence φ *extensionally valid in C* if $\llbracket\varphi\rrbracket_M = 1$ for all $M \in C$, and *intensionally valid in C* if $\llbracket\varphi\rrbracket_{M_{\mathcal{I}}} = \lambda i^s.1$ for all indexed collections $\mathcal{I} = (M_i)_{i\in I}$ consisting of models in C that share the same base domains. Similarly, ψ is an *extensional consequence of $\varphi_1, \dots, \varphi_n$ in C* if $\llbracket\varphi_1\rrbracket_M = \cdots = \llbracket\varphi_n\rrbracket_M = 1$ implies $\llbracket\psi\rrbracket_M = 1$ for all $M \in C$, and ψ is an *intensional consequence of $\varphi_1, \dots, \varphi_n$ in C* if $\llbracket\varphi_1\rrbracket_{M_{\mathcal{I}}} \cap \cdots \cap \llbracket\varphi_n\rrbracket_{M_{\mathcal{I}}} \subseteq \llbracket\psi\rrbracket_{M_{\mathcal{I}}}$ for all indexed collections $\mathcal{I}$ consisting of models in C with the same base domains.[6]

[6] The intensional consequence relation as defined here corresponds to *local* consequence in modal logic (Blackburn et al 2001).

Corollary 8 *Let C be a class of extensional models and let $\varphi, \varphi_1, \dots, \varphi_n, \psi$ be sentences.*

1. *If φ is extensionally valid in C, then φ is intensionally valid in C.*
2. *If ψ is an extensional consequence of $\varphi_1, \dots, \varphi_n$ in C, then ψ is an intensional consequence of $\varphi_1, \dots, \varphi_n$ in C.*

In the presence of conjunction ($\wedge$) and implication ($\rightarrow$) in the object language, the consequence relation between $\varphi_1, \dots, \varphi_n$ and ψ can be defined as the validity of $\varphi_1 \wedge \cdots \wedge \varphi_n \rightarrow \psi$, both in the extensional and in the intensional sense. (Recall that truth-functional connectives behave as desired in intensional models.) This allows us to concentrate on validity.

4 Intensionally Valid Schemata

Corollary 8 does not quite give what Ben-Avi and Winter (2007) were aiming for, because in their method of intensionalization, the denotation of a non-logical constant of type α is not restricted to quasi-extensional objects in $D_{\overline{\alpha}}$. They start from a class C of extensional models that is closed under arbitrary change in the denotations of non-logical constants and obtain by intensionalization a class C' of intensional models that is again closed under arbitrary change in the denotations of non-logical constants. (In their method, intensionalization is only used to determine the denotations of logical constants in intensional models.) Replacing non-logical constants with free variables, we can say in our setting that what they were aiming for was preservation of validity of (and the consequence relation among) open formulas or *schemata*. This is clearly an unreasonably high demand and is impossible to achieve in any general terms;[7] be that as it may, it will be instructive to see the limited extent to which the present method of intensionalization preserves validity of open formulas.

The generalization of the notion of validity to open formulas is the standard one. Let C be a class of extensional models. For an object language expression φ of type t, we say that φ is *extensionally valid in C* if $[\![\varphi]\!]_{M,g} = 1$ for all $M \in C$ and all extensional assignments g suitable for M; we say that φ is *intensionally valid in C* if $[\![\varphi]\!]_{M_{\mathcal{I}},g} = \lambda i^s.1$ for all indexed collections $\mathcal{I}$ consisting of models in C built on the same base domains and all intensional assignments g suitable for $M_{\mathcal{I}}$.

Lemma 3 does not imply that the validity of an open formula is preserved when one moves from extensional models to intensional models created out of them, because not all intensional assignments are quasi-extensional. For example, let

$$\varphi = \rightarrow(\wedge(\boldsymbol{u}^{t\rightarrow t}\top)(\boldsymbol{u}^{t\rightarrow t}\bot))(\boldsymbol{u}^{t\rightarrow t}\boldsymbol{v}^t),$$

or, in a more readable style,

$$(\boldsymbol{u}^{t\rightarrow t}\top \wedge \boldsymbol{u}^{t\rightarrow t}\bot) \rightarrow \boldsymbol{u}^{t\rightarrow t}\boldsymbol{v}^t. \tag{4}$$

[7] As mentioned in the introduction, Ben-Avi and Winter (2007) opted to restrict the types of non-logical constants to those with a very special form.

Let $\rightarrow, \wedge, \top, \bot$ have the usual interpretation in M_i for all $i \in I$. Then φ is extensionally valid in $\mathcal{I} = \{\, M_i \mid i \in I \,\}$, but it is easy to see that there are intensional assignments g such that

$$[\![\varphi]\!]_{M_{\mathcal{I}},g} \neq \lambda i^s.1.$$

The reason that an extensionally valid formula φ with $\mathrm{FV}(\varphi) \neq \varnothing$ need not be intensionally valid is related to the failure of the equality $[\![\psi]\!]_{M_{\mathcal{I}}} = \mathrm{int}_\alpha(\lambda i^s.[\![\psi]\!]_{M_i})$ when ψ is a (closed) λ-abstract. Let $\mathrm{FV}(\varphi) = \{\boldsymbol{u}_1^{\alpha_1}, \dots, \boldsymbol{u}_n^{\alpha_n}\}$. We have $[\![\varphi]\!]_{M_i,g} = 1$ for all extensional assignments g if and only if $[\![\boldsymbol{\lambda u}_1^{\alpha_1} \dots \boldsymbol{u}_n^{\alpha_n}.\varphi]\!]_{M_i} = \lambda x_1^{\alpha_1} \dots x_n^{\alpha_n}.1$. Also, $[\![\varphi]\!]_{M_{\mathcal{I}},g} = \lambda i^s.1$ for all intensional assignments g if and only if $[\![\boldsymbol{\lambda u}_1^{\alpha_1} \dots \boldsymbol{u}_n^{\alpha_n}.\varphi]\!]_{M_{\mathcal{I}}} = \lambda y_1^{\overline{\alpha_1}} \dots y_n^{\overline{\alpha_n}} i^s.1$. Now suppose $[\![\varphi]\!]_{M_i,g} = 1$ for all $i \in I$ and all extensional assignments g. Then $[\![\boldsymbol{\lambda u}_1^{\alpha_1} \dots \boldsymbol{u}_n^{\alpha_n}.\varphi]\!]_{M_i} = \lambda x_1^{\alpha_1} \dots x_n^{\alpha_n}.1$, and this clearly implies

$$\mathrm{int}_{\alpha_1 \to \dots \to \alpha_n \to t}(\lambda i^s.[\![\boldsymbol{\lambda u}_1^{\alpha_1} \dots \boldsymbol{u}_n^{\alpha_n}.\varphi]\!]_{M_i}) = \lambda y_1^{\overline{\alpha_1}} \dots y_n^{\overline{\alpha_n}} i^s.1.$$

But since $[\![\boldsymbol{\lambda u}_1^{\alpha_1} \dots \boldsymbol{u}_n^{\alpha_n}.\varphi]\!]_{M_{\mathcal{I}}} = \mathrm{int}_{\alpha_1 \to \dots \to \alpha_n \to t}(\lambda i^s.[\![\boldsymbol{\lambda u}_1^{\alpha_1} \dots \boldsymbol{u}_n^{\alpha_n}.\varphi]\!]_{M_i})$ need not hold, we cannot infer $[\![\boldsymbol{\lambda u}_1^{\alpha_1} \dots \boldsymbol{u}_n^{\alpha_n}.\varphi]\!]_{M_{\mathcal{I}}} = \lambda y_1^{\overline{\alpha_1}} \dots y_n^{\overline{\alpha_n}} i^s.1$.

The open formula (4) should be clearly distinguished from its closure

$$\forall_{t \to t}(\boldsymbol{\lambda u}^{t \to t}.\forall_t(\boldsymbol{\lambda v}^t.(\boldsymbol{u}^{t \to t}\top \wedge \boldsymbol{u}^{t \to t}\bot) \to \boldsymbol{u}^{t \to t}\boldsymbol{v}^t)) \tag{5}$$

or the open formulas in one free variable

$$\forall_{t \to t}(\boldsymbol{\lambda u}^{t \to t}.(\boldsymbol{u}^{t \to t}\top \wedge \boldsymbol{u}^{t \to t}\bot) \to \boldsymbol{u}^{t \to t}\boldsymbol{v}^t), \tag{6}$$

$$\forall_t(\boldsymbol{\lambda v}^t.(\boldsymbol{u}^{t \to t}\top \wedge \boldsymbol{u}^{t \to t}\bot) \to \boldsymbol{u}^{t \to t}\boldsymbol{v}^t). \tag{7}$$

The three formulas (5), (6), (7), unlike (4), remain valid in intensional models. As mentioned above, the intensionalization of the universal quantifier $\forall_\alpha$ still only quantifies over objects in D_α, so the intensional validity of (5), (6), (7) does not imply the intensional validity of (4).

The intensional validity of (6) and (7) illustrates the fact that extensionally valid open formulas may remain intensionally valid in certain restricted cases. In what follows, we give one sufficient condition for an extensionally valid φ to be intensionally valid.

Fix an indexed collection $\mathcal{I} = (M_i)_{i \in I}$ of extensional models. We call an object language constant c *rigid* (in $\mathcal{I}$) if $M_i(c) = M_j(c)$ for all $i, j \in I$.

Let V be a set of object language variables. Let φ be an object language expression. We define two predicates *V-safe* and *V-protected* by simultaneous induction as follows:

- φ is V-safe if and only if one of the following conditions holds:
  1. φ is a constant or a variable.
  2. $\varphi = \psi\chi$ and either
     - ψ is V-protected and χ is V-safe, or
     - ψ is V-safe, all constants that occur in χ are rigid, $\mathrm{FV}(\chi) \cap V = \varnothing$, and χ is $\varnothing$-protected.
  3. $\varphi = \boldsymbol{\lambda v}.\psi$ and ψ is V-safe.
- φ is V-protected if and only if one of the following conditions holds:

1. φ is a constant or a variable not in V.
2. $\varphi = \psi\chi$ and ψ is V-protected and χ is V-safe.
3. $\varphi = \boldsymbol{\lambda v}.\psi$ and ψ is $V \cup \{\boldsymbol{v}\}$-protected.

More informally, if $\boldsymbol{\lambda u}_1^{\beta_1} \ldots \boldsymbol{u}_n^{\beta_n}.\chi$ occurs in a V-safe formula as an argument of a variable in V, then χ cannot contain any non-rigid constants or variables in V, and χ must be a $\{\boldsymbol{u}_1^{\beta_1}, \ldots, \boldsymbol{u}_n^{\beta_n}\}$-safe formula that does not start with one of $\boldsymbol{u}_1^{\beta_1}, \ldots, \boldsymbol{u}_n^{\beta_n}$.

Lemma 9 *Let V be a set of object language variables, and let φ be an object language expression of type α. Suppose that g is an intensional assignment such that for all variables $\boldsymbol{u}^\delta \in \mathrm{FV}(\varphi) - V$, we have $g(\boldsymbol{u}^\delta) = \mathrm{int}_\delta(\lambda k^s.x)$ for some $x \in D_\delta$. The following hold:*

1. If φ is V-protected, $[\![\varphi]\!]_{M_\mathcal{I},g} = \mathrm{int}_\alpha(\lambda i^s.[\![\varphi]\!]_{M_i,g\downarrow_i})$.
2. If φ is V-safe, $\mathrm{ext}_\alpha[\![\varphi]\!]_{M_\mathcal{I},g}\, i = [\![\varphi]\!]_{M_i,g\downarrow_i}$.

Proof We prove 1 and 2 by simultaneous induction on φ. Note that the equality in 1 implies the equality in 2, so when φ is V-protected, it suffices to prove the former.

Induction basis.

Case 1. φ is a constant c. In this case, φ is V-protected. We have $[\![c]\!]_{M_\mathcal{I},g} = M_\mathcal{I}(c) = \mathrm{int}_\alpha(\lambda i^s.M_i(c)) = \mathrm{int}_\alpha(\lambda i^s.[\![c]\!]_{M_i,g\downarrow_i})$ by the definition of $M_\mathcal{I}(c)$.

Case 2. φ is a variable $\boldsymbol{v}^\alpha$. In this case, φ is V-protected if and only if $\boldsymbol{v}^\alpha \notin V$. We have $\mathrm{ext}_\alpha[\![\boldsymbol{v}^\alpha]\!]_{M_\mathcal{I},g}\, i = \mathrm{ext}_\alpha\, g(\boldsymbol{v}^\alpha)\, i = g\downarrow_i(\boldsymbol{v}^\alpha) = [\![\boldsymbol{v}^\alpha]\!]_{M_i,g\downarrow_i}$ by the definition of $g\downarrow_i$, so the equality in 2 holds. If $\boldsymbol{v}^\alpha \notin V$, then by the assumption on g, we have $[\![\boldsymbol{v}^\alpha]\!]_{M_\mathcal{I},g} = g(\boldsymbol{v}^\alpha) = \mathrm{int}(\lambda i^s.g\downarrow_i(\boldsymbol{v}^\alpha)) = \mathrm{int}(\lambda i^s.[\![\boldsymbol{v}^\alpha]\!]_{M_i,g\downarrow_i})$, so the equality in 1 holds.

Induction step.

Case 1. $\varphi = \psi\chi$, where ψ is of type $\beta \to \alpha$ and χ is of type β.

Case 1a. ψ is V-protected and χ is V-safe. In this case, φ is V-protected (as well as V-safe). By induction hypothesis, $[\![\psi]\!]_{M_\mathcal{I},g} = \mathrm{int}_{\beta\to\alpha}(\lambda i^s.[\![\psi]\!]_{M_i,g\downarrow_i})$ and $\mathrm{ext}_\beta[\![\chi]\!]_{M_\mathcal{I},g}\, i = [\![\chi]\!]_{M_i,g\downarrow_i}$. Hence

$$\begin{aligned}
[\![\psi\chi]\!]_{M_\mathcal{I},g} &= [\![\psi]\!]_{M_\mathcal{I},g}[\![\chi]\!]_{M_\mathcal{I},g}\\
&= \mathrm{int}_{\beta\to\alpha}(\lambda i^s.[\![\psi]\!]_{M_i,g\downarrow_i})[\![\chi]\!]_{M_\mathcal{I},g}\\
&= \mathrm{int}_\alpha(\lambda i^s.[\![\psi]\!]_{M_i,g\downarrow_i}(\mathrm{ext}_\beta[\![\chi]\!]_{M_\mathcal{I},g}\, i))\\
&= \mathrm{int}_\alpha(\lambda i^s.[\![\psi]\!]_{M_i,g\downarrow_i}[\![\chi]\!]_{M_i,g\downarrow_i})\\
&= \mathrm{int}_\alpha(\lambda i^s.[\![\psi\chi]\!]_{M_i,g\downarrow_i}),
\end{aligned}$$

and the condition in 1 is satisfied.

Case 1b. ψ is V-safe, all constants that occur in χ are rigid, $\mathrm{FV}(\chi) \cap V = \varnothing$, and χ is $\varnothing$-protected. In this case, φ is V-safe. Note that $\mathrm{FV}(\chi) \cap V = \varnothing$ implies that $g(\boldsymbol{u}^\delta) = \mathrm{int}_\delta(\lambda k^s.x)$ for some $x \in D_\delta$ for all $\boldsymbol{u}^\delta \in \mathrm{FV}(\chi)$. Hence, the induction hypothesis applies to both ψ and χ and we get $\mathrm{ext}_{\beta\to\alpha}[\![\psi]\!]_{M_\mathcal{I},g} = [\![\psi]\!]_{M_i,g\downarrow_i}$ and $[\![\chi]\!]_{M_\mathcal{I},g} = \mathrm{int}_\beta(\lambda i^s.[\![\chi]\!]_{M_i,g\downarrow_i})$. The fact that $g(\boldsymbol{u}^\delta) = \mathrm{int}_\delta(\lambda k^s.x)$ for some $x \in D_\delta$ for all $\boldsymbol{u}^\delta \in \mathrm{FV}(\chi)$ also implies that $g\downarrow_i$ and $g\downarrow_j$ agree on $\mathrm{FV}(\chi)$ for all $i, j \in I$. Since $M_i(c) = M_j(c)$ for all $i, j \in I$ for all constants c in χ, we see that $[\![\chi]\!]_{M_i,g\downarrow_i} = [\![\chi]\!]_{M_j,g\downarrow_j}$ for all $i, j \in I$. Thus,

$$\mathrm{ext}_\alpha[\![\psi\chi]\!]_{M_\mathcal{I},g}\, i = \mathrm{ext}_\alpha([\![\psi]\!]_{M_\mathcal{I},g}[\![\chi]\!]_{M_\mathcal{I},g})\, i$$

$$
\begin{aligned}
&= \mathrm{ext}_\alpha(\llbracket\psi\rrbracket_{M_\mathcal{I},g}(\mathrm{int}_\beta(\lambda k^s.\llbracket\chi\rrbracket_{M_k,g\downarrow_k})))\ i\\
&= \mathrm{ext}_\alpha(\llbracket\psi\rrbracket_{M_\mathcal{I},g}(\mathrm{int}_\beta(\lambda k^s.\llbracket\chi\rrbracket_{M_i,g\downarrow_i})))\ i\\
&\qquad\qquad \text{since } \llbracket\chi\rrbracket_{M_k,g_k} = \llbracket\chi\rrbracket_{M_i,g\downarrow_i} \text{ for all } k \in I\\
&= \mathrm{ext}_{\beta\to\alpha}\llbracket\psi\rrbracket_{M_\mathcal{I},g}\ i\ \llbracket\chi\rrbracket_{M_i,g\downarrow_i}\\
&= \llbracket\psi\rrbracket_{M_i,g\downarrow_i}\llbracket\chi\rrbracket_{M_i,g\downarrow_i}\\
&= \llbracket\psi\chi\rrbracket_{M_i,g\downarrow_i},
\end{aligned}
$$

and the condition in 2 holds.

Case 2. $\varphi = \boldsymbol{\lambda v}^\beta.\psi$, where ψ is of type γ and $\alpha = \beta \to \gamma$.

Case 2a. ψ is $V \cup \{\boldsymbol{v}^\beta\}$-protected. In this case, φ is V-protected. By induction hypothesis, $\llbracket\psi\rrbracket_{M_\mathcal{I},h} = \mathrm{int}(\lambda i^s.\llbracket\psi\rrbracket_{M_i,h\downarrow_i})$ holds of all h satisfying the following condition:

$$\text{for all } \boldsymbol{u}^\delta \in \mathrm{FV}(\psi) - (V \cup \{\boldsymbol{v}^\beta\}), \text{ there is an } x \in D_\delta \text{ such that } h(\boldsymbol{u}^\delta) = \mathrm{int}_\delta(\lambda k^s.x). \tag{8}$$

We see

$$
\begin{aligned}
\llbracket\boldsymbol{\lambda v}^\beta.\psi\rrbracket_{M_\mathcal{I},g} &= \lambda y^{\overline{\beta}}.\llbracket\psi\rrbracket_{M_\mathcal{I},g[y/\boldsymbol{v}^\beta]}\\
&= \lambda y^{\overline{\beta}}.\,\mathrm{int}_\gamma(\lambda i^s.\llbracket\psi\rrbracket_{M_i,g[y/\boldsymbol{v}^\beta]\downarrow_i})\\
&= \lambda y^{\overline{\beta}}.\,\mathrm{int}_\gamma(\lambda i^s.\llbracket\psi\rrbracket_{M_i,g\downarrow_i[\mathrm{ext}_\beta\, y\, i/\boldsymbol{v}^\beta]})\\
&= \lambda y^{\overline{\beta}}.\,\mathrm{int}_\gamma(\lambda i^s.\llbracket\boldsymbol{\lambda v}^\beta.\psi\rrbracket_{M_i,g\downarrow_i}(\mathrm{ext}_\beta\, y\ i))\\
&= \mathrm{int}_{\beta\to\gamma}(\lambda i^s.\llbracket\boldsymbol{\lambda v}^\beta.\psi\rrbracket_{M_i,g\downarrow_i}).
\end{aligned}
$$

(Note that $h = g[y/\boldsymbol{v}^\beta]$ satisfies (8).) Hence the condition in 1 is satisfied.

Case 2b. ψ is V-safe. In this case, φ is V-safe. By induction hypothesis, $\mathrm{ext}_\gamma\llbracket\psi\rrbracket_{M_\mathcal{I},h}\ i = \llbracket\psi\rrbracket_{M_i,h\downarrow_i}$ holds of all h satisfying the following condition:

$$\text{for all } \boldsymbol{u}^\delta \in \mathrm{FV}(\psi) - V, \text{ there is an } x \in D_\delta \text{ such that } h(\boldsymbol{u}^\delta) = \mathrm{int}_\delta(\lambda k^s.x). \tag{9}$$

We have

$$
\begin{aligned}
\mathrm{ext}_{\beta\to\gamma}\llbracket\boldsymbol{\lambda v}^\beta.\psi\rrbracket_{M_\mathcal{I},g}\ i &= \lambda x^\beta.\,\mathrm{ext}_\gamma(\llbracket\boldsymbol{\lambda v}^\beta.\psi\rrbracket_{M_\mathcal{I},g}(\mathrm{int}_\beta(\lambda k^s.x)))\ i\\
&= \lambda x^\beta.\,\mathrm{ext}_\gamma\llbracket\psi\rrbracket_{M_\mathcal{I},g[\mathrm{int}_\beta(\lambda k^s.x)/\boldsymbol{v}^\beta]}\ i\\
&= \lambda x^\beta.\llbracket\psi\rrbracket_{M_i,g[\mathrm{int}_\beta(\lambda k^s.x)/\boldsymbol{v}^\beta]\downarrow_i}\\
&= \lambda x^\beta.\llbracket\psi\rrbracket_{M_i,g\downarrow_i[x/\boldsymbol{v}^\beta]}\\
&= \llbracket\boldsymbol{\lambda v}^\beta.\psi\rrbracket_{M_i,g\downarrow_i}.
\end{aligned}
$$

(Note that $\mathrm{FV}(\psi) \subseteq \mathrm{FV}(\varphi) \cup \{\boldsymbol{v}^\beta\}$ and $h = g[\mathrm{int}_\beta(\lambda k^s.x)/\boldsymbol{v}^\beta]$ satisfies (9).) Thus, the condition in 2 is satisfied.

This completes the induction step. □

Remark 10 Remark 5 applies, mutatis mutandis, to Lemma 9 as well.

Theorem 11 *Let φ be a formula of the object language that is* $\mathrm{FV}(\varphi)$*-safe. If φ is extensionally valid in a class C of extensional models, then φ is intensionally valid in C.*

Proof Let $\mathcal{I} = (M_i)_{i \in I}$ be an indexed collection of extensional models in C that share the same base domains. Let g be an arbitrary intensional assignment suitable for $M_\mathcal{I}$. Then $[\![\varphi]\!]_{M_\mathcal{I},g} = \mathrm{ext}_t[\![\varphi]\!]_{M_\mathcal{I},g} = \lambda i^s.[\![\varphi]\!]_{M_i,g\downarrow_i}$ by Lemma 9. Thus, if $[\![\varphi]\!]_{M_i,h} = 1$ for all $i \in I$ and all extensional assignments h suitable for M_i, then $[\![\varphi]\!]_{M_\mathcal{I},g} = \lambda i^s.1$. □

Here are some examples illustrating the scope of applicability of Theorem 11. First, all tautologies of propositional logic are intensionally valid. It is easy to see that all formulas built from propositional variables in V are V-safe, because propositional variables are V-safe and truth-functional connectives are V-protected. In fact, we need not invoke Theorem 11 in this case, because all objects in $D_{s \to t}$ are quasi-extensional. Of course, the fact that propositional tautologies are intensionally valid just means that the power set of D_s is a Boolean algebra.

A less trivial example is Aristotelian syllogisms, which are of the form

$$Q_1 \boldsymbol{u}_1^{e\to t} \boldsymbol{u}_2^{e\to t} \wedge Q_2 \boldsymbol{u}_3^{e\to t} \boldsymbol{u}_4^{e\to t} \to Q_3 \boldsymbol{u}_5^{e\to t} \boldsymbol{u}_6^{e\to t},$$

where $\boldsymbol{u}_1^{e\to t}, \ldots, \boldsymbol{u}_6^{e\to t}$ are not necessarily distinct variables and Q_1, Q_2, Q_3 are not necessarily distinct constants of type $(e \to t) \to (e \to t) \to t$. Formulas of this form are $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_6\}$-safe, so if they are extensionally valid, one can instantiate $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_6$ by expressions denoting truly intensional *properties* (functions from individual concepts to sets of possible worlds).

What about first-order logic? Of the usual Hilbert-style axioms,

$$\forall x(\varphi(x) \to \psi(x)) \to (\forall x\, \varphi(x) \to \forall x\, \psi(x))$$

is rendered as

$$\forall_e(\lambda \boldsymbol{v}^e.\boldsymbol{u}_1^{e\to t}\boldsymbol{v}^e \to \boldsymbol{u}_2^{e\to t}\boldsymbol{v}^e) \to (\forall_e(\lambda \boldsymbol{v}^e.\boldsymbol{u}_1^{e\to t}\boldsymbol{v}^e) \to \forall_e(\lambda \boldsymbol{v}^e.\boldsymbol{u}_2^{e\to t}\boldsymbol{v}^e)),$$

where $\to$ (of type $t{\to}t{\to}t$) is written as an infix operator. This object language formula is $\{\boldsymbol{u}_1^{e\to t}, \boldsymbol{u}_2^{e\to t}\}$-safe and is hence intensionally valid, assuming the usual interpretation of $\forall_e$ and $\to$.

In contrast,

$$\forall x\, \varphi(x) \to \varphi(t)$$

is rendered as

$$\forall_e(\lambda \boldsymbol{v}^e.\boldsymbol{u}^{e\to t}\boldsymbol{v}^e) \to \boldsymbol{u}^{e\to t}\boldsymbol{t}^e,$$

which is not $\{\boldsymbol{u}^{e\to t}, \boldsymbol{t}^e\}$-safe. Indeed, it is not intensionally valid, because not all individual concepts are constant functions.

Another axiom that is not intensionally valid, this time from first-order logic with equality, is

$$s = t \to (\varphi(s) \to \varphi(t)),$$

which is rendered as

$$\boldsymbol{s}^e = \boldsymbol{t}^e \to (\boldsymbol{v}^{e\to t}\boldsymbol{s}^e \to \boldsymbol{v}^{e\to t}\boldsymbol{t}^e),$$

where $=$ stands for the equality between individuals in D_e. This formula is not $\{\boldsymbol{s}^e, \boldsymbol{t}^e, \boldsymbol{v}^{e\to t}\}$-safe and is not intensionally valid. This is just the well-known failure of substitutivity in intensional contexts.

Here are a couple of more artificial examples. Let $\mathbf{I}$ be an object language constant that denotes the identity function on $D_{e\to t}$ in all models in C. Then

$$\boldsymbol{u}^{((e\to t)\to e\to t)\to t}\mathbf{I} \to \boldsymbol{u}^{((e\to t)\to e\to t)\to t}(\boldsymbol{\lambda v}^{e\to t}.\boldsymbol{v}^{e\to t})$$

is extensionally valid in C, but not intensionally so. This is because the intensionalization of the identity function on $D_{e\to t}$ is not the identity function on $D_{(s\to e)\to s\to t}$. Note that this formula is not $\{\boldsymbol{u}^{((e\to t)\to e\to t)\to t}\}$-safe because $\boldsymbol{\lambda v}^{e\to t}.\boldsymbol{v}^{e\to t}$ is not $\varnothing$-protected.

Another example is

$$(\boldsymbol{u}^{t\to t}\top \wedge \boldsymbol{u}^{t\to t}\bot) \to \boldsymbol{u}^{t\to t}(\neg \boldsymbol{v}^{t}),$$

where $\top, \bot, \wedge, \to, \neg$ have the usual interpretation. This formula is extensionally valid, but not intensionally so. Observe that it is not $\{\boldsymbol{u}^{t\to t}, \boldsymbol{v}^{t}\}$-safe because $\mathrm{FV}(\neg \boldsymbol{v}^{t}) \cap \{\boldsymbol{u}^{t\to t}, \boldsymbol{v}^{t}\} \neq \varnothing$.

Note that $\mathrm{FV}(\varphi)$-safety is by no means a necessary condition for φ to have the property in Theorem 11. For one thing, φ may be an instance of an $\mathrm{FV}(\psi)$-safe formula ψ while not itself $\mathrm{FV}(\varphi)$-safe.

5 Montague's Typing in PTQ

The mapping

$$\alpha \mapsto \overline{\alpha},$$

which replaces each occurrence of e and t by $s \to e$ and $s \to t$, and the associated intensionalization and extensionalization combinators (int_α and ext_α) are not familiar to linguists. In linguistics, a common practice nowadays is to use the fewest instances of s that are necessary for adequate semantic analysis, rather than systematically replacing each occurrence of an atomic type by its intensional counterpart.

In Montague's original work, however, there was a systematic placement of s in the semantic types associated with syntactic categories. In PTQ (Montague 1973), syntactic categories are built from basic categories e and t by means of two connectives / and //. The semantic type $f(A)$ associated with a syntactic category A was defined by the following recursion:

$$\begin{aligned} f(e) &= e,\\ f(t) &= t,\\ f(A/B) = f(A//B) &= (s \to f(B)) \to f(A). \end{aligned}$$

This gives rise to the following association between an extensional semantic type α and its intensional counterpart $h(\alpha)$:

$$\begin{aligned} h(e) &= e,\\ h(t) &= t,\\ h(\alpha \to \beta) &= (s \to h(\alpha)) \to h(\beta). \end{aligned}$$

This mapping $\alpha \mapsto h(\alpha)$ looks quite different from the above mapping $\alpha \mapsto \overline{\alpha}$. For example, if $\alpha = (e \to t) \to t$, then we have

$$\begin{aligned} \overline{\alpha} &= ((s \to e) \to s \to t) \to s \to t, \\ h(\alpha) &= (s \to ((s \to e) \to t)) \to t. \end{aligned}$$

Note that the number of occurrences of s in the two types is different: it is three for $\overline{\alpha}$ and two for $h(\alpha)$.

Nevertheless, there is a systematic correspondence between the two approaches. First, note that $f(A)$ is the type of the *extension* of an expression of syntactic category A. The type of the *intension* of an expression of syntactic category A is $s \to f(A)$. Thus, what we should really be comparing to $\overline{\alpha}$ is not $h(\alpha)$, but $s \to h(\alpha)$. It is easy to see that the number of occurrences of s in $\overline{\alpha}$ and in $s \to h(\alpha)$ is the same for all α. Indeed, we can go from one type to the other by repeatedly applying the operation of changing the order of arguments:

$$\beta \to \gamma \to \delta \quad \rightsquigarrow \quad \gamma \to \beta \to \delta \tag{10}$$

For example, with $\alpha = (e \to t) \to t$,

$$\begin{aligned} \overline{\alpha} &= ((s \to e) \to s \to t) \to s \to t \\ &\rightsquigarrow (s \to (s \to e) \to t) \to s \to t \\ &\rightsquigarrow s \to (s \to (s \to e) \to t) \to t \\ &= s \to h(\alpha). \end{aligned}$$

In general,

$$\begin{aligned} \overline{a} &= s \to h(a), \\ \overline{\alpha \to \beta} &= \overline{\alpha} \to \overline{\beta} \\ &\rightsquigarrow^{*} (s \to h(\alpha)) \to s \to h(\beta) \\ &\rightsquigarrow s \to (s \to h(\alpha)) \to h(\beta) \\ &= s \to h(\alpha \to \beta). \end{aligned}$$

The domains of the two types in (10) are of course related by the combinator

$$C_{\beta,\gamma,\delta} = \lambda x^{\beta \to \gamma \to \delta} y^{\beta} z^{\gamma}.xzy$$

and its inverse, $C_{\gamma,\beta,\delta}$, which shows that the two types are *isomorphic* (di Cosmo 2005). It easily follows that $\overline{\alpha}$ and $s \to h(\alpha)$ must also be isomorphic; indeed, this is witnessed by the pair of combinators P_α and Q_α defined as follows:

$$\begin{aligned} P_a &= \lambda x^{s \to a}.x, & Q_a &= \lambda y^{s \to a}.y, \\ P_{\alpha \to \beta} &= \lambda x^{\overline{\alpha \to \beta}} i^s y^{s \to h(\alpha)}.P_\beta(x(Q_\alpha y))i, & Q_{\alpha \to \beta} &= \lambda y^{s \to h(\alpha \to \beta)} x^{\overline{\alpha}}.Q_\beta(\lambda i^s.yi(P_\alpha x)). \end{aligned}$$

We have

$$\lambda x^{\overline{\alpha}}.Q_\alpha(P_\alpha x) = \lambda x^{\overline{\alpha}}.x, \quad \lambda y^{s \to h(\alpha)}.P_\alpha(Q_\alpha y) = \lambda y^{s \to h(\alpha)}.y.$$

This allows us to define the PTQ version $\mathrm{int}_\alpha^{\mathrm{PTQ}}$ and $\mathrm{ext}_\alpha^{\mathrm{PTQ}}$ of intensionalization and extensionalization combinators in terms of int_α and ext_α:

$$\begin{aligned}\mathrm{int}_\alpha^{\mathrm{PTQ}} &= \lambda x^{s\to\alpha}.P_\alpha(\mathrm{int}_\alpha\, x),\\ \mathrm{ext}_\alpha^{\mathrm{PTQ}} &= \lambda y^{s\to h(\alpha)}.\, \mathrm{ext}_\alpha(Q_\alpha y).\end{aligned}$$

A direct recursive definition of $\mathrm{int}_\alpha^{\mathrm{PTQ}}$ and $\mathrm{ext}_\alpha^{\mathrm{PTQ}}$ works out as follows:

$$\begin{aligned}\mathrm{int}_a^{\mathrm{PTQ}} &= \lambda x^{s\to a}.x,\\ \mathrm{int}_{\alpha\to\beta}^{\mathrm{PTQ}} &= \lambda x^{s\to\alpha\to\beta} i^s y^{s\to h(\alpha)}.\, \mathrm{int}_\beta^{\mathrm{PTQ}}(\lambda j^s.xj(\mathrm{ext}_\alpha^{\mathrm{PTQ}}\, y\ j))i,\\ \mathrm{ext}_a^{\mathrm{PTQ}} &= \lambda y^{s\to a}.y,\\ \mathrm{ext}_{\alpha\to\beta}^{\mathrm{PTQ}} &= \lambda y^{s\to(s\to h(\alpha))\to h(\beta)} j^s x^\alpha.\, \mathrm{ext}_\beta^{\mathrm{PTQ}}(\lambda i^s.yi(\mathrm{int}_\alpha^{\mathrm{PTQ}}(\lambda k^s.x)))j.\end{aligned}$$

Note that if $\alpha = \alpha_1 \to \cdots \to \alpha_n \to a$, then

$$\begin{aligned}\mathrm{int}_\alpha^{\mathrm{PTQ}} &= \lambda x^{s\to\alpha} i^s y_1^{s\to h(\alpha_1)} \ldots y_n^{s\to h(\alpha_n)}.xi(\mathrm{ext}_{\alpha_1}^{\mathrm{PTQ}}\, y_1\ i)\ldots(\mathrm{ext}_{\alpha_n}^{\mathrm{PTQ}}\, y_n\ i),\\ \mathrm{ext}_\alpha^{\mathrm{PTQ}} &= \lambda y^{s\to h(\alpha)} j^s x_1^{\alpha_1} \ldots x_n^{\alpha_n}.yj\,(\mathrm{int}_{\alpha_1}^{\mathrm{PTQ}}(\lambda k^s.x_1))\ldots(\mathrm{int}_{\alpha_n}^{\mathrm{PTQ}}(\lambda k^s.x_n)).\end{aligned} \tag{11}$$

For example, if $\mathbf{j} \in D_e$, then

$$\mathrm{int}_{(e\to t)\to t}^{\mathrm{PTQ}}(\lambda k^s u^{e\to t}.u\mathbf{j}) = \lambda i^s y^{s\to(s\to e)\to t}.yi(\lambda k^s.\mathbf{j}).$$

This is the simply typed λ-calculus expression corresponding to PTQ's translation of **John**.

A *PTQ model M* of our object language consists of base domains $(D_a)_{s\in A\cup\{s\}}$ together with an assignment of an intension $M(c) \in D_{s\to h(\alpha)}$ to each object language constant c of type α. An object language expression φ has the intension $[\![\varphi]\!]_{M,g}^{\mathrm{PTQ}}$ in a PTQ model M relative to an assignment g of values to variables, where $g(\boldsymbol{v}_l^\alpha) \in D_{s\to h(\alpha)}$ for every variable $\boldsymbol{v}_l^\alpha$ of type α:

$$\begin{aligned}[\![c]\!]_{M,g}^{\mathrm{PTQ}} &= M(c),\\ [\![\boldsymbol{v}_l^\alpha]\!]_{M,g}^{\mathrm{PTQ}} &= g(\boldsymbol{v}_l^\alpha),\\ [\![\varphi\psi]\!]_{M,g}^{\mathrm{PTQ}} &= \lambda i^s.[\![\varphi]\!]_{M,g}^{\mathrm{PTQ}} i[\![\psi]\!]_{M,g}^{\mathrm{PTQ}},\\ [\![\boldsymbol{\lambda v}_l^\alpha.\varphi]\!]_{M,g}^{\mathrm{PTQ}} &= \lambda i^s x^{s\to h(\alpha)}.[\![\varphi]\!]_{M,g[x/\boldsymbol{v}_l^\alpha]}^{\mathrm{PTQ}} i\end{aligned} \tag{12}$$

Note that these are recursive clauses for the *intensions* of object language expressions. In the case of the PTQ fragment, our object language expressions roughly correspond to meaning recipes associated with analysis trees of English expressions.[8]

The last two clauses of the above recursive definition can be recast in terms of *extensions* (not to be confused with extensionalization), i.e., values of intensions at particular indices, as follows. Writing $[\![\varphi]\!]_{M,g,i}^{\mathrm{PTQ}}$ for $[\![\varphi]\!]_{M,g}^{\mathrm{PTQ}} i$, we have

$$\begin{aligned}[\![\varphi\psi]\!]_{M,g,i}^{\mathrm{PTQ}} &= [\![\varphi]\!]_{M,g,i}^{\mathrm{PTQ}}(\lambda i^s.[\![\psi]\!]_{M,g,i}^{\mathrm{PTQ}}),\\ [\![\boldsymbol{\lambda v}_l^\alpha.\varphi]\!]_{M,g,i}^{\mathrm{PTQ}} &= \lambda x^{s\to h(\alpha)}.[\![\varphi]\!]_{M,g[x/\boldsymbol{v}_l^\alpha],i}^{\mathrm{PTQ}}.\end{aligned}$$

[8] The correspondence is not exact, however, for reasons we choose not to go into here.

The former says that the extension of $\varphi\psi$ is the extension of φ applied to the *intension* of ψ. This semantic recipe was called "intensional functional application" by Heim and Kratzer (1998), and it appears in PTQ in the form of the Intensional Logic (IL) expression $\varphi'({}^{\wedge}\psi')$, where φ' and ψ' translate φ and ψ, respectively.

The two intensional interpretations of φ are related by the following equations. For any object language expression φ of type γ, we have

$$\begin{aligned}
[\![\varphi]\!]^{\mathrm{PTQ}}_{M,g} &= P_\gamma([\![\varphi]\!]_{Q\circ M, Q\circ g}) && \text{for PTQ model } M \text{ and PTQ assignment } g,\\
[\![\varphi]\!]_{M,g} &= Q_\gamma([\![\varphi]\!]^{\mathrm{PTQ}}_{P\circ M, P\circ g}) && \text{for intensional model } M \text{ and assignment } g.
\end{aligned}$$

Here, $P \circ M$ is the PTQ model such that $(P \circ M)(c) = P_\alpha(M(c))$ for each constant c of type α, and $P \circ g$ is the PTQ assignment such that $(P \circ g)(\boldsymbol{v}^\alpha_l) = P_\alpha(g(\boldsymbol{v}^\alpha_l))$ for each variable $\boldsymbol{v}^\alpha_l$. The definitions of $Q \circ M$ and $Q \circ g$ are similar. The above equations can be proved by straightforward induction on φ. In particular, when φ is a sentence (i.e., closed object language expression of type t), we have

$$[\![\varphi]\!]^{\mathrm{PTQ}}_{M} = [\![\varphi]\!]_{Q\circ M}$$

for every PTQ model M.

As before, given an indexed collection $\mathcal{I} = (M_i)_{i\in I}$ of extensional models with the same base domains $(D_a)_{a\in A}$, we can create a PTQ model $M^{\mathrm{PTQ}}_{\mathcal{I}}$ based on $\mathcal{I}$, with $D_s = I$, by letting

$$M^{\mathrm{PTQ}}_{\mathcal{I}}(c) = \mathrm{int}^{\mathrm{PTQ}}_\alpha(\lambda i^s.M_i(c))$$

for each object language constant c of type α. Then $Q \circ M^{\mathrm{PTQ}}_{\mathcal{I}} = M_{\mathcal{I}}$ and we can easily prove analogues of Lemma 3 and Theorem 6. In particular, for every sentence φ, we have

$$[\![\varphi]\!]^{\mathrm{PTQ}}_{M^{\mathrm{PTQ}}_{\mathcal{I}}} = \lambda i^s.[\![\varphi]\!]_{M_i} = [\![\varphi]\!]_{M_{\mathcal{I}}}.$$

Finally, it may be of some interest to note that $\mathrm{int}^{\mathrm{PTQ}}_\alpha$ and $\mathrm{ext}^{\mathrm{PTQ}}_\alpha$ are definable in Montague's (1973) IL. For $\alpha = \alpha_1 \to \cdots \to \alpha_n \to a$, let

$$\begin{aligned}
\mathrm{int}^{\mathrm{IL}}_\alpha[x^{s\to\alpha}] &= \lambda y_1^{s\to h(\alpha_1)} \ldots y_n^{s\to h(\alpha_n)}.{}^{\vee}x(\mathrm{ext}^{\mathrm{IL}}_{\alpha_1}[y_1]) \ldots (\mathrm{ext}^{\mathrm{IL}}_{\alpha_n}[y_n]),\\
\mathrm{ext}^{\mathrm{IL}}_\alpha[y^{s\to h(\alpha)}] &= \lambda x_1^{\alpha_1} \ldots x_n^{\alpha_n}.{}^{\vee}y({}^{\wedge}(\mathrm{int}^{\mathrm{IL}}_{\alpha_1}[{}^{\wedge}x_1])) \ldots ({}^{\wedge}(\mathrm{int}^{\mathrm{IL}}_{\alpha_n}[{}^{\wedge}x_n])),
\end{aligned}$$

where the right-hand sides of the equations are IL expressions and equality is syntactic equality. Thus, $\mathrm{int}^{\mathrm{IL}}_\alpha[x^{s\to\alpha}]$ is an IL expression of type $h(\alpha)$ whose only free variable is x, and $\mathrm{ext}^{\mathrm{IL}}_\alpha[y^{s\to h(\alpha)}]$ is an IL expression of type α whose only free variable is y. (We write $\mathrm{int}^{\mathrm{IL}}_\alpha[\varphi]$, where φ is an IL expression of type $s \to \alpha$, for the result of replacing $x^{s\to\alpha}$ by φ in $\mathrm{int}^{\mathrm{IL}}_\alpha[x^{s\to\alpha}]$.) When these IL expressions are translated into simply typed λ-terms (of type $s \to h(\alpha)$ and $s \to \alpha$, respectively), they come out as equivalent to the following, obtained from (11):

$$\begin{aligned}
\mathrm{int}^{\mathrm{PTQ}}_\alpha\, x^{s\to\alpha} &= \lambda i^s y_1^{s\to h(\alpha_1)} \ldots y_n^{s\to h(\alpha_n)}.xi(\mathrm{ext}^{\mathrm{PTQ}}_{\alpha_1}\, y_1\; i) \ldots (\mathrm{ext}^{\mathrm{PTQ}}_{\alpha_n}\, y_n\; i),\\
\mathrm{ext}^{\mathrm{PTQ}}_\alpha\, y^{s\to h(\alpha)} &= \lambda j^s x_1^{\alpha_1} \ldots x_n^{\alpha_n}.yj\,(\mathrm{int}^{\mathrm{PTQ}}_{\alpha_1}(\lambda k^s.x_1)) \ldots (\mathrm{int}^{\mathrm{PTQ}}_{\alpha_n}(\lambda k^s.x_n)).
\end{aligned}$$

The translation in question is

$$\begin{aligned}
x^{\alpha\dagger} &= \lambda k^s.x^\alpha, \\
(\varphi\psi)^\dagger &= \lambda i^s.\varphi^\dagger i(\psi^\dagger i), \\
(\lambda x^\alpha.\varphi)^\dagger &= \lambda i^s x^\alpha.\varphi^\dagger i, \\
({}^\wedge\varphi)^\dagger &= \lambda k^s.\varphi^\dagger, \\
({}^\vee\varphi)^\dagger &= \lambda i^s.\varphi^\dagger ii,
\end{aligned}$$

which is a straightforward rendering of the semantics of IL given in Montague (1973).[9] Note that this translation gives a simply typed λ-term that represents the *intension* of an IL expression, rather than its extension at a particular index, as in Gallin (1975). We have

$$\begin{aligned}
(\mathrm{int}_\alpha^{\mathrm{IL}}[x^{s\to\alpha}])^\dagger &= \mathrm{int}_\alpha^{\mathrm{PTQ}}\, x^{s\to\alpha}, \\
(\mathrm{ext}_\alpha^{\mathrm{IL}}[y^{s\to h(\alpha)}])^\dagger &= \mathrm{ext}_\alpha^{\mathrm{PTQ}}\, y^{s\to h(\alpha)},
\end{aligned}$$

which can be proved by straightforward induction. For example, when c is a constant of IL of an extensional type α, the PTQ-style intensionalization of its intension (which is an object of type $s \to \alpha$) is given by (the intension of) the IL expression $\mathrm{int}_\alpha^{\mathrm{IL}}[{}^\wedge c]$.

Acknowledgements We are indebted to Reinhard Muskens and Yoad Winter for helpful discussions.

References

Ben-Avi G, Winter Y (2007) The semantics of intensionalization. In: Muskens R (ed) Workshop on New Directions in Type-Theoretic Grammars, pp 98–112

Blackburn P, de Rijke M, Venema Y (2001) Modal Logic. Cambridge University Press, Cambridge

di Cosmo R (2005) A short survey of isomorphisms of types. Mathematical Structures in Computer Science 15:825–838

van Eijck J, Unger C (2011) Computational Semantics with Functional Programming. Cambridge University Press, Cambridge

von Fintel K, Heim I (2011) Intensional semantics. URL http://mit.edu/fintel/fintel-heim-intensional.pdf, online lecture notes

Gallin D (1975) Intensional and Higher-Order Modal Logic: With Applications to Montague Semantics. North-Holland, Amsterdam

Heim I, Kratzer A (1998) Semantics in Generative Grammar. Blackwell, Oxford

Montague R (1973) The proper treatment of quantification in ordinary English. In: Hintikka J, Moravcsik J, Suppes P (eds) Approaches to Natural Language: Proceedings of the 1970 Stanford Workshop on Grammar and Semantics, D. Reidel, Dordrecht, pp 221–242

Partee B, Rooth M (1983) Generalized conjunction and type ambiguity. In: Bäuerle R, Schwarze C, von Stechow A (eds) Meaning, Use, and Interpretation of Language, de Gruyter, Berlin, pp 361–383

[9] This translation of IL expressions (without constants) into simply typed λ-calculus must not be confused with (12), which gives the PTQ-style intensional compositional semantics to expressions of our object language, which are simply typed λ-terms, not IL expressions.