On Logical Relations and Conservativity

Philippe de Groote

Inria Nancy - Grand Est
France
Philippe.deGroote@inria.fr

1 Introduction

Having developed a Montague-like interpretation of a fragment of natural language, one may want to modify it in order to accomodate further linguistic phenomena. This may require radical changes in the semantic interpretation of syntactic categories. For instance, one may want to raise the semantic type assigned to the category of adverbial modifiers in order to account for quantified adverbial modifiers, i.e., to interpret adverbial modifiers as terms of type $(((\mathbf{e} \to \mathbf{t}) \to \mathbf{e} \to \mathbf{t}) \to \mathbf{t}) \to \mathbf{t}$ rather than $(\mathbf{e} \to \mathbf{t}) \to \mathbf{e} \to \mathbf{t}$. Such a modification must then be propagated throughout the current interpretation, which might be cumbersome if it cannot be done in a systematic way. Ideally, one would like to propagate the change by applying some automatic procedure that would guarantee that the new obtained interpretation is faithful to the original one.

The intensionalization procedure introduced in [5] is an example of such a systematic procedure. It allows the extensional interpretation of a language to be transformed into an intensionalized interpretation that offers room for accommodating truly intensional phenomena. Moreover, this procedure is conservative in the sense that it preserves the truth conditions of sentences. Another instance of such a systematic procedure is provided by the dynamization procedure described in [6], which allows a static interpretation to be turned into a dynamic one capable of accommodating phenomena related to discourse dynamics. This procedure, which is also conservative, presents several similarities with the intentionalization procedure. This raises the following question: is there some abstract general scheme of which both the intensionalization and dynamization procedures would be instances?

This paper reports some ongoing work along this research direction. It attempts to develop an abstract construct that generalizes the intensionalization and dynamization procedures, and for which conservativity results may be established using the notion of logical relation.

2 Logical Relation

This section presents the main technical tool that we use in this paper, namely, the notion of logical relation.

We assume from the reader some familiarity with the simply typed λ-calculus. Given a set A of atomic types, we write $\mathscr{T}(A)$ for the set of simple types built upon A. We will be concerned with simply typed λ-calculi built upon different sets of typed constants. Such "sets" of typed constants are called higher-order signatures, and are formally defined as follows.

Definition 1. *A* higher-order signature *consists of a triple $\Sigma = \langle A, C, \tau \rangle$, where:*

(i) *A is a finite set of atomic types;*

(ii) *C is a finite set of constants;*

(iii) *$\tau : C \to \mathscr{T}(A)$ is a function that assigns to each constant in C an implicative type in $\mathscr{T}(A)$.*

Given a higher-order signature $\Sigma = \langle A, C, \tau \rangle$, we write $\Lambda(\Sigma)$ for the set of simply typed λ-terms built upon Σ. Given a type $\alpha \in \mathscr{T}(A)$, we write $\Lambda^{\alpha}(\Sigma)$ for the set of simply typed λ-terms of type α built upon Σ.

Given two sets of atomic types A and B, a map $h : \mathscr{T}(A) \to \mathscr{T}(B)$ is called a type homomorphism when it satisfies the following condition:

$$h(\alpha \to \beta) = h(\alpha) \to h(\beta)$$

Similarly, given two signatures Σ_1 and Σ_2, a map $h : \Lambda(\Sigma_1) \to \Lambda(\Sigma_2)$ is called a term homomorphism when it satisfies the following conditions:

$$h(x) = x \qquad h(\lambda x.\, t) = \lambda x.\, h(t) \qquad h(t\, u) = h(t)\, (h(u))$$

The notions of type and term homomorphisms allow us to define the notion of a morphism between two signatures.

Definition 2. *Let $\Sigma_1 = \langle A_1, C_1, \tau_1 \rangle$ and $\Sigma_2 = \langle A_2, C_2, \tau_2 \rangle$ be two higher-order signatures. A morphism $H : \Sigma_1 \to \Sigma_2$ is defined to be a pair of maps $\langle h_1, h_2 \rangle$ such that:*

(i) *$h_1 : \mathscr{T}(A_1) \to \mathscr{T}(A_2)$ is a type homomorphism;*

(ii) *$h_2 : \Lambda(\Sigma_1) \to \Lambda(\Sigma_2)$ is a term homomorphism;*

(iii) *For every $\alpha \in \mathscr{T}(A_1)$, and every $t \in \Lambda^{\alpha}(\Sigma_1)$, we have that $h_2(t) \in \Lambda^{h_1(\alpha)}(\Sigma_2)$.*

For $H = \langle h_1, h_2 \rangle$ such a morphism, we will write $H(\alpha)$ and $H(t)$ for $h_1(\alpha)$ and $h_2(t)$, respectively.

Logical relations may be defined between λ-terms or between denotations of λ-terms. This corresponds to the distinction between the notions of syntactic and semantic logical relations [1]. In this paper, we adopt an intermediate standpoint: we define at the syntactic level a logical relation that has the flavor of a semantic logical relation. To this end, we introduce the notion of span.

Definition 3. *A span is a tuple $\langle \Sigma_0, \Sigma_1, \Sigma_2, F, G \rangle$ where*

(i) *Σ_0, Σ_1, and Σ_2 are higher-order signatures;*

(ii) *$F : \Sigma_0 \to \Sigma_1$ is a morphism between Σ_0 and Σ_1;*

(iii) *$G : \Sigma_0 \to \Sigma_2$ is a morphism between Σ_0 and Σ_2.*

$$\begin{tikzcd} & \Sigma_0 \arrow[dl, "F"'] \arrow[dr, "G"] & \\ \Sigma_1 & & \Sigma_2 \end{tikzcd}$$

In the sequel of this paper, we will let Σ_0, Σ_1, and Σ_2 range over signatures, and we adopt the convention that A_i, C_i, and τ_i stand for the three respective components of Σ_i.

We are now in a position of giving our definition of a logical relation.

Definition 4. *Given a span $\langle \Sigma_0, \Sigma_1, \Sigma_2, (\ )^*, \overline{(\ )} \rangle$, a* logical relation *is defined to be a family of binary relations, $R = \{R_\alpha\}_{\alpha \in \mathscr{T}(A_0)}$, such that:*

(i) $R_\alpha \subset \Lambda^{\alpha^*}(\Sigma_1) \times \Lambda^{\overline{\alpha}}(\Sigma_2)$*, for* $\alpha \in \mathscr{T}(A_0)$*;*

(ii) $t_1 \, R_{\alpha\to\beta} \, t_2$ *iff* $\forall u_1 \in \Lambda^{\alpha^*}(\Sigma_1). \forall u_2 \in \Lambda^{\overline{\alpha}}(\Sigma_2).\, u_1 \, R_\alpha \, u_2 \Rightarrow (t_1\, u_1) \, R_\beta \, (t_2\, u_2)$

We will only be concerned with logical relations that are $\beta\eta$-closed, i.e., logical relations that obey the following definition.

Definition 5. *A logical relation R defined on a span $\langle \Sigma_0, \Sigma_1, \Sigma_2, (\ )^*, \overline{(\ )} \rangle$ is said to be $\beta\eta$-closed if for every $a \in A_0$, every $t_1, u_1 \in \Lambda^{a^*}(\Sigma_1)$ such that $t_1 =_{\beta\eta} u_1$, and every $t_2, u_2 \in \Lambda^{\overline{a}}(\Sigma_2)$ such that $t_2 =_{\beta\eta} u_2$, if $t_1 \, R_a \, t_2$ then $u_1 \, R_a \, u_2$.*

In the above definition, the very property of $\beta\eta$-closure is only required at atomic type. The next lemma shows that this is indeed sufficient to ensure $\beta\eta$-closure at every type.

Lemma 1. *Let R be a logical relation defined on a span $\langle \Sigma_0, \Sigma_1, \Sigma_2, (\ )^*, \overline{(\ )} \rangle$. If R is $\beta\eta$-closed then for every $\alpha \in \mathscr{T}(A_0)$, every $t_1, u_1 \in \Lambda^{\alpha^*}(\Sigma_1)$ such that $t_1 =_{\beta\eta} u_1$, and every $t_2, u_2 \in \Lambda^{\overline{\alpha}}(\Sigma_2)$ such that $t_2 =_{\beta\eta} u_2$, if $t_1 \, R_\alpha \, t_2$ then $u_1 \, R_\alpha \, u_2$.*

Proof. The proof proceeds by structural induction on α. If α is atomic, the property holds by definition. Now, let $\alpha \equiv \beta \to \gamma$. For every $v_1 \in \Lambda^{\beta^*}(\Sigma_1)$ and every $v_2 \in \Lambda^{\overline{\beta}}(\Sigma_2)$ such that $v_1 \, R_\beta \, v_2$, one has $(t_1\, v_1) \, R_\gamma \, (t_2\, v_2)$. Then, by induction hypothesis, $(u_1\, v_1) \, R_\gamma \, (u_2\, v_2)$, which establishes $u_1 \, R_{\beta\to\gamma} \, u_2$. □

We now state and prove a version of the so-called fundamental theorem of logical relations.

Proposition 1. *Let $\langle \Sigma_0, \Sigma_1, \Sigma_2, (\ )^*, \overline{(\ )} \rangle$ be a span, and let R be a $\beta\eta$-closed logical relation defined on this span such that $c^* \, R_{\tau_0(c)} \, \overline{c}$, for every $c \in C_0$. Let $t \in \Lambda^\alpha(\Sigma_0)$ be a λ-term whose free variables are among $x_1, \dots, x_n$ of types $\alpha_1, \dots, \alpha_n$, respectively. Let $u_{11} \in \Lambda^{\alpha_1^*}(\Sigma_1), \dots, u_{1n} \in \Lambda^{\alpha_n^*}(\Sigma_1)$, $u_{21} \in \Lambda^{\overline{\alpha_1}}(\Sigma_2), \dots, u_{2n} \in \Lambda^{\overline{\alpha_n}}(\Sigma_2)$ be such that $u_{11} \, R_{\alpha_1} \, u_{21}$, $\dots$, $u_{1n} \, R_{\alpha_n} \, u_{2n}$. Then,*

$$t^*[x_1 := u_{11}, \dots, x_n := u_{1n}] \, R_\alpha \, \overline{t}[x_1 := u_{21}, \dots, x_n := u_{2n}].$$

Proof. The proof proceeds by structural induction on t.

Let $t \equiv c$ be a constant.

$c^* \, R_{\tau_0(c)} \, \overline{c}$ holds by hypothesis.

Let $t \equiv x_i$ be a variable.

We have:

$$\begin{array}{ll}
t^*[x_1 := u_{11}, \dots, x_n := u_{1n}] & \overline{t}[x_1 := u_{21}, \dots, x_n := u_{2n}] \\
\quad = x_i^*[x_1 := u_{11}, \dots, x_n := u_{1n}] & \quad = \overline{x_i}[x_1 := u_{21}, \dots, x_n := u_{2n}] \\
\quad = x_i[x_1 := u_{11}, \dots, x_n := u_{1n}] & \quad = x_i[x_1 := u_{21}, \dots, x_n := u_{2n}] \\
\quad = u_{1i} & \quad = u_{2i}
\end{array}$$

Then, $u_{1i} \, R_{\alpha_i} \, u_{2i}$ by hypothesis.

Let $t \equiv \lambda x.\, t_1$ be a λ-abstraction.

In this case, there exist β and γ such that $\alpha \equiv \beta \to \gamma$. Then, by induction hypothesis, for every $u_1 \in \Lambda^{\beta^*}(\Sigma_1)$ and every $u_2 \in \Lambda^{\overline{\beta}}(\Sigma_2)$ such that $u_1 \, R_\beta \, u_2$, we have:

$$t_1^*[x_1 := u_{11}, \dots, x_n := u_{1n}, x := u_1] \, R_\gamma \, \overline{t_1}[x_1 := u_{21}, \dots, x_n := u_{2n}, x := u_2].$$

Hence, because R is $\beta\eta$-closed:

$$(\lambda x.\, t_1^*[x_1 := u_{11}, \dots, x_n := u_{1n}]) \, u_1 \, R_\gamma \, (\lambda x.\, \overline{t_1}[x_1 := u_{21}, \dots, x_n := u_{2n}]) \, u_2.$$

Consequently,

$$(\lambda x.\, t_1^*)[x_1 := u_{11}, \dots, x_n := u_{1n}] \, u_1 \, R_\gamma \, (\lambda x.\, \overline{t_1})[x_1 := u_{21}, \dots, x_n := u_{2n}] \, u_2,$$

which establishes:

$$\lambda x.\, t_1^*[x_1 := u_{11}, \dots, x_n := u_{1n}] \, R_{\beta\to\gamma} \, \lambda x.\, \overline{t_1}[x_1 := u_{21}, \dots, x_n := u_{2n}].$$

Let $t \equiv t_1 \, t_2$ be an application.

In this last case, there exists β such that t_1 is of type $\beta \to \alpha$ and t_2 is of type β. Then, by induction hypothesis:

$$t_1^*[x_1 := u_{11}, \dots, x_n := u_{1n}] \, R_{\beta\to\alpha} \, \overline{t_1}[x_1 := u_{21}, \dots, x_n := u_{2n}]$$

and

$$t_2^*[x_1 := u_{11}, \dots, x_n := u_{1n}] \, R_\beta \, \overline{t_2}[x_1 := u_{21}, \dots, x_n := u_{2n}].$$

Hence, because R is logical:

$$(t_1 \, t_2)^*[x_1 := u_{11}, \dots, x_n := u_{1n}] \, R_\alpha \, \overline{(t_1 \, t_2)}[x_1 := u_{21}, \dots, x_n := u_{2n}].$$

□

In the case of closed λ-terms, we obtain the following corollary.

Corollary 1. *Let $\langle \Sigma_0, \Sigma_1, \Sigma_2, (\ )^*, \overline{(\ )} \rangle$ be a span, and let R be a $\beta\eta$-closed logical relation defined on this span such that $c^* \, R_{\tau_0(c)} \, \overline{c}$, for every $c \in C_0$. Let $t \in \Lambda^\alpha(\Sigma_0)$ be a closed λ-term. Then,*

$$t^* \, R_\alpha \, \overline{t}.$$

□

3 Embedding and Projection

Throughout this section, we assume the existence of a span $\langle \Sigma_0, \Sigma_1, \Sigma_2, (\ )^*, \overline{(\ )} \rangle$. One may think of it as modelling the following situation:

- a λ-term $t \in \Lambda(\Sigma_0)$ corresponds to the logical form of some natural language utterance;
- $t^* \in \Lambda(\Sigma_1)$ corresponds to the current (semantic) interpretation of t;
- $\overline{t} \in \Lambda(\Sigma_2)$ corresponds to the new intended interpretation of t.

With this picture in mind, the kind of conservativity properties we are seeking must ensure that the new interpretation $\overline{\phi}$ of a sentence ϕ is valid if and only if its current interpretation ϕ^* is.

Our goal is to establish such conservativity properties by using logical relations. Corollary 1 ensures that the two interpretations of a sentence (i.e., ϕ^* and $\overline{\phi}$) will be logically related provided that the interpretations of the constants of Σ_0 are logically related. In order to ensure this latter property, we construct a procedure that allows a term of type α^* to be transformed into a term of type $\overline{\alpha}$ in such a way that the original term and its transform are logically related. To this end, we define a family of embedding functions, $\{\mathbb{E}_\alpha\}_{\alpha\in\mathscr{T}(A_0)}$, and a family of projection functions, $\{\mathbb{P}_\alpha\}_{\alpha\in\mathscr{T}(A_0)}$. These functions are typable with the following types:

$$\begin{aligned}\mathbb{E}_\alpha &: \mathrm{T}\alpha^* \to \overline{\alpha}\\ \mathbb{P}_\alpha &: \overline{\alpha} \to \mathrm{T}\alpha^*\end{aligned}$$

where T is a type transformer coming with three operations: U, $\bullet$ (written as an infix operator), and C. These operations are assigned the following type schemes:

$$\begin{aligned}\mathsf{U} &: \alpha \to \mathrm{T}\alpha\\ \bullet &: \mathrm{T}(\alpha\to\beta) \to \mathrm{T}\alpha \to \mathrm{T}\beta\\ \mathsf{C} &: (\alpha \to \mathrm{T}\beta) \to \mathrm{T}(\alpha\to\beta)\end{aligned}$$

In addition, they obey the following laws:

$$(\mathsf{U}\,f) \bullet (\mathsf{U}\,a) = \mathsf{U}\,(f\,a) \tag{C1}$$

$$\mathsf{C}\,(\lambda x.\,\mathsf{U}\,(f\,x)) = \mathsf{U}\,f \tag{C2}$$

The embedding and projection functions are defined by induction on the types in $\mathscr{T}(A_0)$. For every atomic type $a \in A_0$, we assume the existence of an embedding $\mathbb{E}_a$ and a projection $\mathbb{P}_a$ such that:

$$\mathbb{P}_a\,(\mathbb{E}_a\,(\mathsf{U}\,t)) = \mathsf{U}\,t \tag{C3}$$

Then, at functional type, the embedding and the projection functions are defined by means of the following equations:

$$\begin{aligned}\mathbb{E}_{\alpha\to\beta}\,t &= \lambda x^{\overline{\alpha}}.\,\mathbb{E}_\beta\,(t \bullet (\mathbb{P}_\alpha\,x))\\ \mathbb{P}_{\alpha\to\beta}\,t &= \mathsf{C}\,(\lambda x^{\alpha^*}.\,\mathbb{P}_\beta\,(t\,(\mathbb{E}_\alpha\,(\mathsf{U}\,x))))\end{aligned}$$

We may now define an operation, say T, that allows a term of type α^* to be transformed into a term of type $\overline{\alpha}$:

$$\mathsf{T}\,t = \mathbb{E}_\alpha\,(\mathsf{U}\,t)$$

As we said, we want t and $(\mathsf{T}\,t)$ to be somehow logically related. For this purpose, we define the following relation, which is logical by construction.

$$\begin{aligned}t_1 \prec_a t_2 &\text{ iff } t_2 =_{\beta\eta} \mathbb{E}_a\,(\mathsf{U}\,t_1),\ \text{for } a\in A_0\\ t_1 \prec_{\alpha\to\beta} t_2 &\text{ iff } \forall u_1 \in \Lambda^{\alpha^*}(\Sigma_1).\forall u_2 \in \Lambda^{\overline{\alpha}}(\Sigma_2).\,u_1 \prec_\alpha u_2 \Rightarrow (t_1\,u_1) \prec_\beta (t_2\,u_2)\end{aligned}$$

The next proposition is the main technical result of this paper. It will be used in the next section when establishing conservativity results.

Proposition 2. *Let $t_1 \in \Lambda^{\alpha^*}(\Sigma_1)$ and $t_2 \in \Lambda^{\overline{\alpha}}(\Sigma_2)$. The following properties hold:*

(i) *If $t_2 =_{\beta\eta} \mathbb{E}_\alpha\,(\mathsf{U}\,t_1)$ then $t_1 \prec_\alpha t_2$.*

(ii) *If $t_1 \prec_\alpha t_2$ then $\mathbb{P}_\alpha\,t_2 =_{\beta\eta} \mathsf{U}\,t_1$.*

Proof. The proof proceeds by structural induction on α.

Let α be atomic.

(i) We have that $t_1 \prec_\alpha t_2$ by definition of $\prec$.

(ii) By definition of $\prec$, we have that $t_2 =_{\beta\eta} \mathbb{E}_\alpha\,(\mathsf{U}\,t_1)$. Hence,

$$\begin{aligned} \mathbb{P}_\alpha\,t_2 &=_{\beta\eta} \mathbb{P}_\alpha\,(\mathbb{E}_\alpha\,(\mathsf{U}\,t_1)) \\ &=_{\beta\eta} \mathsf{U}\,t_1 && \text{(by C3)} \end{aligned}$$

Let $\alpha \equiv \beta \to \gamma$.

(i) Let $u_1 \prec_\beta u_2$.

$$\begin{aligned} t_2\,u_2 &= \mathbb{E}_{\beta\to\gamma}\,(\mathsf{U}\,t_1)\,u_2 \\ &= (\lambda x^{\overline{\beta}}.\,\mathbb{E}_\gamma\,((\mathsf{U}\,t_1) \bullet (\mathbb{P}_\beta\,x)))\,u_2 \\ &\to_\beta \mathbb{E}_\gamma\,((\mathsf{U}\,t_1) \bullet (\mathbb{P}_\beta\,u_2)) \\ &=_{\beta\eta} \mathbb{E}_\gamma\,((\mathsf{U}\,t_1) \bullet (\mathsf{U}\,u_1)) && \text{(by induction hypothesis ii)} \\ &=_{\beta\eta} \mathbb{E}_\gamma\,(\mathsf{U}\,(t_1\,u_1)) && \text{(by C1)} \end{aligned}$$

Hence, by induction hypothesis i, we obtain

$$(t_1\,u_1) \prec_\gamma (t_2\,u_2),$$

which establishes

$$t_1 \prec_\gamma t_2.$$

(ii) By induction hypothesis i, we have that $x \prec_\beta (\mathbb{E}_\beta\,(\mathsf{U}\,x))$. Consequently,

$$(t_1\,x) \prec_\gamma (t_2\,(\mathbb{E}_\beta\,(\mathsf{U}\,x))).$$

Then, we have:

$$\begin{aligned} \mathbb{P}_{\beta\to\gamma}\,t_2 &= \mathsf{C}\,(\lambda x^{\beta^*}.\,\mathbb{P}_\gamma\,(t_2\,(\mathbb{E}_\beta\,(\mathsf{U}\,x)))) \\ &=_{\beta\eta} \mathsf{C}\,(\lambda x^{\beta^*}.\,\mathsf{U}\,(t_1\,x)) && \text{(by induction hypothesis ii)} \\ &=_{\beta\eta} \mathsf{U}\,t_1 && \text{(by C2)} \end{aligned}$$

□

4 Applications

4.1 Intensionalization

As a first application, we show that the intensionalization procedure defined in [5] may be obtained as an instance of the general scheme we have developed in the previous section.

We consider as given a signature $\Sigma_0 = \langle A_0, C_0, \tau_0 \rangle$, where $A_0 = \{\mathbf{e}, \mathbf{t}\}$. This signature must be thought of as specifying an object language used for expressing the Montague semantics of some extensional fragment of natural language. We also consider as given a subset $C_r \subset C_0$. Intuitively, this subset corresponds to the rigid constants, that is the constants whose interpretations do not vary from one possible world to another.

We then construct two signatures, $\Sigma_1 = \langle A_1, C_1, \tau_1 \rangle$ and $\Sigma_2 = \langle A_2, C_2, \tau_2 \rangle$ as follows:

$$A_1 = \{\mathbf{e}, \mathbf{t}, \mathbf{s}\} \qquad\qquad A_2 = \{\mathbf{e}, \mathbf{t}, \mathbf{s}\}$$

$$C_1 = C_0 \cup \{\mathrm{i}\} \qquad\qquad C_2 = C_0$$

$$\tau_1(c) = \begin{cases} \mathbf{s} \text{ if } c = \mathrm{i} \\ \tau_0(c) \text{ if } c \in C_r \\ (\mathbf{s} \to \tau_0(c)) \text{ if } c \notin C_r \end{cases} \qquad\qquad \tau_2(c) = \begin{cases} \tau_0(c) \text{ if } c \in C_r \\ (\mathbf{s} \to \tau_0(c)) \text{ if } c \notin C_r \end{cases}$$

We define a first morphism, $(\ )^*$, from Σ_0 to Σ_1:

$$\begin{aligned} \mathbf{e}^* &= \mathbf{e} \\ \mathbf{t}^* &= \mathbf{t} \\ c^* &= \begin{cases} c \text{ if } c \in C_r \\ (c\,\mathrm{i}) \text{ if } c \notin C_r \end{cases} \end{aligned}$$

Finally, we define a second morphism, $\overline{(\ )}$, from Σ_0 to Σ_2:

$$\begin{aligned} \overline{\mathbf{e}} &= \mathbf{s} \to \mathbf{e} \\ \overline{\mathbf{t}} &= \mathbf{s} \to \mathbf{t} \\ \overline{c} &= \mathbb{E}_{\tau_0(c)}\,(\mathsf{U}\,c^*) \end{aligned}$$

where the primitives that are needed for defining the embedding and projection operators are defined as follows:

$$\begin{aligned} \mathrm{T}\alpha &= \mathbf{s} \to \alpha & \qquad \mathbb{E}_{\mathbf{e}}\,t &= t \\ \mathsf{U}\,t &= \lambda i.\, t[\mathrm{i} := i] & \qquad \mathbb{E}_{\mathbf{t}}\,t &= t \\ t \bullet u &= \lambda i.\, t\,i\,(u\,i) & \qquad \mathbb{P}_{\mathbf{e}}\,t &= t \\ \mathsf{C}\,t &= \lambda ix.\, t\,x\,i & \qquad \mathbb{P}_{\mathbf{t}}\,t &= t \end{aligned}$$

One can easily verify that Conditions (C1), (C2), and (C3) are satisfied. In the case of (C1), for instance, we have:

$$\begin{aligned} (\mathsf{U}\,t) \bullet (\mathsf{U}\,u) &= \lambda i.\, \mathsf{U}\,t\,i\,(\mathsf{U}\,u\,i) \\ &\to_\beta \lambda i.\, t[\mathrm{i} := i]\,(\mathsf{U}\,u\,i) \\ &\to_\beta \lambda i.\, t[\mathrm{i} := i]\,(u[\mathrm{i} := i]) \\ &= \lambda i.\, (t\,u)[\mathrm{i} := i] \\ &= \mathsf{U}\,(t\,u) \end{aligned}$$

Now, by Proposition 2, we have $c^* \prec_{\tau_0(c)} \overline{c}$ for every $c \in C_0$. Consequently, by Corollary 1, we have $t^* \prec_\alpha \overline{t}$ for every closed term $t \in \Lambda^\alpha(\Sigma_0)$. In particular, for every sentence ϕ (i.e., every closed term of type $\mathbf{t}$), we have that $\overline{\phi} =_{\beta\eta} \lambda i.\, \phi^*[\mathrm{i} := i]$. Consequently, $\overline{\phi}$ denotes $\boldsymbol{\lambda} a.\, 1$ if and only if ϕ^* denotes 1 for every possible valuation of i. This establishes the conservativity of the intensional interpretation over the extensional one.

4.2 Dynamization

A second instance of our general scheme is the dynamization procedure given in [6]. This procedure is based on the continuation-based approach to dynamics introduced in [4]. Accordingly, dynamic propositions are seen as terms of type $\mathbf{c} \to (\mathbf{c} \to \mathbf{t}) \to \mathbf{t}$, where $\mathbf{c}$ is the type of the discourse contexts, and $(\mathbf{c} \to \mathbf{t})$ is the type of the discourse continuations.

Here again, we consider as given a signature $\Sigma_0 = \langle A_0, C_0, \tau_0 \rangle$, with $A_0 = \{\mathbf{e}, \mathbf{t}\}$. We then take Σ_1 to be Σ_0, and $(\ )^*$ to be the identity.

As for Σ_2, it consists of Σ_0 enriched with a set C_d of dynamic primitives. Accordingly, we have:

$$\begin{aligned} A_2 &= \{\mathbf{e}, \mathbf{t}, \mathbf{c}\} \\ C_2 &= C_0 \cup C_d \\ \tau_2(c) &= \tau_0(c) \text{ for every } c \in C_0 \end{aligned}$$

We leave C_d unspecified. Nevertheless, we require the existence of a constant $\mathsf{nil} \in C_d$, such that $\tau_2(\mathsf{nil}) = \mathbf{c}$. Intuitively, nil stands for the empty context.

Finally, the morphism $\overline{(\ )}$ from Σ_0 to Σ_2 is defined as follows:

$$\begin{aligned} \overline{\mathbf{e}} &= \mathbf{e} \\ \overline{\mathbf{t}} &= \mathbf{c} \to (\mathbf{c} \to \mathbf{t}) \to \mathbf{t} \\ \overline{c} &= \mathbb{E}_{\tau_0(c)}\,(\mathsf{U}\,c^*) \end{aligned}$$

where the needed primitives are defined as follows:

$$\begin{aligned} \mathrm{T}\alpha &= \mathbf{c} \to \alpha & \qquad \mathbb{E}_{\mathbf{e}}\,t &= t\,\mathsf{nil} \\ \mathsf{U}\,t &= \lambda c.\, t & \mathbb{E}_{\mathbf{t}}\,t &= \lambda ck.\, (t\,c) \wedge (k\,c) \\ t \bullet u &= \lambda c.\, t\,c\,(u\,c) & \mathbb{P}_{\mathbf{e}}\,t &= \lambda c.\, t \\ \mathsf{C}\,t &= \lambda cx.\, t\,x\,c & \mathbb{P}_{\mathbf{t}}\,t &= \lambda c.\, t\,c\,(\lambda c.\,\mathsf{true}) \end{aligned}$$

A conservativity result may then be established along the same lines as in the case of intensionalization.

It is to be noted, however, that the dynamization procedure given in [6] uses the above general scheme for dynamizing all the constants but the logical connectives, for which special translations are given. Dynamic conjunction, for instance, is defined as follows:

$$\overline{\wedge} = \lambda abek.\, a\,e\,(\lambda e.\, b\,e\,k)$$

Consequently, in order to establish the conservativity result, it must be shown "by hand" that the logical connectives are logically related to their dynamic interpretations.

4.3 Type Raising

As a last example, we consider Montague's type raising. Here the picture is a little bit different. The signatures Σ_0 and Σ_1, which are given, are such that $A_0 = \{\mathbf{n}, \mathbf{np}, \mathbf{s}\}$ and $A_1 = \{\mathbf{e}, \mathbf{t}\}$. Intuitively, Σ_0 specifies the syntax of a fragment of natural language that does not contain quantified noun phrases. Σ_1 specifies the object language in which a Montagovian interpretation of this fragment is given, and the morphism $(\ )^*$ corresponds to this Montagovian interpretation. In particular, at type level, we have:

$$\begin{aligned} \mathbf{n}^* &= \mathbf{e} \to \mathbf{t} \\ \mathbf{np}^* &= \mathbf{e} \\ \mathbf{s}^* &= \mathbf{t} \end{aligned}$$

Σ_2 is then identical to Σ_1, while the morphism $\overline{(\ )}$ interprets noun phrases as type-raised entities:

$$\begin{aligned} \overline{\mathbf{n}} &= \mathbf{e} \to \mathbf{t} \\ \overline{\mathbf{np}} &= (\mathbf{e} \to \mathbf{t}) \to \mathbf{t} \\ \overline{\mathbf{s}} &= \mathbf{t} \\ \overline{c} &= \mathbb{E}_{\tau_0(c)}\,(\mathsf{U}\, c^*) \end{aligned}$$

with the following primitives:

$$\begin{aligned} \mathrm{T}\alpha &= (\alpha \to \mathbf{t}) \to \mathbf{t} \\ \mathsf{U}\, t &= \lambda k.\, k\, t \\ t \bullet u &= \lambda k.\, t\,(\lambda x.\, u\,(\lambda y.\, k\,(x\, y))) \\ \mathsf{C}\, t &= \lambda k.\, k\,(\lambda x.\, \mathsf{L}\,(t\, x)) \end{aligned} \qquad \begin{aligned} \mathbb{E}_{\mathbf{n}}\, t &= \lambda x.\, t\,(\lambda k.\, k\, x) \\ \mathbb{E}_{\mathbf{np}}\, t &= t \\ \mathbb{E}_{\mathbf{s}}\, t &= t\,(\lambda x.\, x) \\ \mathbb{P}_{\mathbf{n}}\, t &= \lambda k.\, k\, t \\ \mathbb{P}_{\mathbf{np}}\, t &= t \\ \mathbb{P}_{\mathbf{s}}\, t &= \lambda k.\, k\, t \end{aligned}$$

where L is a function of type $((\alpha \to \mathbf{t}) \to \mathbf{t}) \to \alpha$ such that $\mathsf{L}\,(\lambda k.\, k\, t) = t$.[1]

To illustrate the above embedding, assume that C_0 contains a constant $\textsc{which}$ of type $(\mathbf{np} \to \mathbf{s}) \to \mathbf{n} \to \mathbf{n}$, with the following Montagovian interpretation:

$$\textsc{which}^* = \lambda cnx.\, (n\, x) \wedge (c\, x)$$

Then, we obtain:

$$\begin{aligned} \overline{\textsc{which}} &= \mathbb{E}_{(\mathbf{np} \to \mathbf{s}) \to \mathbf{n} \to \mathbf{n}}\,(\mathsf{U}\, \textsc{which}^*) \\ &= \lambda cnx.\, (n\, x) \wedge (c\,(\lambda k.\, k\, x)) \end{aligned}$$

5 Future Work

As explained in the introduction, this paper reports some ongoing work. Several questions remain to be settled.

[1] At the semantic level, such a function exists for every type α. However, it is not λ-definable. Consequently, at the syntactic level, we must add constants of the appropriate types, together with additional reduction rules.

A first question concerns the genericity of the approach. Did we reach an appropriate level of abstraction? To answer this question, we should study more possible instances of our construct. With this respect, a possible candidate is Champolion's proposal of interpreting sentences as generalized quantifiers over events [2].

One may also question the nature of the type transformer T. Does it correspond somehow to any known kind of structure? For instance, the types assigned to U and $\bullet$ make sense from a modal logic point of view. This, however, is not the case for the type assigned to C. It is also the case that operations U and $\bullet$ can be defined in every strong monad in such a way that C1 holds. Again, this is not the case for C. Is there any sensible condition that one could add to the definition of a monad in order to ensure the existence of C?

We have chosen to work at the syntactic level of the λ-terms because we think that it captures Montague's strategy of expressing the semantic interpretation of a natural language fragment through a translation into an intermediate object language [8]. Our treatment, however, is not completely satisfactory. For dynamization, for instance, the notion of equality we need at type $\mathbf{t}$ is a relation of logical equivalence stronger than $\beta\eta$-equivalence. Similarly, for type-raising, we need to require that $\mathsf{L}\,(\lambda k.\, k\, x) = x$. The situation would be simpler at the semantic level of Henkin models because the needed notions of equality would correspond with the equality in the models. So, maybe would it better to define our construct at a model theoretical level (as it is the case in [5]). The syntactic version of the construct would then be obtained by considering term models.

Our ultimate goal is to establish conservativity properties. To this end, it would be nice to prove a generic conservativity theorem at the abstract level of our construct. This requires the definition of an abstract notion of consequence relation.

The types and the λ-terms built upon a given signature may be seen as the objects and the arrows of a Cartesian closed category. With respect to this vision, our notion of morphism between two signatures is a functor between two Cartesian closed categories. This turns out to be a particular case of the notion of *type refinement system* recently introduced by Melliès and Zeilberger [7]. In fact, it might be the case that Melliès and Zeilberger's setting provides a good abstract framework in which we could express our generic construct. This is a question we would like to explore.

References

[1] H. Barendregt, W. Dekkers, and R. Statman. *Lambda Calculus with Types.* Cambridge University Press, 2013.

[2] L. Champollion. The interaction of compositional semantics and event semantics. *Linguistic & Philosophy*, 38(1):31–66, 2015.

[3] Ph. de Groote. Towards abstract categorial grammars. In *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference*, pages 148–155, 2001.

[4] Ph. de Groote. Towards a montagovian account of dynamics. In *Semantics and Linguistic Theory XVI.* Cornell University, Ithaca, NY, 2006.

[5] Ph.de Groote and M. Kanazawa. A note on intensionalization. *Journal of Logic, Language and Information*, 22:173–194, 2013.

[6] E. Lebedeva. *Expression de la dynamique du discours à l'aide de continuations.* Thèse de doctorat, Universit de Lorraine, 2012.

[7] P.-A. Melliès and N. Zeilberger. Functors are type refinement systems. In *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2015*, pages 3–16, 2015.

[8] R. Montague. The proper treatment of quantification in ordinary english. In J. Hintikka, J. Moravcsik, and P. Suppes, editors, *Approaches to natural language: proceedings of the 1970 Stanford workshop on Grammar and Semantics*, Dordrecht, 1973. Reidel. Reprinted: [9, pages 247–270].

[9] R. Montague. *Formal Philosophy: selected papers of Richard Montague, edited and with an introduction by Richmond Thomason.* Yale University Press, 1974.