Weakly definite Descriptions

Staffan Angere

University of Lund

May 23, 2014

Defining operators from predicates

In logic textbooks aimed at philosophers it is common to only include predicates in the language and rely on contextually defined (Russellian) definite descriptions to handle function symbols.

This quickly becomes tedious when one works with a lot of functions, as in e.g. algebra. Therefore more mathematically oriented first-order logicians tend to prefer their languages to include function symbols as primitives as well.

A nice way to get both the metalinguistic simplicity of having only relations, and the usefulness of function symbols, is to add a definite description operator $\imath$ instead and define the functions from relations.

Defining operators from predicates

Common requirements on a definite description operator: $\iota x \varphi(x)$ denotes iff

1. there is at least one x such that $\varphi(x)$, and
2. there is at most one x such that $\varphi(x)$.

We will not overly concern ourselves here with cases in which (i) does not hold, although we will be working in free logic in which it is less of a problem if it does not hold. Our main aim in this talk is to weaken (ii).

Isomorphism and identity

Why would we need to weaken (*ii*)? Let's look at some mathematics.

Most mathematicians tend to work with objects that are determined only up to isomorphism, or up to some other equivalence: if she proves that all things satisfying a certain theory are isomorphic, she will be able to use that description to denote one of those things. No uniqueness necessary.

Philosophical structuralist: in mathematics, isomorphic structures *are* identical, so uniqueness is automatically satisfied here!

The philosophical structuralist is wrong; to identify all isomorphic objects leads to contradiction or loss of descriptive power, at least in classical logic. "Isomorphism is one thing, identity another".

Isomorphism and identity

$x = y$ iff all predicates of $\mathcal{L}$ are shared among x and y.[1]
$x \cong y$ iff there is an invertible transforamation $f : x \to y$ in $\mathcal{C}$.

Both are relative, but $=$ is relative to a language $\mathcal{L}$, and $\cong$ to a category $\mathcal{C}$. So what counts as an isomorphism between mathematical objects depends on what category you see them as being objects of, which in turn depends on what transformations you count as structure-preserving. E.g. Hilbert spaces with bounded linear maps and Hilbert spaces with unitary transformations lead to different isomorphism relations.

In particular, $x \cong y$ is *not* sufficient for us to be allowed to infer $\varphi(y)$ from $\varphi(x)$ for all φ; this holds only for *structural* properties $\varphi(\,\cdot\,)$, where 'structural' is determined by $\mathcal{C}$.

[1]For something close to the standard semantic interpretation of '=' substitute 'iff all predicates of $\mathcal{L}$'s metalanguage are shared among $[\![x]\!]$ and $[\![y]\!]$'.

The logic of isomorphism

Isomorphism logic is a modal logic in which $\Diamond\varphi$ is interpretable as *up to isomorphism*, φ. It is an extension of *transformational logic*, which is a class of modal logics where $\Box$ and $\Diamond$ quantify over first-order model homomorphisms rather than worlds. Compared to modal logics with Kripkean semantics, these can be used to describe multigraphs, and not only graphs.

The advantage of isomorphism logic is that it allows several forms of equivalence to coexist peacefully: in the theory **Cat** of categories, we have:

$$\begin{aligned} C = D & \quad \text{iff } C \text{ and } D \text{ are identical} \\ \Diamond C = D & \quad \text{iff } C \text{ and } D \text{ are isomorphic} \\ \Diamond\Diamond C = D & \quad \text{iff } C \text{ and } D \text{ are category equivalent} \end{aligned}$$

They were originally designed for working in weak higher categories, in which an infinity of such equivalences are relevant.

Frames for TL

A *frame* for a transformational logic language $\mathcal{L}$ is a class $\mathcal{M}$ of first-order models for $\mathcal{L}$'s signature together with a mapping $\Theta : \mathcal{M} \to \wp(\mathcal{M}^{\mathcal{M}})$ such that all elements in Θ_M are

1. first-order model homomorphisms,
2. onto, and
3. have a submodel of their domain as codomain.

A *transformational model* is a transformational frame $\mathcal{M}, \Theta$ together with a specified element $M \in \mathcal{M}$, i.e. a pointed t-frame. The truth conditions of formulae in a model $\mathcal{M}, \Theta, M$ under a valuation v can be taken to be very close to the usual ones for first-order logic with identity. The main additions are the rules for $\Box$ and $\Diamond$:

$$\begin{array}{lll} \mathcal{M}, \Theta, M \vDash_v & \Box\varphi & \text{iff } \mathcal{M}, \Theta, \mathrm{cod}\,\tau \vDash_{\tau \circ v} \varphi \text{ for all } \tau \in \Theta_M \\ \mathcal{M}, \Theta, M \vDash_v & \Diamond\varphi & \text{iff } \mathcal{M}, \Theta, \mathrm{cod}\,\tau \vDash_{\tau \circ v} \varphi \text{ for some } \tau \in \Theta_M \end{array}$$

Adding ε-terms: syntax

1. ‘ ’ (the empty string) is a 0-term.
2. $x_1, x_2, x_3, \ldots$ are 1-terms.
3. If x_k, x_l are n- and m-terms, respectively, then x_k, x_l is an $n + m$-term.
4. If t is an n-term and f is an n-ary function symbol, $f(t)$ is a 1-term.
5. If t is an n-term and P is an n-ary predicate, then $P(t)$ is a formula.
6. If t_1, t_2 are 1-terms, then $t_1 = t_2$ is a formula. If t is a 1-term, then $t\downarrow$ is a formula.
7. If φ is a formula, then $\varepsilon x_{k_1}, \ldots, x_{k_n}\, \varphi$ is an $n - m$-term, where m is the number of free variables of φ that also occur in $x_{k_1}, \ldots, x_{k_n}$.
8. Clauses for quantifiers and connectives.

Adding ε-terms: semantics

$$P(t_1, \ldots, t_k)$$

n-predicate | *n*-term

Why the introduction of *n*-terms and arbitrary-arity epsilon operators? They are useful when we want to work with *n*-tuples, as in definitions of products and coproducts in category theory.

For each $M \in \mathcal{M}$ with domain D we let a variable assignment s be a function from Var to D. Denote the set of all variable assignments on M with S_M. Let an *n-term valuation* tv_s^n based on s be a partial function from $Term^n$ to D^n, and let V_M^n be the set of all *n*-term valuations on M. Let a *formula valuation* based on s be a function fv_s from the formulae to $\wp(S_M)$. Add a function $c_M : \wp(S_M) \to S_M$ such that $c_S(X) \in X$, i.e. a choice function on S_M.

Adding ε-terms: semantics

Impose the following axioms on tv:

1. The value of tv_s^n, when defined, is an n-tuple.
2. $tv_s^1(x_k) = s(x_k)$.
3. $tv_s^{n+m}(t_1, t_2) = \langle tv_s^n(t_1), tv_s^m(t_2)\rangle$ where t_1 is an n-term and t_2 is an m-term.
4. $tv_s^n(f(t)) = M(f)(tv_s^n(t))$ for each n-ary function symbol f and n-term t.
5. $tv_{\tau\circ s}^1(\tau(tv^1(t))) = \tau(tv_s^1(t))$ for each $\tau : M \to M'$ in Θ_M and each 1-term t.
6. $tv_s^n(\varepsilon x_{k_1}, \ldots, x_{k_m}\ \varphi) =$ $\langle c_s(fv_s(\varphi))(x_{k_1}), \ldots, c_s^m(fv_s(\varphi))(x_{k_m})\rangle$.

Adding ε-terms: semantics

And the following axioms on fv:

1. $fv_s(P(t)) = \{s \in S_M \mid tv_s^n(t) \in M(P)\}$ for any n-term t.
2. $fv_s(t\downarrow) = \{s \in S_M \mid tv_s^n(t)$ is defined$\}$.
3. $fv_s(t_1 = t_2) = \{s \in S_M \mid t_1\downarrow \wedge t_2\downarrow \wedge tv_s^1(t_1) = tv_s^1(t_2)\}$.
4. $fv_s(\Box\varphi) = \bigcap_{\tau\in\Theta_M} fv_{\tau\circ s}(\varphi)$
5. $fv_s(\Diamond\varphi) = \bigcup_{\tau\in\Theta_M} fv_{\tau\circ s}(\varphi)$
6. $fv_s(\varphi \wedge \psi) = fv_s(\varphi) \cap fv_s(\psi)$
7. Other clauses for connectives and quantifiers.

I.e. the semantic value of a formula is the set of variable assignments under which it is true.

Inference in TL

A sound and complete tableau-based inference system for basic TL, where the formulae are labeled with transformation signs, already exists. We outline it here before going through the additions needed to add ε-terms.

Rules for connectives: as usual.

Quantifiers:

$$\frac{\tau : \forall x_k\ \varphi \qquad \sigma \circ \tau : t\!\downarrow}{\sigma \circ \tau : \varphi[t \,/\, x_k]} \qquad \frac{\tau : \neg\forall x_k\ \varphi}{\tau : \neg\varphi[t' \,/\, x_k]}$$

$$\frac{\tau : \exists x_k\ \varphi}{\tau : \varphi[t' \,/\, x_k]} \qquad \frac{\tau : \neg\exists x_k\ \varphi \qquad \sigma \circ \tau : t\!\downarrow}{\sigma \circ \tau : \neg\varphi[t \,/\, x_k]}$$

(t' new)

Inference in TL

Identity:

$$\frac{\tau : t\downarrow}{\tau : t = t} \qquad \frac{\begin{array}{c}\sigma : t_1 = t_2 \\ \tau \circ \sigma : \varphi\end{array}}{\tau \circ \sigma : \varphi[t_1/t_2]}$$

Modalities:

$$\frac{\sigma : \Box\varphi}{\tau \circ \sigma : \varphi} \qquad \frac{\sigma : \neg \Box \varphi}{\tau' \circ \sigma : \neg\varphi}$$

$$\frac{\sigma : \Diamond\varphi}{\tau' \circ \sigma : \varphi} \qquad \frac{\sigma : \neg \Diamond \varphi}{\tau \circ \sigma : \neg\varphi}$$

(τ' new and atomic)

Inference in TL

Definedness rules:

$$\frac{\tau : P^n(t_1, \ldots, t_n)}{\tau : t_k{\downarrow}} \qquad \frac{\tau : f^n(t_1 \ldots, t_n){\downarrow}}{\tau : t_k{\downarrow}}$$

$$\frac{\tau : t_1 = t_2}{\begin{array}{l}\tau : t_1{\downarrow} \\ \tau : t_2{\downarrow}\end{array}} \qquad \frac{\tau : \varphi}{\tau : t'{\downarrow}}$$

t_k is required to be one of $t_1, \ldots, t_n$, and t' is required to be atomic.

Inference in TL

Homomorphism rule:

$$\frac{\sigma : P^n(t_1, \ldots, t_n)}{\tau \circ \sigma : P^n(t_1, \ldots, t_n)} \qquad \frac{\tau \circ \sigma : \neg P^n(t_1, \ldots, t_n)}{\sigma : \neg P^n(t_1, \ldots, t_n)}$$

Submodel rule:

$$\begin{array}{c} \tau \circ \sigma : P^n(t_1, \ldots, t_n) \\ \hline \tau \circ \sigma : t_1 = t_1' \\ \vdots \quad \vdots \quad \vdots \\ \tau \circ \sigma : t_n = t_n' \\ \sigma : P^n(t_1', \ldots, t_n') \end{array}$$

where $t_1', \ldots, t_n'$ are new.

Inferences with ε

Two additions to the previous system:

$$\frac{\tau : \varphi(\varepsilon x_1, \ldots, x_n\ \psi) \qquad \tau : \varepsilon x_1, \ldots, x_n \psi{\downarrow}}{\begin{array}{c}\tau : \varphi[t_1, \ldots, t_n \,/\, x_1, \ldots, x_n] \\ \tau : \psi[t_1, \ldots, t_n \,/\, x_1, \ldots, x_n]\end{array}}$$

$$\frac{\tau : \neg\varphi(\varepsilon x_1, \ldots, x_n\ \psi) \qquad \tau : \varepsilon x_1, \ldots, x_n\ \psi{\downarrow}}{\begin{array}{c}\tau : \neg\varphi[t_1, \ldots, t_n \,/\, x_1, \ldots, x_n] \\ \tau : \psi[t_1, \ldots, t_n \,/\, x_1, \ldots, x_n]\end{array}}$$

where $t_1, \ldots, t_n$ are new.

Inferences with ε

From these it follows that

$$\forall x \varphi(x) \Leftrightarrow \varphi(\varepsilon x \neg \varphi(x))$$
$$\exists x \varphi(x) \Leftrightarrow \varphi(\varepsilon x \varphi(x))$$

as usual with the ε-operator.

I do not know yet whether this inference system is complete with respect to its intended semantics. It is probably sound though.

A weakly definite description operator

Define ι_w^k, for $k \in \mathbb{N}$, as

$$\iota_w^k x\varphi(x) =_{df.} \varepsilon x(\varphi(x) \wedge \forall y(\varphi(y) \rightarrow \Diamond^k y = x))$$

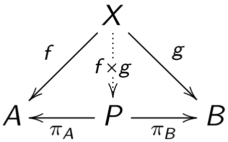
It follows at once that $\iota_w^1\varphi(x)$ is defined in M under the term valuation tv iff there is an x such that

1. $\varphi(x)$ and
2. for any two x, y such that $\varphi(x)$ and $\varphi(y)$, there is a $\tau \in \Theta_M$ such that $\tau(vt(x)) = \tau(vt(y))$.

When Θ contains isomorphisms in the appropriate sense, $\iota_w^1\varphi(x)$ is defined iff x is unique up to isomorphism.

An application

A *product* of the objects A, B in a category $\mathcal{C}$ is a triple $\langle P, \pi_A, \pi_B \rangle$ where $\pi_A : P \to A$ and $\pi_B : P \to B$ such that the following diagram commutes for all $f : X \to A$ and $g : X \to B$:



Such products are determined only up to isomorphism, but it is still common to work with them as if they were unique. In the following it will be advantageous to interpret an object A as a triple $A, 1_A, 1_A$. Let $Pr(P, \pi_A, \pi_B, A, B)$ be the 5-ary predicate P, π_A, π_B *is a product of* A *and* B . We define

$$(A, f, g) \times (B, j, k) =_{df.} \iota_w x, y, z \; Pr(x, y, z, A, B)$$

An application

This lets us work with products algebraically and form complex expressions such as $A \times (B \times C)$. These will not in general satisfy commutativity or associativity but instead the weaker relations

$$\Diamond(A \times B = B \times A)$$

$$\Diamond(A \times (B \times C) = (A \times B) \times C)$$

Why is this useful? Because we can simply work as usual, algebraically. Although we do not have identities so we cannot guarantee that $\Diamond A = B$ enables us to replace A by B, it does so in all contexts preceded by a $\Box$: it becomes easier to isolate the parts of a calculation that depend on the choice of products from the parts that do not.

Structuralism

This provides a convenient way to work structurally. Although there is no purely structural classical-logical way to do mathematics (i.e. one that does not require using non-identical isomorphic objects), using $\gimel_w$ and the transformational modalities shows which statements are structural, i.e. do not depend on those specific identities.

Thank You!

Another application

We have not imposed any extensionality conditions on ε-terms. If we do so, it may give a way to define *canonical* isomorphisms rather than just arbitrary ones. This would be useful in the definition of weak structures, since these generally require the composition of some (but not all) specified morphisms to give unique results.

Example: monoidal categories, i.e. categories equipped with a binary operator that commutes and associates up to isomorphism. These are taken to be specified by the category (but not usually unique). A *coherence theorem* is proved in order to show that composing such isomorphisms again gives a canonical isomorphism.

Such coherence theorems are often difficult to prove, at least for more complicated categorical structures. Finding the right kind of extensionality for ε-terms could help in this by giving a coherence theorem in a very general form, through simple semantical requirements on the composition of morphisms involved in the definition of these terms.