TRANSLATION FROM MODAL LANGUAGE INTO NONMODAL LANGUAGE

EVGENY BORISOV

1. First-order modal language $\mathcal{L}$

The vocabulary of $\mathcal{L}$ includes:

- the set Var of individual variables $x, y, ...$,
- the set Con of individual constants $a, b, ...$,
- the set $Pred$ of n-ary predicate letters $P, Q, ...$ $(n \geq 1)$,
- logical operators $\neg, \&, \Diamond, \exists, \lambda$,
- brackets (,).

All the sets of symbols are assumed to be pairwise disjoint. This also holds for the language $\mathcal{L}'$ to be described in the next section.

The syntax of $\mathcal{L}$

$\mathcal{L}$'s terms are individual variables and constants. The set of formulae ϕ of $\mathcal{L}$ is defined as follows:

$$\phi ::= P(x_1, ..., x_n) \,|\, \neg\phi \,|\, \phi\&\phi \,|\, \Diamond\phi \,|\, \exists x\phi \,|\, (\lambda x.\phi)(t),$$

where P is an n-ary predicate, $x, x_1, ..., x_n$ are variables, t is a term. Notice that individial constants cannot occur in atomic formulae. If you whant to say in $\mathcal{L}$ that a is P, you have to write $(\lambda x.P(x))(a)$, not $P(a)$.

Notational convention

In what follows, I write $(\lambda x.\phi)(t)$ as $(t/x)\phi$.

Models for $\mathcal{L}$

A model $\mathcal{M}$ for $\mathcal{L}$ is a quadruple $\langle \mathcal{G}, \mathcal{R}, (\mathcal{D}_w)_{w\in G}, \mathcal{I}\rangle$, where:

- $\mathcal{G}$ is a nonempty set of possible worlds,
- $\mathcal{R} \subseteq \mathcal{G}^2$ is the accessability relation between possible worlds,
- $(\mathcal{D}_w)_{w\in G}$ is a familty of nonempty sets - domains of possible worlds,
- $\mathcal{I}$ is the interpretation of individual constants and predicates, such that
  - for each individual constant a, $\mathcal{I}(a) : G \to D$,
  - for each n-ary predicate letter P, $\mathcal{I}(P) : G \to \mathcal{P}(\mathcal{D}^n)$, where $\mathcal{D} := \bigcup_{w\in\mathcal{G}} \mathcal{D}_w$, and $\mathcal{P}$ stands for powerset. [1]

Notational conventions

1) $\mathcal{M}, w, v \vDash \phi$ says that the formula ϕ is true at the possible world w of the model $\mathcal{M}$ under the variable valuation v.

[1] If we want to have a logic with equality, we add $=$ to $Pred$ and set $\mathcal{I}(=)(w)$ to be identity relation on $\mathcal{D}$ for every w.

2) Let v be a variable valuation in $\langle \mathcal{G}, \mathcal{R}, (\mathcal{D}_w)_{w\in G}, \mathcal{I}\rangle$, i.e. $v : Var \to D$. Then $v\mathcal{I}$ is a function mapping each term and possible world to an object, such that for each term t and possible world w,

$$v\mathcal{I}(t,w) = \begin{cases} v(t) & \text{if } t \in Var \\ \mathcal{I}(t)(w) & \text{if } t \in Con. \end{cases}$$

3) If v is a variable valuation in $\langle \mathcal{G}, \mathcal{R}, (\mathcal{D}_w)_{w\in G}, \mathcal{I}\rangle$, $e \in \mathcal{D}$, and $x \in Var$, v^e_x is an x-variant of v mapping x to e, i.e.:

$$v^e_x(y) = \begin{cases} e & \text{if } y = x, \\ v(y) & \text{if } y \neq x. \end{cases}$$

Truth for $\mathcal{L}$

Let $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, (\mathcal{D}_w)_{w\in G}, \mathcal{I}\rangle$ be a model for $\mathcal{L}$, w a possible world in $\mathcal{G}$, v a variable valuation in $\mathcal{M}$, P an n-ary predicate letter, $x, x_1, ..., x_n$ variables, ϕ and ψ formulae. Then:

- $\mathcal{M}, w, v \vDash P(x_1, ..., x_n) \iff \langle v(x_1), ..., v(x_n)\rangle \in \mathcal{I}(P)(w)$;
- $\mathcal{M}, w, v \vDash \neg\phi \iff \mathcal{M}, w, v \nvDash \phi$;
- $\mathcal{M}, w, v \vDash \phi \& \psi \iff \mathcal{M}, w, v \vDash \phi$ and $\mathcal{M}, w, v \vDash \psi$;
- $\mathcal{M}, w, v \vDash \Diamond\phi \iff \mathcal{M}, w', v \vDash \phi$ for some w' with $w\mathcal{R}w'$;
- $\mathcal{M}, w, v \vDash \exists x\phi \iff \mathcal{M}, w, v^e_x \vDash \phi$ for some $e \in \mathcal{D}_w$;
- $\mathcal{M}, w, v \vDash (t/x)\phi \iff \mathcal{M}, w, v_x^{v\mathcal{I}(t,w)} \vDash \phi$.

Note that quantifiers range "just"over domains of possible worlds whereas predicates and constants are interpreted on the unions of domains of all possible worlds. Because of this, formulae like $(a/x)P(x)\&\forall x\neg P(x)$ might be true.

2. Two-sorted first-order nonmodal language $\mathcal{L}'$

In two-sorted languages, terms, functional symbols and predicate letters are typed using two elementary types. We will use elementary types $[e]$ and $[w]$. When matching models for $\mathcal{L}$ and models for $\mathcal{L}'$, we will associate $[e]$ with entities in models for $\mathcal{L}$, and $[w]$ with possible worlds in models for $\mathcal{L}$. Types of predicate letters are generated from elementary types as follows: if P is an n-ary predicate letter, its type is the string $[\tau_1, ..., \tau_n]$ where each τ_i is either $[e]$ or $[w]$. Same holds for functional symbols.

Vocabulary of $\mathcal{L}'$ includes:

- the set Var of individual variables $x, y, ...$ (taken from $\mathcal{L}$'s vocabulary); its members are set to be of $[e]$ type,
- the set Var^+ of possible world variables $\alpha, \beta, ...$; its members are of $[w]$ type,
- the set Con' of one-place [w]-type functional symbols. Symbols in Con' are generated by adding $'$ to individual constants in Con. E.g., if a is in Con, we have a' in Con';
- the set $Pred'$ of predicate letters. The members of $Pred'$ are obtained from letters in $Pred$ by adding $'$. E.g., if P is in $Pred$, P' is in $Pred'$. If P is an n-ary predicate letter in $Pred$, P' is an $n+1$-ary predicate letter of the type $[w, \underbrace{e, ..., e}_{n \text{ times}}]$,
- two additional predicate letters: R (of the type $[w, w]$) and E (of the type $[w, e]$,
- logical operators $\neg, \&, \Diamond, \exists$ (note that λ-operator is not it $\mathcal{L}'$'s vocabulary),
- brackets (,).

The syntax of $\mathcal{L}'$

$\mathcal{L}'$'s terms of $[e]$ type are individual variables (members of Var) and expressions of the form $a'(\alpha)$, where $a' \in Con'$ and $\alpha \in Var^+$. [2]

$\mathcal{L}'$'s terms of $[w]$ type are possible world variables (members of Var^+).

The set of formulae ϕ of $\mathcal{L}'$ is defined as follows:

$$\phi ::= P(t_1, ..., t_n) \,|\, \neg\phi \,|\, \phi \& \phi \,|\, \exists x \phi \,|\, \exists \alpha \phi,$$

where P is an n-ary predicate letter (it may be either Q' for some Q in $Pred$, or R, or E); $t_1, ..., t_n$ are terms whose types match the type of P, i.e. if P is of the type $[\tau_1, ..., \tau_n]$, t_i is of the type $[\tau_i]$ for every i $(1 \leq i \leq n)$. [3]

Models for $\mathcal{L}'$

A model $\mathcal{M}$ for $\mathcal{L}'$ is a triple $\langle \mathcal{D}, \mathcal{G}, \mathcal{I} \rangle$, where $\mathcal{D}$ and $\mathcal{G}$ are nonempty domains, and $\mathcal{I}$ is an interpretation of function symbols and predicate letters, such that:

- for every a' in Con', $\mathcal{I}(a') : G \to D$,
- for every $n+1$-ary predicate letter P' in $Pred'$, $\mathcal{I}(P') \subseteq \mathcal{G} \times \mathcal{D}^n$;
- $\mathcal{I}(R) \subseteq \mathcal{G}^2$;
- $\mathcal{I}(E) \subseteq \mathcal{G} \times \mathcal{D}$.

Variable valuations im Models for $\mathcal{L}'$

Let $\mathcal{M} = \langle \mathcal{D}, \mathcal{G}, \mathcal{I} \rangle$ be a model for $\mathcal{L}'$. A variable valuation v im $\mathcal{M}$ is a function mapping each variable in Var to a member of $\mathcal{D}$, and each variable in Var^+ to a member of $\mathcal{G}$. Thus, we have:

- $v : Var \cup Var^+ \to \mathcal{D} \cup \mathcal{G}$
- for each x in Var, $v(x) \in \mathcal{D}$
- for each α in Var^+, $v(\alpha) \in \mathcal{G}$.

Notational conventions

When evaluating formulae of $\mathcal{L}'$ in models for $\mathcal{L}'$ we will write $\mathcal{M}, v \models \phi$ for «ϕ is true in $\mathcal{L}'$ under v». [4]

Denotation for $\mathcal{L}'$

For each function symbol a' in Con', $\mathcal{I}(a') : \mathcal{G} \to \mathcal{D}$.

$v\mathcal{I}$ is a function assigning to each term t a denotation as follows:

$$v\mathcal{I}(t) = \begin{cases} v(t) & \text{if } t \in Var \cup Var^+ \\ \mathcal{I}(a')(v(\alpha)) & \text{if } t = a'(\alpha) \text{ for some } a' \text{ and } \alpha.^5 \end{cases}$$

Truth for $\mathcal{L}'$

Let $\mathcal{M} = \langle \mathcal{D}, \mathcal{G}, \mathcal{I} \rangle$ be a model for $\mathcal{L}'$, v a variable valuation in $\mathcal{M}$, P an n-ary predicate letter, $t_1, ..., t_n$ terms (whose tipes match the type of P), x a variable in Var, α a variable in Var^+, ϕ and ψ formulae. Then:

- $\mathcal{M}, v \models P(t_1, ..., t_n) \iff \langle v\mathcal{I}(t_1), ..., v\mathcal{I}(t_n) \rangle \in \mathcal{I}(P')$
- $\mathcal{M}, v \models \neg\phi \iff \mathcal{M}, v \not\models \phi$

---

[2] We intend to interprete a' as a function from possible worlds to entities. That is why a' is applied to arguments of $[w]$ type whereas the term $a'(\alpha)$ is of $[e]$ type.

[3] For instance, if P is of the type $[e, w]$, $P(x, \alpha)$ is a formula whereas the following expressions are not: $P(x, y)$, $P(\alpha, \beta)$, $P(\alpha, x)$.

[4] Thus, $\models$ has different meanings in evaluation of formulae of $\mathcal{L}$ and in evaluation of formulae of $\mathcal{L}'$ but this should not lead to confusion.

- $\mathcal{M}, v \vDash \phi \& \psi \iff \mathcal{M}, v \vDash \phi$ and $\mathcal{M}, v \vDash \psi$
- $\mathcal{M}, v \vDash \exists x \phi \iff \mathcal{M}, v^e_x \vDash \phi$ for some $e \in \mathcal{D}$
- $\mathcal{M}, v \vDash \exists \alpha \phi \iff \mathcal{M}, v^w_x \vDash \phi$ for some $w \in \mathcal{G}$.

3. Translation from $\mathcal{L}$ into $\mathcal{L}'$

We define a family $(T_\alpha)_{\alpha \in Var^+}$ of translation functions mapping terms and formulae of $\mathcal{L}$ to terms and formulae of $\mathcal{L}'$.

One more notational convention

If ϕ is a formula, x is an individual variable and t a term, ϕ^t_x is a formula obtained from ϕ by replacing each free occurrence of x by an occurrence of t. This convention can be applied to formulae of both languages but if we apply it to a formula of $\mathcal{L}'$ we should make sure that t is of $[e]$ type.

Translation functions

Let α be a member of Var^+. Then the translation function T_α is defined by the following clauses:

- for every $x \in Var$, $T_\alpha(x) = x$;
- for every $a \in Con$, $T_\alpha(a) = a'(\alpha)$;
- for every P in $Pred$, $T_\alpha(P(x_1, ..., x_n)) = P'(\alpha, x_1, ..., x_n)$;
- $T_\alpha(\neg\phi) = \neg T_\alpha(\phi)$;
- $T_\alpha(\phi \& \psi) = T_\alpha(\phi) \& T_\alpha(\psi)$;
- $T_\alpha(\Diamond\phi) = \exists\beta(R(\alpha, \beta) \& T_\beta(\phi))$, where β is a new variable of $[w]$ type; [6]
- $T_\alpha(\exists x\phi) = \exists x(E(\alpha, x) \& T_\alpha(\phi))$;
- $T_\alpha((t/x)\phi) = (T_\alpha(\phi))^{T_\alpha(t)}_x$.

An illustration. Here is an example of translation:

$$T_\alpha\Big(\exists x \Diamond (a/y) \neg P(x, y)\Big) =$$

$$= \exists x \left(E(\alpha, x) \& T_\alpha\Big(\Diamond (a/y) \neg P(x, y)\Big) \right) =$$

$$= \exists x \left(E(\alpha, x) \& \exists\beta\Big(R(\alpha, \beta) \& T_\beta\big[(a/y)\neg P(x, y)\big]\Big) \right) =$$

$$= \exists x \left(E(\alpha, x) \& \exists\beta\Big(R(\alpha, \beta) \& \big[T_\beta(\neg P(x, y))\big]^{a'(\beta)}_y\Big) \right) =$$

$$= \exists x \left(E(\alpha, x) \& \exists\beta\Big(R(\alpha, \beta) \& \neg\big[T_\beta(P(x, y)\big]^{a'(\beta)}_y\Big) \right) =$$

$$= \exists x \left(E(\alpha, x) \& \exists\beta\Big(R(\alpha, \beta) \& \neg\big[P'(\beta, x, y)\big]^{a'(\beta)}_y\Big) \right) =$$

$$= \exists x \left(E(\alpha, x) \& \exists\beta\Big(R(\alpha, \beta) \& \neg P'\big(\beta, x, a'(\beta)\big)\Big) \right)$$

[6] Notice the switch from T_α to T_β.

4. Translations are truth-presering

Translation functions just defined are truth-preserving in a certain sense. To state this precisely, we have to define a function g from models for $\mathcal{L}$ to models for $\mathcal{L}'$.

The function g from models for $\mathcal{L}$ to models for $\mathcal{L}'$
Let $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, (\mathcal{D}_w)_{w \in G}, \mathcal{I} \rangle$ be a model for $\mathcal{L}'$.
Then $g(\mathcal{M}) = \langle \mathcal{G}, \mathcal{D}, \mathcal{I}' \rangle$ with $\mathcal{D} = \bigcup_{w \in \mathcal{G}} \mathcal{D}_w$ and $\mathcal{I}'$ defined as follows:

- For every a' in Con', $\mathcal{I}(a')$ is a function mapping each w in $\mathcal{G}$ to $\mathcal{I}(a)(w)$. I.e., $\mathcal{I}'(a')(w) = \mathcal{I}(a)(w)$.
- For every $n+1$-ary predicate letter P' in $Pred'$, $\mathcal{I}(P') = \{\langle w, e_1, ..., e_n \rangle : \langle e_1, ..., e_n \rangle \in \mathcal{I}(P)(w)\}$.
- $\mathcal{I}'(E) = \{\langle w, e \rangle : e \in \mathcal{D}_w\}$.
- $\mathcal{I}'(R) = \mathcal{R}$.

Now we are in a position to precisely state in what sense T_α truth-preserving.

Proposition
Let $\mathcal{M}$ be a model for $\mathcal{L}$, w a possible world in $\mathcal{M}$, v a variable valuation in $\mathcal{M}$, $\mathcal{M}' = g(\mathcal{M})$, v' a variable valuation in $\mathcal{M}'$ such that v and v' agree on all variables in Var, and $v'(\alpha) = w$. Then for every formula ϕ of $\mathcal{L}$,

$$\mathcal{M}, w, v \vDash \phi \iff \mathcal{M}', v' \vDash T_\alpha(\phi).$$

Proof. By induction on the structure of ϕ.

Homework. Show that the truth conditions of $\exists x \Diamond (a/y) \neg P(x,y)$ w.r.t. $\mathcal{M}$, w and v are equivalent to those of $T_\alpha\Big(\exists x \Diamond (a/y) \neg P(x,y)\Big)$ w.r.t. $\mathcal{M}'$ and v' provided $\mathcal{M}$, w, v, $\mathcal{M}'$, and v' meet the conditions of the proposition above.