Homotopy Type Theory Class

LMFI 2019-2020

Syntactic translations

Hugo Herbelin

Syntactic translations in type theory

We call *syntactic translations* translations from the syntax of type theory (the source) to the syntax of type theory (the target) which preserve derivability.

Such translations can be used to validate the relative consistency of an axiom in the source with respect to the target. As such, they play the rôle of models.

The target being computational (it is a type theory), it indirectly gives a computational content to the axiom in the source. As such, they play the rôle of computational models.

We studied the following syntactic translations

- Enrichment of functions with hidden computational data so as to invalidate functional extensionality (and univalence)
- Collapse of all types to the unit type in the absence of a universe

We will here study the following syntactic translations

- Realisability translation
- Parametricity translation

Bibliography

Guilhem Jaber, Gabriel Lewerstovski, Pierre-Marie Pédrot, Matthieu Sozeau, Nicolas Tabareau, *The definitional side of forcing*, 2016.

Pierre-Marie Pédrot, Nicolas Tabareau, *An effectful way to eliminate addiction to dependence*, 2017.

Simon Boulier, Pierre-Marie Pédrot, Nicolas Tabareau, *The next 700 syntactic models of type theory*, 2018.

Pierre-Marie Pédrot, Nicolas Tabareau, *Failure is not an option, The exceptional type theory*, 2018.

Thorsten Altenkirch, Simon Boulier, Ambrus Kaposi, and Nicolas Tabareau, *Setoid type theory, a syntactic translation*, 2019.

Realisability translation

The key idea of realisability is to interpret a type by a set of realisers, i.e. by a set of terms which have the property of "respecting" the type seen as a specification.

The original idea, by Kleene, uses Gödel numbers of partial recursive functions as realisers of Heyting Arithmetic.

Later, Kreisel uses programs of System T as realisers of Heyting Arithmetic extended with quantification over all programs of System T (so-called HA^ω, or arithmetic in finite types).

This generalizes to type theory where one can take the terms themselves as realisers.

The relation $t \mathsf{~r~} A$, read "t realises A" is inductively defined below as a predicate of A, i.e. as a predicate $[\![A]\!]$ of type $A \to U_0$.

Realisability translation

We concentrate on the core theory with Π-types:

$$
\begin{array}{lcl}
[\![a]\!]_\rho & \triangleq & \rho(a).\star \\
[\![\lambda a : A.\, t]\!]_\rho & \triangleq & \lambda a : A. \lambda a_\star : [\![A]\!]\, a.\, [\![t]\!]_{\rho,(a \mapsto (a, a_\star))} \\
[\![t\, u]\!]_\rho & \triangleq & [\![t]\!]_\rho\, u\, [\![u]\!]_\rho \\
 & & \\
[\![\Pi a : A.\, B]\!]_\rho & \triangleq & \lambda f : (\Pi a : A.\, B). \Pi a : A.\, \Pi a_\star : [\![A]\!]_\rho\, a.\, [\![B]\!]_{\rho,(a \mapsto (a, a_\star))}\, (f\ a) \\
[\![U_0]\!]_\rho & \triangleq & \lambda A : U_0. (A \to U_0) \\
 & & \\
[\![\square]\!] & \triangleq & \square \\
[\![\Gamma, a : A]\!] & \triangleq & [\![\Gamma]\!], a : A, a_\star : [\![A]\!]_{id_\Gamma}\, a \\
 & & \\
id_\square & \triangleq & \emptyset \\
id_{\Gamma, a:A} & \triangleq & id_\Gamma, (a \mapsto (a, a_\star))
\end{array}
$$

Note: we use a presentation of universes à la Russell; we also write $.\star$ to take the second projection of an element in ρ.

Realisability translation

Adequacy/soundness theorem:

$$\begin{pmatrix} \Gamma \ \mathsf{wf} \\ \Gamma \vdash t : A \\ \Gamma \vdash t \equiv u : A \\ \Gamma \vdash A : U_0 \\ \Gamma \vdash A \equiv B : U_0 \end{pmatrix} \text{ implies } \begin{pmatrix} [\![\Gamma]\!] \ \mathsf{wf} \\ [\![\Gamma]\!] \vdash [\![t]\!]_{id_\Gamma} : [\![A]\!]_{id_\Gamma}\, t \\ [\![\Gamma]\!] \vdash [\![t]\!]_{id_\Gamma} \equiv [\![u]\!]_{id_\Gamma} : [\![A]\!]_{id_\Gamma}\, t \\ [\![\Gamma]\!] \vdash [\![A]\!]_{id_\Gamma} : A \to U_0 \\ [\![\Gamma]\!] \vdash [\![A]\!]_{id_\Gamma} \equiv [\![B]\!]_{id_\Gamma} : A \to U_0 \end{pmatrix}$$

Realisability translation

We can extend with equality, along the lines of Kleene:

$$\begin{array}{lcl}
[\![\mathsf{refl}_A\, t]\!]_\rho & \triangleq & \mathsf{refl}_A\, t \\
[\![\mathsf{subst}^{a.b.P}\, p\, \mathsf{in}\, v]\!]_\rho & \triangleq & \mathsf{subst}^{a_\Box . b_\Box . ([\![P]\!]_{(\rho, a \mapsto a_\Box, b \mapsto b_\Box)}\, \mathsf{subst}^{a.b.P}\, b_\Box .1\, \mathsf{in}\, v)}\, [\![p]\!]_\rho\, \mathsf{in}\, [\![v]\!]_\rho \\
 & & \\
[\![t =_A u]\!]_\rho & \triangleq & \lambda p : t =_A u.\, t =_A u
\end{array}$$

where $a_\Box$ is the pair of a a and a $a_\star$ variable, and similarly for $b_\Box$.

Realisability translation

Exercise: we extend the translation to $\mathbb{N}$ by defining $[\![]\!]_\rho \triangleq N$ (i.e. in a simpler way as done in class). Gives the corresponding clauses for 0, $\succ$ and the recursor so that soundness holds.

Parametricity translation

The parametricity translation is the variant of realisability with relations: to each type is associated a relation over terms and the translation expresses that related terms map to related terms. The parametricity translation is related to cubical type theory, which is a "nice" form of type theory computing with univalence.

Example: The judgement

$$A : U_0, a : A \vdash a : A$$

was translated by realisability to:

$$A : U_0, A_\star : A \to U_0, a : A, a_\star : A_\star\, a \vdash a_\star : A_\star\, a$$

For parametricity, it is translated to:

$$A_0 : U_0, A_1 : U_0, A_\star : A_0 \to A_0 \to U_0, a_0 : A_0, a_1 : A_1, a_\star : A_\star\, a_0\, a_1 \vdash a_\star : A_\star\, a_0\, a_1$$

Parametricity translation

The parametricity translation triplicates objects, so we shall use three translations:

- two are the identities up to choosing the first or the second copy of terms; for realisability, there was only one identity translation, so we could keep it tacit; for parametricity, we have to distinguish them;
- the third one relates objects

They all depend on a substitution with three components, which components we shall denote with projection names $.0$, $.1$, and $.\star$.

Also, if ρ is a substitution, we shall write ρ_0 for its restriction to the fields .0 of ρ, and ρ_1 for its restriction to the fields .1 of ρ.

Parametricity translation

We concentrate first on Π types and the universe:

$$\begin{array}{lcl}
[\![a]\!]_\rho & \triangleq & \rho(a).\star \\
[\![\lambda a : A.\, t]\!]_\rho & \triangleq & \lambda a_0 : \rho_0(A).\lambda a_1 : \rho_1(A).\lambda a_\star : [\![A]\!]\, a_0\, a_1.\, [\![t]\!]_{\rho,(a\mapsto(a_0,a_1,a_\star))} \\
[\![t\, u]\!]_\rho & \triangleq & [\![t]\!]_\rho\, \rho_0(u)\, \rho_1(u)\, [\![u]\!]_\rho \\
 & & \\
[\![\Pi a : A.\, B]\!]_\rho & \triangleq & \lambda f_0 : (\Pi a_0 : \rho_0(A).\, (\rho_0, a \mapsto a_0)(B)).\, \lambda f_1 : (\Pi a_1 : \rho_1(A).\, (\rho_1, a \mapsto a_1)(B)). \\
 & & \Pi a_0 : \rho_0(A).\, \Pi a_1 : \rho_1(A).\, \Pi a_\star : [\![A]\!]_\rho\, a_0\, a_1.\, [\![B]\!]_{\rho,(a\mapsto(a_0,a_1,a_\star))}\, (f_0\, a_0)\, (f_1\, a_1) \\
[\![U_0]\!]_\rho & \triangleq & \lambda A_0 : U_0.\, \lambda A_1 : U_0.\, A_0 \to A_1 \to U_0) \\
 & & \\
[\![\square]\!] & \triangleq & \square \\
[\![\Gamma, a : A]\!] & \triangleq & [\![\Gamma]\!], a_0 : \rho_0(A), a_1 : \rho_1(A), a_\star : [\![A]\!]_{id_\Gamma}\, a_0\, a_1 \\
 & & \\
id_\square & \triangleq & \emptyset \\
id_{\Gamma,a:A} & \triangleq & id_\Gamma, (a \mapsto (a_0, a_1, a_\star))
\end{array}$$

Parametricity translation

Adequacy/soundness theorem:

$$\begin{pmatrix} \Gamma \text{ wf} \\ \Gamma \vdash t : A \\ \Gamma \vdash t \equiv u : A \\ \Gamma \vdash A : U_0 \\ \Gamma \vdash A \equiv B : U_0 \end{pmatrix} \text{ implies } \begin{pmatrix} [\![\Gamma]\!] \text{ wf} \\ [\![\Gamma]\!] \vdash [\![t]\!]_{id_\Gamma} : [\![A]\!]_{id_\Gamma}\, (id_\Gamma)_0(t)\, (id_\Gamma)_1(t) \\ [\![\Gamma]\!] \vdash [\![t]\!]_{id_\Gamma} \equiv [\![u]\!]_{id_\Gamma} : [\![A]\!]_{id_\Gamma}\, (id_\Gamma)_0(t)\, (id_\Gamma)_1(t) \\ [\![\Gamma]\!] \vdash [\![A]\!]_{id_\Gamma} : (id_\Gamma)_0(A) \to (id_\Gamma)_1(A) \to U_0 \\ [\![\Gamma]\!] \vdash [\![A]\!]_{id_\Gamma} \equiv [\![B]\!]_{id_\Gamma} : (id_\Gamma)_0(A) \to (id_\Gamma)_1(A) \to U_0 \end{pmatrix}$$

Parametricity translation

There are several ways to interpret Martin-Löf's identity type. In a first step, we take them as themselves (i.e. in a "non-parametric" way):

$$
\begin{array}{lcl}
[\![\mathsf{refl}_A\, t]\!]_\rho & \triangleq & (\mathsf{refl}_{\rho_0(A)}\, \rho_0(t), \mathsf{refl}_{\rho_1(A)}\, \rho_1(t)) \\
[\![\mathsf{subst}^{a.b.P}\, p\, \mathsf{in}\, v]\!]_\rho & \triangleq & \mathsf{subst}^{a_{0\Box}.b_{0\Box}.([\![P]\!]_{\rho^0}\, (\mathsf{subst}^{a_0.b_0.\rho'_0(P)}\, b_{0\Box}.0\, \mathsf{in}\, \rho_0(v))\, (\mathsf{subst}^{a_1.b_1.\rho'_1(P)}\, [\![p]\!]_\rho.2\, \mathsf{in}\, \rho_1(v)))}\, [\![p]\!]_\rho.1\, \mathsf{in} \\
 & & \mathsf{subst}^{a_{1\Box}.b_{1\Box}.([\![P]\!]_{\rho^1}\, (\mathsf{subst}^{a_0.b_0.\rho'_0(P)}\, b_{0\Box}.0\, \mathsf{in}\, \rho_0(v))\, (\mathsf{subst}^{a_1.b_1.\rho'_1(P)}\, b_{1\Box}.1\, \mathsf{in}\, \rho_1(v)))}\, [\![p]\!]_\rho.2\, \mathsf{in}\, [\![v]\!]_\rho \\
[\![t =_A u]\!]_\rho & \triangleq & \lambda p_0 : \rho_0(t) =_{\rho_0(A)} \rho_0(u).\, \lambda p_1 : \rho_1(t) =_{\rho_1(A)} \rho_1(u) \rho. \\
 & & \rho_0(t) =_{\rho_0(A)} \rho_0(u) \wedge \rho_1(t) =_{\rho_1(A)} \rho_1(u)
\end{array}
$$

where we wrote ρ'_0 for $(\rho_0, a \mapsto a_0, b \mapsto b_0)$, ρ'_1 for $(\rho_1, a \mapsto a_1, b \mapsto b_1)$, ρ^0 for $(\rho, a \mapsto a_{0\Box}, b \mapsto b_{0\Box})$ and ρ^1 for $(\rho, a \mapsto a_{1\Box}, b \mapsto b_{1\Box})$, as well as writing $.1$ and $.2$ for the projections of a pair.

The translation of $\mathsf{subst}^{a.b.P}\, p\, \mathsf{in}\, v$ is however difficult and painful to follow, so we remind the corresponding typing rule so as to be able to check that it works for those interested in tedious details:

$$
\frac{\Gamma \vdash p : t =_A u \qquad \Gamma, a : A, b : t =_A a \vdash P : U_0 \qquad \Gamma \vdash v : P[t/a][\mathsf{refl}\, t/b]}{\Gamma \vdash \mathsf{subst}^{a.b.P}\, p\, \mathsf{in}\, v : P[u/a][p/b]}
$$

Parametricity translation

We could add Σ-types:

$$\begin{array}{lcl}
[\![(t,u)]\!]_\rho & \triangleq & ([\![t]\!]_\rho, [\![u]\!]_\rho) \\
[\![t.1]\!]_\rho & \triangleq & [\![t]\!]_\rho.1 \\
[\![t.2]\!]_\rho & \triangleq & [\![t]\!]_\rho.2
\end{array}$$

$$\begin{array}{lcl}
[\![\Sigma a : A.B]\!]_\rho & \triangleq & \lambda c_0 : \Sigma a_0 : \rho_0(A).(\rho_0, a \mapsto a_0)(B). \\
 & & \lambda c_1 : \Sigma a_1 : \rho_1(A).(\rho_1, a \mapsto a_1)(B). \\
 & & \Sigma a_\star : ([\![A]\!]_\rho\, c_0.1\, c_1.1).[\![B]\!]_{\rho, a \mapsto (c_0.1, c_1.1, a_\star)}\, c_0.2\, c_1.2
\end{array}$$

Iterated parametricity translation

We could iterated the parametricity translation. For instance, by applying twice parametricity to $A : U_0, a : A \vdash a : A$, we obtain:

$$\begin{array}{lll}
A_{00} : U_0 & A_{10} : U_0 & A_{\star 0} : A_{00} \to A_{10} \to U_0 \\
A_{01} : U_0 & A_{11} : U_0 & A_{\star 1} : A_{01} \to A_{11} \to U_0 \\
A_{0\star} : A_{00} \to A_{01} \to U_0 & A_{1\star} : A_{10} \to A_{11} \to U_0 & A_{\star\star} : \begin{array}{l} \Pi a_{00} : A_{00}. \Pi a_{01} : A_{01}. A_{0\star}\, a_{00}\, a_{01} \to \\ \Pi a_{10} : A_{10}. \Pi a_{11} : A_{11}. A_{1\star}\, a_{10}\, a_{11} \to \\ A_{\star 0}\, a_{00}\, a_{10} \to A_{\star 1}\, a_{01}\, a_{11} \to U_0 \end{array}
\end{array}$$

$$\begin{array}{lll}
a_{00} : A_{00} & a_{10} : A_{10} & a_{\star 0} : A_{\star 0}\, a_{00}\, a_{10} \\
a_{01} : A_{01} & a_{11} : A_{11} & a_{\star 1} : A_{\star 1}\, a_{01}\, a_{11} \\
a_{0\star} : A_{0\star}\, a_{00}\, a_{01} & a_{1\star} : A_{1\star}\, a_{10}\, a_{11} & a_{\star\star} : A_{\star\star}\, a_{00}\, a_{01}\, a_{0\star}\, a_{10}\, a_{11}\, a_{1\star}\, a_{\star 0}\, a_{\star 1}
\end{array}$$

$$\vdash a_{\star\star} : A_{\star\star}\, a_{00}\, a_{01}\, a_{0\star}\, a_{10}\, a_{11}\, a_{1\star}\, a_{\star 0}\, a_{\star 1}$$

$$\begin{tikzcd}
A_{00} \arrow[r, "A_{\star 0}"] \arrow[d, "A_{0\star}"'] \arrow[dr, phantom, "\overset{A_{\star\star}}{\Rrightarrow}"] & A_{10} \arrow[d, "A_{1\star}"] & , & a_{00} \arrow[r, "a_{\star 0}"] \arrow[d, "a_{0\star}"'] \arrow[dr, phantom, "\overset{a_{\star\star}}{\Rrightarrow}"] & a_{10} \arrow[d, "a_{1\star}"] & \vdash a_{\star\star} \\
A_{01} \arrow[r, "A_{\star 1}"'] & A_{11} & & a_{01} \arrow[r, "a_{\star 1}"'] & a_{11} &
\end{tikzcd}$$

Setoid translation, validating functional extensionality

The next step will be to interpret a type not only be a relation, but by an equivalence relation. This will eventually allows to validate functional extensionality.