Translating a Dependently-Typed Logic to First-Order Logic

Kristina Sojakova, Florian Rabe

Jacobs University Bremen

Abstract. DFOL is a logic that extends first-order logic with dependent types. We give a translation from DFOL to FOL formalized as an institution comorphism and show that it admits the model expansion property. This property together with the borrowing theorem implies the soundness of borrowing — a result that enables us to reason about entailment in DFOL by using automated tools for FOL. In addition, the translation permits us to deduce properties of DFOL such as completeness, compactness, and existence of free models from the corresponding properties of FOL, and to regard DFOL as a fragment of FOL. We give an example that shows how problems about DFOL can be solved by using the automated FOL prover Vampire. Future work will focus on the integration of the translation into the specification and translation tool HeTS.

1 Introduction and Related Work

Dependent type theory, DTT, ([ML75]) provides a very elegant language for many applications ([HHP93,NPS90]). However, its definition is much more involved than that of simple type theory because all well-formed terms, types, and their equalities must be defined in a single joint induction. Several quite complex model classes, mainly related to locally cartesian closed categories, have been studied to provide a model theory for DTT (see [Pit00] for an overview).

Many of the complications disappear if dependently-typed extensions of first-order logic are considered, i.e., systems that have dependent types, but no (simple or dependent) function types. Such systems were investigated in [Mak97], [Rab06], and [Bel08]. They provide very elegant axiomatizations of many important mathematical theories such as those of categories or linear algebra while retaining completeness with respect to straightforward set-theoretic models.

However, these systems are of relatively little practical use because no automated reasoning tools, let alone efficient ones, are available. Therefore, our motivation is to translate one of these systems into first-order logic, FOL. Such a translation would translate a proof obligation to FOL and discharge it by calling existing FOL provers. This is called borrowing ([CM93]).

In principle, there are two ways how to establish the soundness of borrowing: proof-theoretically by translating the obtained proof back to the original logic, or model-theoretically by exhibiting a model-translation between the two logics. Proof-theoretical translations of languages with dependent types have been used in [JM93] to translate parts of DTT to simple type theory, in [Urb03] to translate Mizar ([TB85]) into FOL, and in Scunak [Bro06] to translate parts of DTT into FOL. The Scunak translation is only partial as for example the translation of lambda expressions is omitted. Similar partial translations, but in the simply-typed case, are used in Omega ([BCF+97]), Leo-II ([BPTF07]) and in the sledgehammer tactic of Isabelle ([Pau94]). If the FOL prover succeeds, the reconstruction of the FOL proof term is possible in practice but somewhat tricky: For example, sledgehammer uses the output of a strong prover to guide a second, weaker prover, from whose output the proof term is reconstructed. In the cases of Mizar and Scunak, it is not done at all. Furthermore, the more complex the translation of proof goals is, the more difficult it becomes to translate the FOL proof term back into the original logic.

Here we take the model-theoretic approach and formulate a translation from the system introduced in [Rab06] to FOL within the framework of institutions ([GB92]). Mathematically, our main results can be summarized as follows. We use the institution DFOL as given in [Rab06] and give an institution comorphism from DFOL into FOL. Every DFOL-signature is translated to a FOL-theory whose axioms are used to express the typing properties of the translated symbols. The signature translation uses an n+1-ary FOL-predicate P_s for every dependent type constructor s with n arguments. Then the formulas quantifying over x of type $s(t_1, \ldots, t_n)$ can be translated by relativizing (see [Obe62]) using the predicate $P_s(t_1, \ldots, t_n, x)$. Finally, we show that this comorphism admits model expansion. Using the borrowing theorem ([CM93]), this yields the soundness of the translation.

Thus, we provide a simple way to write problems in the conveniently expressive DFOL syntax and solve them by calling FOL theorem provers. It is also possible to extend FOL theorem provers with dependently typed input languages, or to integrate DFOL seamlessly into existing implementations of institution-based algebraic specification languages such as OBJ ([GWM+93]) and CASL ([ABK+02]). Finally, our result provides easier proofs of the free model and completeness theorems given in [Rab06].

2 Definitions

We now present some definitions necessary for our work. We assume that the reader is familiar with the basic concepts of category theory and logic. For introduction to category theory, see [Lan98].
Using categories and functors we can define an *institution*, which is a formalization of a logical system abstracting from notions such as formulas, models, and satisfaction. Institutions structure the variety of different logics and allow us to formulate institution-independent theorems for the general theory of logic. For more on institutions, see [GB92].

Definition 1 (Institution). *An institution is a 4-tuple* $(Sig, Sen, Mod, \models)$ *where*

– *Sig is a category,*
– *$Sen : Sig \to Set$ is a functor,*
– *$Mod : Sig \to Cat^{op}$ is a functor,*
– *$\models$ is a family of relations $\models_\Sigma$ for $\Sigma \in |Sig|$, $\models_\Sigma \subseteq Sen(\Sigma) \times |Mod(\Sigma)|$*

such that for each morphism $\sigma : \Sigma \to \Sigma'$, sentence $F \in Sen(\Sigma)$, and model $M' \in |Mod(\Sigma')|$ we have

$$Mod(\sigma)(M') \models_\Sigma F \quad \textit{iff} \quad M' \models_{\Sigma'} Sen(\sigma)(F)$$

The category Sig is called the *category of signatures*. The morphisms in Sig are called *signature morphisms* and represent notation changes. The functor Sen assigns to each signature Σ a set of *sentences* over Σ and to each morphism $\sigma : \Sigma \to \Sigma'$ the induced *sentence translation* along σ. Similarly, the functor Mod assigns to each signature Σ a category of *models* for Σ and to each morphism $\sigma : \Sigma \to \Sigma'$ the induced *model reduction* along σ. For a signature Σ, the relation $\models_\Sigma$ is called a *satisfaction relation*.
We now define what *entailment* and *theory* are in the context of institutions.

Definition 2 (Entailment). *Let $(Sig, Sen, Mod, \models)$ be an institution. For a fixed Σ, let $T \subseteq Sen(\Sigma)$ and $F \in Sen(\Sigma)$. Then we say that T entails F, denoted $T \models_\Sigma F$, if for any model $M \in |Mod(\Sigma)|$ we have that*

$$\textit{if } M \models_\Sigma G \textit{ for all } G \in T \textit{ then } M \models_\Sigma F$$

Definition 3 (Category of theories). *Let $I = (Sig, Sen, Mod, \models)$ be an institution. We define the category of theories of I to be the category Th^I where*

– *The objects are pairs (Σ, T), with $\Sigma \in |Sig|$, $T \subseteq Sen(\Sigma)$*
– *σ is a morphism from (Σ, T) to (Σ', T') iff σ is a signature morphism from Σ to Σ' in I and for each $F \in T$ we have that $T' \models_\Sigma Sen(\sigma)(F)$*

The objects in Th^I are called *theories* of I, and for each theory $Th = (\Sigma, T)$, the set T is called the set of *axioms* of Th. The morphisms in Th^I are called *theory morphisms*. For a theory (Σ, T) and a sentence F over Σ, we say $(\Sigma, T) \models F$ in place of $T \models_\Sigma F$.
For a given institution I, we sometimes need to construct another institution I^{Th}, whose signatures are the theories of I. We have the following lemma.

Lemma 1 (Institution of theories). *Let $I = (Sig, Sen, Mod, \models)$ be an institution. Denote by I^{Th} the tuple $(Th^I, Sen^{Th}, Mod^{Th}, \models^{Th})$ where*

– *$Sen^{Th}(\Sigma, T) = Sen(\Sigma)$ and $Sen^{Th}(\sigma) = Sen(\sigma)$ for $\sigma : (\Sigma, T) \to (\Sigma', T')$.*
– *$Mod^{Th}(\Sigma, T)$ is the full subcategory of $Mod(\Sigma)$ whose objects are those models M in $|Mod(\Sigma)|$ for which we have $M \models_\Sigma G$ whenever $G \in T$. For a theory morphism $\sigma : (\Sigma, T) \to (\Sigma', T')$, $Mod^{Th}(\sigma)$ is the restriction of $Mod(\sigma)$ to $Mod^{Th}(\Sigma', T')$.*
– *$\models^{Th}_{(\Sigma,T)}$ is the restriction of $\models_\Sigma$ to $|Mod^{Th}(\Sigma, T)| \times Sen^{Th}(\Sigma, T)$.*

Then I^{Th} is an institution, called the institution of theories of I.

We are now ready to define a certain kind of translation between two institutions.

Definition 4 (Institution comorphism). *Let $I = (Sig^I, Sen^I, Mod^I, \models^I)$, $J = (Sig^J, Sen^J, Mod^J, \models^J)$ be two institutions. An institution comorphism from I to J is a triple (Φ, α, β) where*

- $\Phi : Sig^I \to Sig^J$ *is a functor,*
- $\alpha : Sen^I \to \Phi; Sen^J$ *is a natural transformation,*
- $\beta : Mod^I \leftarrow \Phi; Mod^J$ *is a natural transformation*

such that for each $\Sigma \in |Sig^I|$, $F \in Sen^I(\Sigma)$, and $M' \in |Mod^J(\Phi(\Sigma))|$ we have

$$\beta_\Sigma(M') \models^I_\Sigma F \quad \textit{iff} \quad M' \models^J_{\Phi(\Sigma)} \alpha_\Sigma(F)$$

where β_Σ is regarded as a morphism from $Mod^J(\Phi(\Sigma))$ to $Mod^I(\Sigma)$ in the category Cat.

Institution comorphisms are particularly useful if they have the following property.

Definition 5 (Model expansion property). *Let (Φ, α, β) be an institution comorphism from I to J. We say that the comorphism has the model expansion property if each functor β_Σ for $\Sigma \in Sig^I$ is surjective on objects.*

The following lemma is then applicable.

Lemma 2 (Borrowing). *Let (Φ, α, β) be an institution comorphism from I^{Th} to J^{Th} having the model expansion property. Then for any theory (Σ, T) in I and a sentence F over Σ, we have that*

$$(\Sigma, T) \models^I F \quad \textit{iff} \quad \Phi(\Sigma, T) \models^J \alpha_\Sigma(F)$$

In other words, we can use the institution J to reason about theories in I. For more on borrowing, see [CM93].

3 DFOL and FOL as Institutions

The formal definition of a dependent type theory is typically very complex and long because both for the syntax and for the semantics a joint induction over signatures, contexts, terms, and types must be used. Therefore, in [Rab06], the syntax of DFOL is defined within the Edinburgh logical framework (LF, [HHP93]), thus saving one induction. In [Rab08], a model theory for LF is given so that both inductions can be done once and for all in the logical framework, thus permitting a very elegant and compact definition of DFOL.

Here, to be self-contained, we give the syntax directly, but omit the precise definition of well-formed expressions. Then the semantics is given by a partial interpretation function defined only for well-formed expressions. This has the advantage of making the main concepts intuitively clear while being short and precise.

3.1 Signatures

In DFOL, we have three base types, defined as follows:

$$\mathbf{S} : type \qquad Univ : \mathbf{S} \rightarrow type \qquad o : \ type$$

Here $\mathbf{S}$ is the type of sorts (semantically: names of universes). The type $Univ$ is an operator assigning to each sort the type of its terms (semantically: its universe of individuals). The type o is the type of formulas (semantically: the values $true$ and $false$).

A DFOL signature consists of a finite sequence of declarations of the form

$$c : \mathit{\Pi} x_1 : Univ(S_1), \ldots, \mathit{\Pi} x_n : Univ(S_n).\ T$$

meaning that c is a function taking n arguments of types $S_1, \ldots, S_n$ respectively, and returning an argument of type T, where T is one of the three base types. Here $\mathit{\Pi} x_i : Univ(S_i)$ denotes the domain of a dependent function type, i.e., x_i may occur in $S_{i+1}, \ldots, S_n, T$.

When the return type of c is o, we say that c is a *predicate symbol*. Likewise, if the return type is $\mathbf{S}$ or $Univ(S)$, we say that c is a *sort symbol* or a *function symbol* respectively. We abbreviate $\mathit{\Pi} x : Univ(S)$ as $\mathit{\Pi} x : S$ and $\mathit{\Pi} x : A.\ B$ as $A \rightarrow B$ if the variable x does not occur in B.

We define DFOL signatures $\mathit{\Sigma}$ inductively on the number of declarations. Let $\mathit{\Sigma}_k$ be a DFOL signature consisting of k declarations, $k \geq 0$. We define a *function* over $\mathit{\Sigma}_k$ as follows:

– Any variable symbol is a function over $\mathit{\Sigma}_k$
– If f in $\mathit{\Sigma}_k$ is a function symbol of arity n and $\mu_1, \ldots, \mu_n$ are functions over $\mathit{\Sigma}_k$, then $f(\mu_1, \ldots, \mu_n)$ is a function over $\mathit{\Sigma}_k$

If s in $\mathit{\Sigma}_k$ is a sort symbol of arity n and $\mu_1, \ldots, \mu_n$ are functions over $\mathit{\Sigma}_k$, then $s(\mu_1, \ldots, \mu_n)$ is a *sort* over $\mathit{\Sigma}_k$. Similarly, if p in $\mathit{\Sigma}_k$ is a predicate symbol of arity n and $\mu_1, \ldots, \mu_n$ are functions over $\mathit{\Sigma}_k$, then $p(\mu_1, \ldots, \mu_n)$ is a *predicate* over $\mathit{\Sigma}_k$. The word *term* refers to either a function, a sort, or a predicate.

Clearly, not all terms are well-formed in DFOL. A context $\mathit{\Gamma}$ for a signature $\mathit{\Sigma}$ in DFOL has the form $\mathit{\Gamma} = x_1 : S_1, \ldots, x_n : S_n$, where $S_1, \ldots, S_n$ are sorts and S_i contains no variables except possibly $x_1, \ldots, x_{i-1}$. Given a valid context $\mathit{\Gamma}$, a DFOL term is well-formed with respect to $\mathit{\Gamma}$ only if it is well-typed in the LF type theory. For details we refer the reader to [Rab06].

Now the $k+1$-th declaration has one of the following forms:

– $s : \mathit{\Pi} x_1 : S_1,\ \ldots,\ \mathit{\Pi} x_n : S_n.\ \mathbf{S}$
  where $S_1, \ldots, S_n$ are sorts over $\mathit{\Sigma}_k$ and S_i contains no variables except possibly $x_1, \ldots, x_{i-1}$. We say that s is a sort symbol.
– $f : \mathit{\Pi} x_1 : S_1,\ \ldots,\ \mathit{\Pi} x_n : S_n.\ S_{n+1}$
  where $S_1, \ldots, S_{n+1}$ are sorts over $\mathit{\Sigma}_k$ and S_i contains no variables except possibly $x_1, \ldots, x_{i-1}$.

– $p : \Pi x_1 : S_1, \ \ldots, \ \Pi x_n : S_n. \ o$
where $S_1, \ldots, S_n$ are sorts over Σ_k and S_i contains no variables except possibly $x_1, \ldots, x_{i-1}$. We say that p is a predicate symbol.

As with terms, the declaration must be a well-typed according to the rules of LF.

Running example The theory of categories has the following DFOL signature
$Ob : \mathbf{S}$
$Mor : Ob \ \rightarrow \ Ob \ \rightarrow \ \mathbf{S}$
$id : \Pi A : Ob. \ Mor(A, A)$
$\circ : \Pi A, B, C : Ob. \ Mor(A, B) \ \rightarrow \ Mor(B, C) \ \rightarrow \ Mor(A, C)$
$term : Ob \ \rightarrow \ o$
$isom : Ob \ \rightarrow \ Ob \ \rightarrow \ o$ For simplicity, we declare the signature model morphisms in DFOL to just be the identity morphisms.

3.2 Sentences

The set of DFOL formulas over a signature Σ can be described as follows:

– If P is a predicate over Σ, then P is a Σ-formula
– If μ_1, μ_2 are functions over Σ, then $\mu_1 \doteq \mu_2$ is a Σ-formula
– If F is a Σ-formula, then $\neg F$ is a Σ-formula
– If F, G are Σ-formulas, then $F \wedge G$, $F \vee G$, and $F \Rightarrow G$ are Σ-formulas
– If F is a Σ-formula and S is a sort term over Σ, then $\forall x : S. \ F$ is a Σ-formula
– If F is a Σ-formula and S is a sort term over Σ, then $\exists x : S. \ F$ is a Σ-formula

Closed and atomic formulas are defined in the obvious way analogous to first-order logic. As with terms, DFOL formulas are well-formed only if they are well-typed in the LF type theory. For a precise definition, see [Rab06].

Running example We have the following axioms for the theory of categories, with equivalence defined as usual
$I1 : \forall A, B : Ob. \ \forall f : Mor(A, B). \ id(A) \circ f \doteq f$
$I2 : \forall A, B : Ob. \ \forall f : Mor(B, A). \ f \circ id(A) \doteq f$
$A1 : \forall A, B, C, D : Ob. \ \forall f : Mor(A, B). \ \forall g : Mor(B, C). \ \forall h : Mor(C, D).$
$f \circ (g \circ h) \doteq (f \circ g) \circ h$
$D1 : \forall A : Ob. \ (term(A) \iff \forall B : Ob. \ \exists f : Mor(B, A). \ \forall g : Mor(B, A). \ f \doteq g)$
$D2 : \forall A, B : Ob. \ (isom(A, B) \iff \exists f : Mor(A, B). \ \exists g : Mor(B, A).$
$(f \circ g \doteq id(A) \ \wedge \ g \circ f \doteq id(B)))$

3.3 Models

A model of a DFOL signature Σ is an interpretation function I. Since the declaration of a symbol may depend on symbols declared before, we define I inductively on the number of declarations.
Suppose I is defined for the first k declarations, $k \geq 0$. An *assignment function*

φ for I is a function mapping each variable to an element of any set defined by I as an interpretation of a sort symbol.
Let μ be a term over Σ. We define the *interpretation* of μ induced by φ to be $I_\varphi(\mu)$, where I_φ is given by:

- $I_\varphi(x) = \varphi(x)$ for any variable x
- $I_\varphi(d(\mu_1, \ldots, \mu_k)) = d^I(I_\varphi(\mu_1), \ldots, I_\varphi(\mu_k))$ for a sort, predicate, or function symbol d if
  - each of the interpretations $I_\varphi(\mu_1), \ldots, I_\varphi(\mu_n)$ exists and
  - d^I is defined for the tuple $(I_\varphi(\mu_1), \ldots, I_\varphi(\mu_n))$

  Otherwise we say $I_\varphi(d(\mu_1, \ldots, \mu_k))$ does not exist.

Given a valid context $\Gamma = x_1 : S_1, \ldots, x_n : S_n$ and an assignment function φ, we say that φ is an *assignment function for* Γ if for each i we have that $I_\varphi(S_i)$ exists and $\varphi(x_i) \in I_\varphi(S_i)$. From now on, we will only talk about assignment functions for a context; the general definition was introduced only to avoid some technical difficulties.
Now the $k+1$-st declaration has one of the following forms:

- $s : \Pi x_1 : S_1,\ \ldots,\ \Pi x_n : S_n.\ \mathbf{S}$
  Let φ be any assignment function for the context $\Gamma = x_1 : S_1, \ldots, x_n : S_n$. Then

  $$s^I(\varphi(x_1), \ldots, \varphi(x_n)) \text{ is a (possibly empty) set}$$

  disjoint from any other set defined by I as an interpretation of a sort symbol.
- $f : \Pi x_1 : S_1,\ \ldots,\ \Pi x_n : S_n.\ S$
  Let φ be any assignment function for the context $\Gamma = x_1 : S_1, \ldots, x_n : S_n$. Then

  $$f^I(\varphi(x_1), \ldots, \varphi(x_n)) \in I_\varphi(S)$$

- p is a predicate symbol, $p : \Pi x_1 : S_1,\ \ldots,\ \Pi x_n : S_n.\ o$
  Let φ be any assignment function for the context $\Gamma = x_1 : S_1, \ldots, x_n : S_n$. Then

  $$p^I(\varphi(x_1), \ldots, \varphi(x_n)) \in \{true, false\}$$

Running example An example model I for the signature of categories is given by any small category C. Then we have $Ob^I = |C|$, $Mor^I(A, B) = C(A, B)$, and the obvious interpretations for composition and identity. Furthermore, we can put $term^I(A) = true$ iff A is a terminal element and $isom^I(A, B) = true$ iff A and B are isomorphic.

3.4 Satisfaction relation

To define the satisfaction relation, we first define the interpretation of formulas. Let Σ be a DFOL signature, I be a DFOL model for Σ, and Γ be a valid context over Σ. Furthermore, let φ be an assignment function for Γ and F be a well-formed DFOL formula for Γ. Then we define $I_\varphi(F)$ recursively on the structure of F:

- F is a predicate. Then $I_\varphi(F)$ is true if and only if $p^I(I_\varphi(\mu_1), \ldots, I_\varphi(\mu_n)) = true$.
- F is of the form $\mu_1 \doteq \mu_2$. Then $I_\varphi(F)$ is true if and only if $I_\varphi(\mu_1) = I_\varphi(\mu_2)$.
- F is of the form $\neg G$. Then $I_\varphi(F)$ is true if and only if $I_\varphi(G)$ is false.
- F is of the form $F_1 \wedge F_2$. Then $I_\varphi(F)$ is true if and only if both $I_\varphi(F_1)$ and $I_\varphi(F_2)$ are true.
- F is of the form $F_1 \vee F_2$. Then $I_\varphi(F)$ is true if and only if $I_\varphi(F_1)$ is true or $I_\varphi(F_2)$ is true.
- F is of the form $F_1 \implies F_2$. Then $I_\varphi(F)$ is true if and only if $I_\varphi(F_1)$ is false or $I_\varphi(F_2)$ is true.
- F is of the form $\exists x : S.\ G$. Then $I_\varphi(F)$ is true if and only if $I_{\varphi[x/a]}(G)$ is true for some $a \in I_\varphi(S)$.
- F is of the form $\forall x : S.\ G$. Then $I_\varphi(F)$ is true if and only if $I_{\varphi[x/a]}(G)$ is true for any $a \in I_\varphi(S)$.

Now if F is in fact a closed formula, its interpretation is independent of φ. Hence, we define that I satisfies F if and only if $I_\varphi(F)$ is true for some φ.

Running example It is easy to see that the example model for the signature of categories satisfies the axioms given in section 3.2.

Putting our previous definitions together, we have the following lemma.

Lemma 3. *$DFOL = (Sig, Sen, Mod, \models)$ is an institution.*

The FOL institution is then obtained from DFOL by restricting the signatures to contain a unique sort symbol, having arity 0. Any other symbols are either function or predicate symbols. (Technically, this does not yield FOL because DFOL permits empty universes. But our FOL signatures will always have a nullary function symbol so that this does not constitute a problem). A FOL model is then denoted as (U, I), where I is the interpretation function and U is the universe corresponding to the unique sort symbol.

4 Translation of DFOL to FOL

The main idea of the translation is to associate with each n-ary sort symbol in DFOL an $n+1$-ary predicate in FOL and relativize the universal and existential quantifiers (the technique of relativization was first introduced by Oberschelp in [Obe62]).

Formally, the translation will be given as an institution comorphism from $DFOL$ to FOL^{Th}. We specify a functor Φ, mapping DFOL signatures to FOL theories and DFOL signature morphisms to FOL theory morphisms. For each DFOL signature Σ, we give a function α_Σ mapping DFOL sentences over Σ to FOL sentences over the translated signature $\Phi(\Sigma)$, and show that the family of functions α_Σ defines a natural transformation. Similarly, for each DFOL signature Σ we give a functor β_Σ mapping FOL models for the translated signature $\Phi(\Sigma)$ to DFOL models for Σ, and show that the family of functors β_Σ defines a natural transformation. Finally, we prove the satisfaction condition for (Φ, α, β) and show that the comorphism has the model expansion property.

Definition 6 (Signature translation). *Let Σ be a DFOL signature. We define $\Phi(\Sigma)$ to be the FOL theory (Σ', T'), where Σ' and T' are specified as follows. Σ' contains:*

- *an n-ary function symbol f for each n-ary function symbol f in Σ,*
- *an n-ary predicate symbol p for each n-ary predicate symbol p in Σ,*
- *an $n+1$-ary predicate symbol s for each n-ary sort symbol s in Σ,*
- *a special constant symbol $\perp$, different from any of the above symbols,*
- *no other symbols besides the above*

T' contains:

S1. Axioms ensuring that no element can belong to the universe of more than one sort. For any two sort symbols s_1, s_2 with s_1 different from s_2, we have the axiom

$$\forall x_1, \ldots, x_n, y_1, \ldots, y_m, z. \ \big(s_1(x_1, \ldots, x_n, z) \implies \neg s_2(y_1, \ldots, y_m, z)\big)$$

and for each sort symbol s_1 we have the axiom

$$\forall x_1, \ldots, x_n, y_1, \ldots, y_n, z. \ \big(s_1(x_1, \ldots, x_n, z) \ \wedge \ s_1(y_1, \ldots, y_n, z) \\ \implies x_1 \doteq y_1 \ \wedge \ \ldots \ \wedge \ x_n \doteq y_n\big)$$

S2. An axiom ensuring that each element different from $\perp$ belongs to the universe of at least one sort. If $s_1, \ldots, s_k$ are the sort symbols, then we have the axiom

$$\forall y. \ \big(\neg y \doteq \perp \implies \exists x_1, \ldots, x_{n_1}. \ s_1(x_1, \ldots, x_{n_1}, y) \ \vee \ \ldots \ \vee \\ \exists x_1, \ldots, x_{n_k}. \ s_k(x_1, \ldots, x_{n_k}, y)\big)$$

S3. Axioms ensuring that the special symbol $\perp$ is not contained in the universe of any sort. For each sort symbol s, we have the axiom

$$\forall x_1, \ldots, x_n. \ \neg s(x_1, \ldots, x_n, \perp)$$

S4. Axioms ensuring that if the arguments to a sort constructor are not of the correct types, the resulting sort has an empty universe. For each sort symbol $s : \Pi x_1 : s_1(\mu_1^1, \ldots, \mu_{k_1}^1), \ \ldots, \ \Pi x_n : s_n(\mu_1^n, \ldots, \mu_{k_n}^n). \ \boldsymbol{S}$, we have the axiom

$$\forall x_1, \ldots, x_n. \ \big(\neg s_1(\mu_1^1, \ldots, \mu_{k_1}^1, x_1) \ \vee \ \ldots \ \vee \neg s_n(\mu_1^n, \ldots, \mu_{k_n}^n, x_n) \\ \implies \forall y. \ \neg s(x_1, \ldots, x_n, y)\big)$$

F1. Axioms ensuring that if the arguments to a function are of the correct types, the function returns a value of the correct type. For each function symbol $f : \Pi x_1 : s_1(\mu_1^1, \ldots, \mu_{k_1}^1), \ \ldots, \ \Pi x_n : s_n(\mu_1^n, \ldots, \mu_{k_n}^n). \ s(\mu_1, \ldots, \mu_k)$, we have the axiom

$$\forall x_1, \ldots, x_n. \ \big(s_1(\mu_1^1, \ldots, \mu_{k_1}^1, x_1) \ \wedge \ \ldots \ \wedge s_n(\mu_1^n, \ldots, \mu_{k_n}^n, x_n) \implies \\ s(\mu_1, \ldots, \mu_k, f(x_1, \ldots, x_n))\big)$$

F2. Axioms ensuring that if the arguments to a function are not of the correct types, the function returns the special symbol $\bot$. For each function symbol $f : \Pi x_1 : s_1(\mu^1_1, \ldots, \mu^1_{k_1}),\ \ldots,\ \Pi x_n : s_n(\mu^n_1, \ldots, \mu^n_{k_n}).\ s(\mu_1, \ldots, \mu_k)$, we have the axiom

$$\forall x_1, \ldots, x_n. \big(\neg s_1(\mu^1_1, \ldots, \mu^1_{k_1}, x_1) \ \vee\ \ldots\ \vee \neg s_n(\mu^n_1, \ldots, \mu^n_{k_n}, x_n) \\ \implies f(x_1, \ldots, x_n) \doteq \bot\big)$$

P1. Axioms ensuring that if the arguments to a predicate are not of the correct types, the predicate is false. For each predicate symbol $p : \Pi x_1 : s_1(\mu^1_1, \ldots, \mu^1_{k_1}),\ \ldots,\ \Pi x_n : s_n(\mu^n_1, \ldots, \mu^n_{k_n}).\ o$, we have the axiom

$$\forall x_1, \ldots, x_n. \big(\neg s_1(\mu^1_1, \ldots, \mu^1_{k_1}, x_1) \ \vee\ \ldots\ \vee \neg s_n(\mu^n_1, \ldots, \mu^n_{k_n}, x_n) \\ \implies \neg p(x_1, \ldots, x_n)\big)$$

N. No other axioms besides the above

Defining Φ on signature morphisms is trivial since by our definition the only signature morphisms in DFOL are the identity morphisms. From this it follows immediately that Φ is a functor.

Running example Denote the translated signature of categories by the theory (Σ', T'). Then Σ' contains the following symbols:

- Function symbols:
  - id of arity 1
  - $\circ$ of arity 5
  - $\bot$ of arity 0
- Predicate symbols:
  - ob of arity 1
  - mor of arity 3
  - term of arity 1
  - isom of arity 2

The theory T' consists of the axioms $S1.1$ up to $P1.2$ in Fig.1.

Definition 7 (Sentence translation). *Let Σ be a DFOL signature. We define the function α_Σ on the set of all DFOL formulas over Σ. We do this recursively on the structure of the formula F:*

- *If F is of the form $p(\mu_1, \ldots, \mu_n)$, we set $\alpha_\Sigma(F) = F$*
- *If F is of the form $\mu_1 \doteq \mu_2$, we set $\alpha_\Sigma(F) = F$*
- *If F is of the form $\neg G$, we set $\alpha_\Sigma(F) = \neg\alpha_\Sigma(G)$*
- *If F is of the form $F_1 \wedge F_2$, we set $\alpha_\Sigma(F) = \alpha_\Sigma(F_1) \wedge \alpha_\Sigma(F_2)$*
- *If F is of the form $F_1 \vee F_2$, we set $\alpha_\Sigma(F) = \alpha_\Sigma(F_1) \vee \alpha_\Sigma(F_2)$*
- *If F is of the form $F_1 \implies F_2$, we set $\alpha_\Sigma(F)$ to be the formula*

$$\alpha_\Sigma(F_1) \implies \alpha_\Sigma(F_2)$$

– *If F is of the form $\forall x : s(\mu_1, \ldots, \mu_n).\, G$, we set $\alpha_\Sigma(F)$ to be the formula*

$$\forall x.\ s(\mu_1, \ldots, \mu_n, x) \implies \alpha_\Sigma(G)$$

– *If F is of the form $\exists x : s(\mu_1, \ldots, \mu_n).\, G$, we set $\alpha_\Sigma(F)$ to be the formula*

$$\exists x.\ s(\mu_1, \ldots, \mu_n, x) \wedge \alpha_\Sigma(G)$$

It is easy to see that α_Σ maps closed formulas to closed formulas. Hence, we can restrict α_Σ to the set of DFOL sentences over Σ to obtain our desired translation map. The naturality of α_Σ follows immediately since the only signature morphisms in DFOL are the identity morphisms.

Running example The translated axioms of the theory of categories are the axioms $I1$ up to $D2$ in Fig.1.

$S1.1 : \forall A_1, B_1, f.\ (mor(A_1, B_1, f) \implies \forall A_2, B_2.\ (mor(A_2, B_2, f) \implies A_2 \doteq A_1 \ \wedge\ B_2 \doteq B_1))$
$S1.2 : \forall y, B, C.\ (ob(y) \implies \neg mor(B, C, y))$
$S2 : \forall y.\ (\neg y \doteq \bot \implies ob(y) \ \vee\ \exists A, B.\ mor(A, B, y))$
$S3.1 : \neg ob(\bot)$
$S3.2 : \forall A, B.\ \neg mor(A, B, \bot)$
$S4 : \forall A, B.\ ((\neg ob(A) \ \vee\ \neg ob(B)) \implies \forall f.\ \neg mor(A, B, f))$
$F1.1 : \forall A.\ (ob(A) \implies mor(A, A, id(A)))$
$F1.2 : \forall A, B, C, f, g.\ (ob(A) \ \wedge\ ob(B) \ \wedge\ ob(C) \ \wedge\ mor(A, B, f) \ \wedge\ mor(B, C, g) \implies mor(A, C, f; g))$
$F2.1 : \forall A.\ (\neg ob(A) \implies id(A) \doteq \bot)$
$F2.2 : \forall A, B, C, f, g.\ (\neg ob(A) \ \vee\ \neg ob(B) \ \vee\ \neg ob(C) \ \vee\ \neg mor(A, B, f) \ \vee\ \neg mor(B, C, g) \implies f; g \doteq \bot)$
$P1.1 : \forall A.\ (\neg ob(A) \implies \neg term(A))$
$P1.2 : \forall A, B.\ (\neg ob(A) \ \vee\ \neg ob(B) \implies \neg isom(A, B))$

$I1 : \forall A, B, f.\ (ob(A) \ \wedge\ ob(B) \ \wedge\ mor(A, B, f) \implies id(A); f \doteq f)$
$I2 : \forall A, B, f.\ (ob(A) \ \wedge\ ob(B) \ \wedge\ mor(B, A, f) \implies f; id(A) \doteq f)$
$A1 : \forall A, B, C, D, f, g, h.\ (ob(A) \ \wedge\ ob(B) \ \wedge\ ob(C) \ \wedge\ ob(D) \ \wedge\ mor(A, B, f) \ \wedge\ mor(B, C, g) \ \wedge\ mor(C, D, h) \implies f; (g; h) \doteq (f; g); h)$
$D1 : \forall A.\ (ob(A) \implies (term(A) \iff \forall B.\ (ob(B) \implies \exists f.\ (mor(B, A, f) \ \wedge\ \forall g.\ (mor(B, A, g) \implies f \doteq g)))))$
$D2 : \forall A, B.\ (ob(A) \ \wedge\ obj(B) \implies (isom(A, B) \iff \exists f, g.\ (mor(A, B, f) \ \wedge\ mor(B, A, g) \ \wedge\ f; g \doteq id(A) \ \wedge\ g; f \doteq id(B))))$

$Conjecture : \forall A, B.\ (ob(A) \ \wedge\ obj(B) \ \wedge\ term(A) \ \wedge\ term(B) \implies isom(A, B))$

Fig. 1. Translation of the running example

Definition 8 (Model reduction). *Let Σ be a DFOL signature and $M = (U, I)$ be a FOL model for $\Phi(\Sigma)$. We define the translated DFOL model $\beta_\Sigma(M)$ for Σ*

to be the interpretation function J, defined inductively on the number of declarations in Σ.
Suppose J is defined for the first k symbols in Σ, $k \geq 0$. Then the $(k+1)$-st declaration has one of the following forms:

- $s : \Pi x_1 : S_1,\ \ldots,\ \Pi x_n : S_n.\ \boldsymbol{S}$
  Let φ be any assignment function for the context $\Gamma = x_1 : S_1, \ldots, x_n : S_n$. We set
  $$s^J(\varphi(x_1), \ldots, \varphi(x_n)) = \{u \in U \mid s^I(\varphi(x_1), \ldots, \varphi(x_n), u)\}$$
- $f : \Pi x_1 : S_1,\ \ldots,\ \Pi x_n : S_n.\ S$
  Let φ be any assignment function for the context $\Gamma = x_1 : S_1, \ldots, x_n : S_n$. We set
  $$f^J(\varphi(x_1), \ldots, \varphi(x_n)) = f^I(\varphi(x_1), \ldots, \varphi(x_n))$$
- *p is a predicate symbol, $c : \Pi x_1 : S_1,\ \ldots,\ \Pi x_n : S_n.\ o$*
  Let φ be any assignment function for the context $\Gamma = x_1 : S_1, \ldots, x_n : S_n$. We set
  $$p^J(\varphi(x_1), \ldots, \varphi(x_n)) \quad \textit{iff} \quad p^I(\varphi(x_1), \ldots, \varphi(x_n))$$

We note here how the axioms introduced earlier are needed to ensure that J is indeed a DFOL model for Σ. We now turn to the proof of the satisfaction condition.

Theorem 1 (Satisfaction condition). *(Φ, α, β) is an institution comorphism.*

Proof. We have already shown that Φ is a functor and α, β are natural transformations. It remains to show that the satisfaction condition holds.
Let Σ be a DFOL signature, Γ be a valid context for Σ, φ be an assignment function for Γ, and F be a well-formed DFOL sentence for Γ. Furthermore, let $M = (U, I)$ be a FOL model for the translated signature $\Phi(\Sigma)$, and J be the translated model $\beta_\Sigma(M)$. We first observe the following two facts:

- φ is also an assignment function for M
- if μ is a well-formed function term for Γ, then $J_\varphi(\mu) = I_\varphi(\mu)$

Both of these facts follow directly from the construction of J. We now show that we have
$$J_\varphi(F) \quad \text{iff} \quad I_\varphi(\alpha_\Sigma(F))$$
To prove the claim, we proceed recursively on the structure of F:

- F is of the form $p(\mu_1, \ldots, \mu_n)$. Then $J_\varphi(F)$ is true if and only if $p^J(J_\varphi(\mu_1), \ldots, J_\varphi(\mu_n))$. By the construction of J, we have
  $$p^J(J_\varphi(\mu_1), \ldots, J_\varphi(\mu_n)) \quad \text{iff} \quad p^I(J_\varphi(\mu_1), \ldots, J_\varphi(\mu_n))$$
  As noted above, $J_\varphi(\mu_i) = I_\varphi(\mu_i)$ for each i, hence

$$p^J(J_\varphi(\mu_1), \ldots, J_\varphi(\mu_n)) \quad \text{iff} \quad p^I(I_\varphi(\mu_1), \ldots, I_\varphi(\mu_n))$$

Thus we have $J_\varphi(F)$ if and only if $I_\varphi(F)$. Since $F = \alpha_\Sigma(F)$, this proves the claim.

- F is of the form $\mu_1 \doteq \mu_2$. Then $J_\varphi(F)$ is true if and only if $J_\varphi(\mu_1) = J_\varphi(\mu_2)$. As noted above, $J_\varphi(\mu_1) = I_\varphi(\mu_1)$ and $J_\varphi(\mu_2) = I_\varphi(\mu_2)$, hence

$$J_\varphi(\mu_1) = J_\varphi(\mu_2) \quad \text{iff} \quad I_\varphi(\mu_1) = I_\varphi(\mu_2)$$

Thus we have $J_\varphi(F)$ if and only if $I_\varphi(F)$. Since $F = \alpha_\Sigma(F)$, this proves the claim.

- F is of the form $\neg G$. Then $J_\varphi(F)$ is true if and only if $J_\varphi(G)$ is false. By the induction hypothesis, we have $J_\varphi(G)$ iff $I_\varphi(\alpha_\Sigma(G))$. Thus $J_\varphi(F)$ is true if and only if $I_\varphi(\alpha_\Sigma(G))$ is false, or equivalently

$$J_\varphi(F) \quad \text{iff} \quad I_\varphi(\neg\alpha_\Sigma(G))$$

Since $\neg\alpha_\Sigma(G) = \alpha_\Sigma(F)$, this proves the claim.

- F is of the form $F_1 \wedge F_2$. Then $J_\varphi(F)$ is true if and only if both $J_\varphi(F_1)$ and $J_\varphi(F_2)$ are true. By the induction hypothesis, we have $J_\varphi(F_1)$ iff $I_\varphi(\alpha_\Sigma(F_1))$ and $J_\varphi(F_2)$ iff $I_\varphi(\alpha_\Sigma(F_2))$. Hence, $J_\varphi(F)$ is true if and only if both $I_\varphi(\alpha_\Sigma(F_1))$ and $I_\varphi(\alpha_\Sigma(F_2))$ are true. Equivalently,

$$J_\varphi(F) \quad \text{iff} \quad I_\varphi(\alpha_\Sigma(F_1) \wedge \alpha_\Sigma(F_2))$$

Since $\alpha_\Sigma(F_1) \wedge \alpha_\Sigma(F_2) = \alpha_\Sigma(F)$, this proves the claim.

- F is of the form $F_1 \vee F_2$. Since F is equivalent to the formula $\neg(\neg F_1 \wedge \neg F_2)$, the claim follows from the previous steps.
- F is of the form $F_1 \implies F_2$. Since F is equivalent to the formula $\neg F_1 \vee F_2$, the claim follows from the previous steps.
- F is of the form $\exists x : s(\mu_1, \ldots, \mu_n).\ G$. By definition, $J_\varphi(F)$ is true if and only if there exists an $a \in J_\varphi(s(\mu_1, \ldots, \mu_n))$ such that $J_{\varphi[x/a]}(G)$ is true. Again by definition,

$$J_\varphi(s(\mu_1, \ldots, \mu_n)) = s^J(J_\varphi(\mu_1), \ldots, J_\varphi(\mu_n))$$

Since $J_\varphi(\mu_i) = I_\varphi(\mu_i)$ for each i, we have

$$s^J(J_\varphi(\mu_1), \ldots, J_\varphi(\mu_n)) = s^J(I_\varphi(\mu_1), \ldots, I_\varphi(\mu_n))$$

By the construction of J, we have that a belongs to $s^J(I_\varphi(\mu_1), \ldots, I_\varphi(\mu_n))$ if and only if a belongs to U and $s^I(I_\varphi(\mu_1), \ldots, I_\varphi(\mu_n), a) = \mathit{true}$. Now since μ_i does not contain x for any i, we have that

$$s^I(I_\varphi(\mu_1), \ldots, I_\varphi(\mu_n), a) = I_{\varphi[x/a]}(s(\mu_1, \ldots, \mu_n, x))$$

Also, by the induction hypothesis we have that

$$J_{\varphi[x/a]}(G) \quad \text{iff} \quad I_{\varphi[x/a]}(\alpha_\Sigma(G))$$

Combining this, we get precisely that

$$J_\varphi(F) \quad \text{iff} \quad I_\varphi(\exists x.\, s(\mu_1,\ldots,\mu_n,x) \;\wedge\; \alpha_\Sigma(G))$$

Since $\exists x.\, s(\mu_1,\ldots,\mu_n,x) \wedge \alpha_\Sigma(G) = \alpha_\Sigma(F)$, this proves the claim.

– F is of the form $\forall x : s(\mu_1,\ldots,\mu_n).\, G$. Since F is equivalent to the formula $\neg\exists x : s(\mu_1,\ldots,\mu_n).\, \neg G$, the claim follows from the previous steps.

At last, we prove the model expansion property.

Theorem 2 (Model expansion property). *The institution comorphism (Φ,α,β) has the model expansion property.*

Proof. Let Σ be a DFOL signature and J be a DFOL model for Σ. We construct a FOL model $M = (U,I)$ for the translated signature $\Phi(\Sigma)$ such that $J = \beta_\Sigma(M)$.
To define U, let $s_1,\ldots,s_k$ be the sort symbols of Σ. For s_i of arity n_i, set

$$U_i = \bigcup_{(x_1,\ldots,x_{n_i})} s_i(x_1,\ldots,x_{n_i})$$

where $(x_1,\ldots,x_{n_i})$ ranges through all n_i-tuples for which s_i is defined. Set

$$U = \{\bot\} \cup U_1 \cup \ldots \cup U_n$$

We now define I as follows.

– Let p be a predicate symbol in Σ, $p : \Pi x_1 : S_1,\ \ldots,\ \Pi x_n : S_n.\ o$. Let φ be an assignment function for M. If φ is also an assignment function for the context $\Gamma = x_1 : S_1,\ldots,x_n : S_n$, we set

$$p^I(\varphi(x_1),\ldots,\varphi(x_n)) \quad \text{iff} \quad p^J(\varphi(x_1),\ldots,\varphi(x_n))$$

otherwise we set $p^I(\varphi(x_1),\ldots,\varphi(x_n))$ to be false.

– Let f be a function symbol in Σ, $f : \Pi x_1 : S_1,\ \ldots,\ \Pi x_n : S_n.\ S$. Let φ be an assignment function for M. If φ is also an assignment function for the context $\Gamma = x_1 : S_1,\ldots,x_n : S_n$, we set

$$f^I(\varphi(x_1),\ldots,\varphi(x_n)) = f^J(\varphi(x_1),\ldots,\varphi(x_n))$$

otherwise we set $f^I(\varphi(x_1),\ldots,\varphi(x_n)) = \bot$.

– Let s be a sort symbol in Σ, $s : \Pi x_1 : S_1,\ \ldots,\ \Pi x_n : S_n.\ \mathbf{S}$. Let φ be an assignment function for M. If φ is also an assignment function for the context $\Gamma = x_1 : S_1,\ldots,x_n : S_n$, we set

$$s^I(\varphi(x_1),\ldots,\varphi(x_n),\varphi(y)) \quad \text{iff} \quad \varphi(y) \in s^J(\varphi(x_1),\ldots,\varphi(x_n))$$

otherwise we set $s^I(\varphi(x_1),\ldots,\varphi(x_n),\varphi(y)) = false$.

It is easy to see that $M = (U,I)$ satisfies all the axioms in the translated signature $\Phi(\Sigma)$ and that we have $J = \beta_\Sigma(M)$.

Hence, the institution comorphism (Φ,α,β) permits borrowing and we have that a DFOL theory entails a sentence if and only if the translated FOL theory entails the translated sentence.

5 Conclusion and Future Work

We have given an institution comorphism from a dependently-typed logic to FOL and have shown that it admits model expansion. Together with the borrowing theorem [CM93] this implies the soundness of borrowing.

This result is important for several reasons. The need for dependent types arises in several areas of mathematics such as linear algebra and category theory. DFOL provides a more natural way of formulating mathematical problems while staying close to FOL formally and intuitively. On the other hand, for FOL we have machine support in the form of automated theorem-provers and model-finders. The translation enables us to formulate a DFOL problem, translate it to FOL, and then use the known automated methods for FOL (e.g., theorem-provers such as Vampire [RV02] or SPASS [WAB+99], and model finders such as Paradox [CS03]) to find a solution.

First experiments with the translation have proved successful: For example, Vampire was able to prove instantaneously that the translation of our running example is a FOL theorem. It remains to be seen how much the encoding of type information in predicates and the addition of axioms in the translation affects the performance of FOL provers on larger theories. In the future we will integrate our translation into HeTS ([MML07]), a CASL-based application that provides a framework for the implementation of institutions and institution translations. That will provide the infrastructure to create and translate big, structured DFOL theories, and thus to apply our translation on a larger scale.

Since DFOL is defined within LF, we will also treat it as a running example for an implementation of the framework introduced in [Rab08]. That will permit to define arbitrary institutions and institution translations in LF and then incorporate these definitions into HeTS.

On the theoretical side, the translation shows that DFOL can be regarded as a fragment of FOL, which generalizes the well-known results for many-sorted first-order logic. In particular, we are able to derive properties of DFOL such as completeness, compactness, and the existence of free models immediately from the corresponding properties of FOL.

References

[ABK+02] E. Astesiano, M. Bidoit, H. Kirchner, B. Krieg-Brückner, P. Mosses, D. Sannella, and A. Tarlecki. CASL: The Common Algebraic Specification Language. *Theoretical Computer Science*, 2002.

[BCF+97] C. Benzmüller, L. Cheikhrouhou, D. Fehrer, A. Fiedler, X. Huang, M. Kerber, M. Kohlhase, K. Konrad, E. Melis, A. Meier, W. Schaarschmidt, J. Siekmann, and V. Sorge. ΩMEGA: Towards a mathematical assistant. In W. McCune, editor, *Proceedings of the 14th Conference on Automated Deduction*, pages 252–255. Springer, 1997.

[Bel08] J. Belo. Dependently Sorted Logic. In M. Miculan, I. Scagnetto, and F. Honsell, editors, *TYPES 2008*, pages 33–50. Springer, 2008.

[BPTF07] C. Benzmller, L. Paulson, F. Theiss, and A. Fietzke. The LEO-II Project. In *Automated Reasoning Workshop*, 2007.

[Bro06] C. Brown. Combining Type Theory and Untyped Set Theory. In N. Shankar and U. Furbach, editors, *Proceedings of the 3rd International Joint Conference on Automated Reasoning*, pages 205–219. Springer, 2006.

[CM93] M. Cerioli and J. Meseguer. May I Borrow Your Logic? In A. Borzyszkowski and S. Sokolowski, editors, *Mathematical Foundations of Computer Science*, pages 342–351. Springer, 1993.

[CS03] K. Claessen and N. Sorensson. New techniques that improve MACE-style finite model finding. In *19th International Conference on Automated Deduction (CADE-19) Workshop on Model Computation - Principles, Algorithms, Applications*, 2003.

[GB92] J. Goguen and R. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1):95–146, 1992.

[GWM+93] J. Goguen, Timothy Winkler, J. Meseguer, K. Futatsugi, and J. Jouannaud. Introducing OBJ. In Joseph Goguen, editor, *Applications of Algebraic Specification using OBJ*. Cambridge, 1993.

[HHP93] R. Harper, F. Honsell, and G. Plotkin. A framework for defining logics. *Journal of the Association for Computing Machinery*, 40(1):143–184, 1993.

[JM93] B. Jacobs and T. Melham. Translating dependent type theory into higher order logic. In M. Bezem and J. Groote, editors, *Typed Lambda Calculi and Applications*, pages 209–29, 1993.

[Lan98] S. Mac Lane. *Categories for the Working Mathematician*. Springer, 1998.

[Mak97] M. Makkai. First order logic with dependent sorts (FOLDS), 1997. Unpublished.

[ML75] P. Martin-Löf. An Intuitionistic Theory of Types: Predicative Part. In *Proceedings of the Logic Colloquium 1973*, pages 73–118, 1975.

[MML07] T. Mossakowski, C. Maeder, and K. Lüttich. The Heterogeneous Tool Set. In O. Grumberg and M. Huth, editor, *TACAS 2007*, volume 4424 of *Lecture Notes in Computer Science*, pages 519–522, 2007.

[NPS90] B. Nordström, K. Petersson, and J. Smith. *Programming in Martin-Löf's Type Theory: An Introduction*. Oxford University Press, 1990.

[Obe62] A. Oberschelp. Untersuchungen zur mehrsortigen Quantorenlogik. *Mathematische Annalen*, 145:297–333, 1962.

[Pau94] L. Paulson. *Isabelle: A Generic Theorem Prover*, volume 828 of *Lecture Notes in Computer Science*. Springer, 1994.

[Pit00] A. Pitts. Categorical Logic. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science, Volume 5. Algebraic and Logical Structures*, chapter 2, pages 39–128. Oxford University Press, 2000.

[Rab06] F. Rabe. First-Order Logic with Dependent Types. In N. Shankar and U. Furbach, editors, *Proceedings of the 3rd International Joint Conference on Automated Reasoning*, volume 4130 of *Lecture Notes in Computer Science*, pages 377–391. Springer, 2006.

[Rab08] F. Rabe. *Representing Logics and Logic Translations*. PhD thesis, Jacobs University Bremen, 2008.

[RV02] A. Riazanov and A. Voronkov. The design and implementation of Vampire. *AI Communications*, 15:91–110, 2002.

[TB85] A. Trybulec and H. Blair. Computer assisted reasoning with Mizar. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 26–28, Los Angeles, CA, 1985.

[Urb03] J. Urban. Translating Mizar for first-order theorem provers. In *MKM*, pages 203–215, 2003.

[WAB+99] C. Weidenbach, B. Afshordel, U. Brahm, C. Cohrs, T. Engel, E. Keen, C. Theobalt, and D. Topić. System description: SPASS version 1.0.0. In Harald Ganzinger, editor, *Proceedings of the 16th International Conference on Automated Deduction (CADE-16)*, volume 1632 of *Lecture Notes in Artificial Intelligence*, pages 314–318, Trento, Italy, 1999. Springer.