Internalising modified realisability in constructive type theory

Erik Palmgren
Uppsala University
Department of Mathematics

TYPES meeting in Nottingham
April 19, 2006

Modified realisability

- Modified realisability interpretation : constructive interpretation of logical system into a *simple* type structure
- Used in Minlog and Coq for extracting programs from proofs.

Modified realisability

- Modified realisability interpretation : constructive interpretation of logical system into a *simple* type structure
- Used in Minlog and Coq for extracting programs from proofs.
- Extracted programs are to a large extent free from the computationally irrelevant parts that might be present in programs arising from direct interpretations into constructive type theory (CTT).

Modified realisability

- Modified realisability interpretation : constructive interpretation of logical system into a *simple* type structure
- Used in Minlog and Coq for extracting programs from proofs.
- Extracted programs are to a large extent free from the computationally irrelevant parts that might be present in programs arising from direct interpretations into constructive type theory (CTT).
- The interpretation requires a separate proof of correctness, usually left unformalised.

- We present a completely formalised modified realisability interpretation carried out in the proof support system Agda/Alfa.

- We present a completely formalised modified realisability interpretation carried out in the proof support system Agda/Alfa.
- We use *modified realisability with truth* which has the property that anything realised is also true in CTT. This makes it possible to use and reason about extracted programs in CTT.

- We present a completely formalised modified realisability interpretation carried out in the proof support system Agda/Alfa.
- We use *modified realisability with truth* which has the property that anything realised is also true in CTT. This makes it possible to use and reason about extracted programs in CTT.
- A difference from interpretations as for Minlog, is that the logic interpreted goes beyond first order logic: it is a (constructively) infinitary logic, which arises naturally from the type-theoretic notion of universe.

- We present a completely formalised modified realisability interpretation carried out in the proof support system Agda/Alfa.
- We use *modified realisability with truth* which has the property that anything realised is also true in CTT. This makes it possible to use and reason about extracted programs in CTT.
- A difference from interpretations as for Minlog, is that the logic interpreted goes beyond first order logic: it is a (constructively) infinitary logic, which arises naturally from the type-theoretic notion of universe.
- Our extension to infinitary logic seems to be a novel result.

Method of type universes

Use first two levels of the type hierarchy in Agda

$$\text{Set} \subseteq \text{Type} \subseteq \cdots .$$

Define inductively a type SP : Type of *Simple Propositions*

1. If $A : \text{Set}$, then $\text{atom}(A) : \mathrm{SP}$.
2. $\perp : \mathrm{SP}$.
3. If $P, Q : \mathrm{SP}$, then $P \wedge Q, P \vee Q, P \rightarrow Q : \mathrm{SP}$.
4. If $A : \text{Set}$ and $P : A \rightarrow \mathrm{SP}$, then $\forall(A, P), \exists(A, P) : \mathrm{SP}$.

Method of type universes

Use first two levels of the type hierarchy in Agda

$$\text{Set} \subseteq \text{Type} \subseteq \cdots .$$

Define inductively a type SP : Type of *Simple Propositions*

1. If A : Set, then atom$(A) : \mathrm{SP}$.
2. $\perp : \mathrm{SP}$.
3. If $P, Q : \mathrm{SP}$, then $P \wedge Q, P \vee Q, P \to Q : \mathrm{SP}$.
4. If A : Set and $P : A \to \mathrm{SP}$, then $\forall(A, P), \exists(A, P) : \mathrm{SP}$.

Due to (4) the formulae may be infinitary. Set could also be replaced by a suitably closed universe.

SP	Tp (type of BHK-proofs)	Cr (crude type of realisers)
$\bot$	$\emptyset$	Unit
atom(A)	A	Unit
$P \wedge Q$	$\mathrm{Tp}(P) \times \mathrm{Tp}(Q)$	$\mathrm{Cr}(P) \times \mathrm{Cr}(Q)$
$P \vee Q$	$\mathrm{Tp}(P) + \mathrm{Tp}(Q)$	$\mathrm{Cr}(P) + \mathrm{Cr}(Q)$
$P \rightarrow Q$	$\mathrm{Tp}(P) \rightarrow \mathrm{Tp}(Q)$	$\mathrm{Cr}(P) \rightarrow \mathrm{Cr}(Q)$
$\forall(A, R)$	$(\Pi x : A)\mathrm{Tp}(R(x))$	$(\Pi x : A)\mathrm{Cr}(R(x))$
$\exists(A, R)$	$(\Sigma x : A)\mathrm{Tp}(R(x))$	$(\Sigma x : A)\mathrm{Cr}(R(x))$

SP	Tp (type of BHK-proofs)	Cr (crude type of realisers)
$\perp$	$\emptyset$	Unit
atom(A)	A	Unit
$P \wedge Q$	$\mathrm{Tp}(P) \times \mathrm{Tp}(Q)$	$\mathrm{Cr}(P) \times \mathrm{Cr}(Q)$
$P \vee Q$	$\mathrm{Tp}(P) + \mathrm{Tp}(Q)$	$\mathrm{Cr}(P) + \mathrm{Cr}(Q)$
$P \rightarrow Q$	$\mathrm{Tp}(P) \rightarrow \mathrm{Tp}(Q)$	$\mathrm{Cr}(P) \rightarrow \mathrm{Cr}(Q)$
$\forall(A, R)$	$(\Pi x : A)\mathrm{Tp}(R(x))$	$(\Pi x : A)\mathrm{Cr}(R(x))$
$\exists(A, R)$	$(\Sigma x : A)\mathrm{Tp}(R(x))$	$(\Sigma x : A)\mathrm{Cr}(R(x))$

For $P : \mathrm{SP}$ and $s : \mathrm{Cr}(P)$ the predicate $\mathrm{MR}(P, s)$: Type is defined by recursion on P and expresses that *s is a realiser for P.*

Soundness and conservativity

Soundness Theorem: The axioms and rules of infinitary first logic (with atomic absurdity rule), using sorts in Set, are MR-realised.

Soundness and conservativity

Soundness Theorem: The axioms and rules of infinitary first logic (with atomic absurdity rule), using sorts in Set, are MR-realised.

Mathematical axioms: N-induction and constructive choice for types are MR-realised.

Soundness and conservativity

Soundness Theorem: The axioms and rules of infinitary first logic (with atomic absurdity rule), using sorts in Set, are MR-realised.

Mathematical axioms: N-induction and constructive choice for types are MR-realised.

Conservativity Theorem: If $\mathrm{MR}(P, r)$ holds, then $\mathrm{Tp}(P)$ is true.

Soundness and conservativity

Soundness Theorem: The axioms and rules of infinitary first logic (with atomic absurdity rule), using sorts in Set, are MR-realised.

Mathematical axioms: N-induction and constructive choice for types are MR-realised.

Conservativity Theorem: If $\mathrm{MR}(P, r)$ holds, then $\mathrm{Tp}(P)$ is true.

Remark: The full absurdity rule can be realised with a slight increase in the complexity of the interpretation.

Uses of the interpretation

- Eliminate type dependencies in extracted programs by proving existence in the first order part of the logic. No need to go outside the proof support system.
- Programs from proofs in the infinitary part still has less type depedencies than BHK-programs.
- Only toy examples tested so far. Limitation in the normalisation algorithm for Agda.

References

U Berger, W Buchholz and H Schwichtenberg. *Refined Program Extraction from Classical Proofs* **Annals of Pure and Applied Logic**, 114(2002), 3 – 25.

E Palmgren. *Internalising modified realisability in constructive type theory.* **Logical Methods in Computer Science.** Iss. 2, vol. 1(2005), 1–7. URL: www.lmcs-online.org/