Certification de programmes avec des effets calculatoires

Burak Ekici

Jean-Guillaume Dumas & Dominique Duval & Damien Pous

May 25, 2013 - Dinard, France

1 - Introduction & Motivation

- Burak Ekici, PhD Student at University Joseph Fourier (Grenoble I). Supervised by: Jean-Guillaume Dumas, Dominique Duval & Damien Pous.

- Modeling the computational effects, such as State, Exceptions, IO, Non-Determinism, of non-functional programming languages by using decorated logic [Dominguez&Duval'10]

- Developing the model framework in the Coq proof management system to verify its internal validity & consistency.

2 - States Effect Specification

- Sets (Objects):
  - N (values), U ($unit : \{*\}$)
- Functions (Morphisms):
  - $id_N : \mathsf{N} \to \mathsf{N}$, $id_U : \mathsf{U} \to \mathsf{U}$, $\langle\,\rangle_N : \mathsf{N} \to \mathsf{U}$
  - $lookup_i : \mathsf{U} \to \mathsf{N}$, $update_i : \mathsf{N} \to \mathsf{U}$

  where $i \in \{X, Y, ...\}$ is a memory location identifier. [Dumas et al.'12]

Functions are Classified:

- pure functions: $E.g.\ id^{(0)}_N : \mathsf{N} \to \mathsf{N}$, $id^{(0)}_U : \mathsf{U} \to \mathsf{U}$, $\langle\,\rangle^{(0)}_N : \mathsf{N} \to \mathsf{U}$
- accessors: $E.g.\ lookup^{(1)}_i : \mathsf{U} \to \mathsf{N}$
- modifiers: $E.g.\ update^{(2)}_i : \mathsf{N} \to \mathsf{U}$

Equations are Classified:

- strong equality ($f \equiv g$): if f and g calculate the same result with the same effects on the state structure.
- weak equality ($f \sim g$): if f and g calculate the same result and might have different effects on the state structure.

$$E.g.\ (lookup_i \circ update_i) \sim id_i$$

Properties of the State Structure by Plotkin et al.

1. Annihilation lookup-update. $\forall\, i \in Loc,\ u_i \circ l_i \equiv \mathrm{id}_U : U \to U$
2. Interaction lookup-lookup. $\forall\, i \in Loc,\ l_i \circ \langle\,\rangle_i \circ l_i \equiv l_i : U \to N$
3. Interaction update-update. $\forall\, i \in Loc,\ u_i \circ \pi_2 \circ (u_i \rtimes id_i) \equiv u_i \circ \pi_2 : N \times N \to U$
4. Interaction update-lookup. $\forall\, i \in Loc,\ l_i \circ u_i \sim \mathrm{id}_i : N \to N$
5. Commutation lookup-lookup. $\forall\, i \neq j \in Loc,\ (id_i \ltimes\ l_j) \circ l_i \equiv perm_{j,i} \circ (id_j \ltimes\ l_i) \circ l_j : U \to N \times N$
6. Commutation update-update. $\forall\, i \neq j \in Loc,\ u_j \circ \pi_2 \circ (u_i \rtimes \mathrm{id}_j) \equiv u_i \circ \pi_1 \circ (\mathrm{id}_i \ltimes u_j) : N \times N \to U$
7. Commutation update-lookup. $\forall\, i \neq j \in Loc,\ l_j \circ u_i \circ \pi_1 \equiv \pi_2 \circ (u_i \rtimes \mathrm{id}_j) \circ (id_i \ltimes l_j) : N \times U \to U$

Proofs and Proof Verification Structure in Coq

To prove propositions by Plotkin et al.:

- 75 rules have been used.
  - 23 rules $\Rightarrow$ Monadic Equational Logic.
  - 9 rules $\Rightarrow$ Categorical Products.
  - 43 rules $\Rightarrow$ Derived.
- $\approx$ 1100 lines of Coq Code have been written for proof verifications.

What is next?

- To model other effects (with combinations) and use Coq for related proof verifications.

References

[1] César Domínguez and Dominique Duval. Diagrammatic logic applied to a parameterization process. *MSCS*, 20:639–654, 2010.

[2] Jean-Guillaume Dumas, Dominique Duval, Laurent Fousse, and Jean-Claude Reynaud. Decorated proofs for computational effects: States. In *Proc ACCAT 2012, Electronic Proceedings in Theoretical Computer Science 93*, pages 45–59, 2012.

[3] Gordon Plotkin and John Power. Notions of computation determine monads. In *Proc. FOSSACS 2002, Lecture Notes in Computer Science 2303*, pages 342–356. Springer, 2002.

Many thanks for your patience!

Questions?