Isabelle/HOL and the UTP

Part 3: UTP Values, Models and Predicates

Simon Foster

University of York

February 5, 2013

Recap

- Functions, Datatypes, Record and Subtypes
- The simplifier
- Deduction Rules
- Automated deduction with blast
- Automated FO reasoning with sledgehammer
- The Isar proof script language
- Inductive Proofs

Today

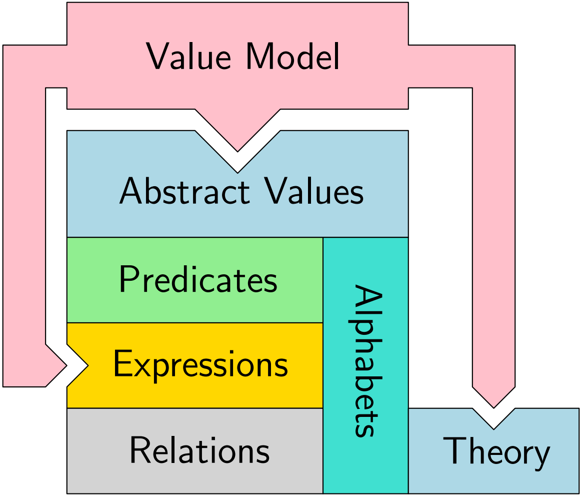
Unifying Theories of Programming

- a predicative relation algebra for defining programming/specification language semantics
- emphasises denotational semantics: precise operator definition
- theories are defined by healthiness conditions – idemptotent functions under which theory elements must be closed
- examples: designs, CSP, objects

Isabelle/UTP

- A deep embedding of the UTP in Isabelle/HOL

Overview of Isabelle/UTP



Predicates

- encoded as subsets of $\mathbb{P}$ (variable $\rightarrow$ value)
- represents possible values each variable can take
- unconstrained variables can take any value, hence the functions are total
- $\emptyset$ represents false, UNIV represents true
- predicate operators generally map to set operators
- we first need to mechanise a notion of variables and values
- mechanisation of values requires some domain theory

Domain Theory 101

Complete Partial Orders

- a chain-complete partial order $(D, \sqsubseteq)$
- ensures the existence of suprema for any chain
- can be thought of as a values ordered by definedness
- usually also pointed: possessing a $\bot$ element

HOLCF (HOL + LCF)

- an implementation of Scott domain theory in Isabelle/HOL
- has a universal domain for injecting all domains (udom)
- gives an account to partial continuous functions

Value Model

- we require a notion of value and type in a model
- each type must exhibit at least one defined value
- values/types specified by mean of the VALUE type-class
- user supplies
  - a value sort 'VALUE
  - a typing relation $\_ : \_$ :: 'VALUE $\Rightarrow$ udom $\Rightarrow$ bool
  - a definedness predicate $\mathcal{D}$
- value sort can be an arbitrary an arbitrary Isabelle type
- types must be injectable into udom
- (type-classes may only have one parameter)

Predicate Encoding

- predicates are introduced in two stages:
  - predicates with no alphabet
  - alphabetised predicates (next time)
- operators are polymorphic over 'VALUE
- additional value axioms can be introduced by value sort classes
- hence proofs only rely on precisely what they need

Predicate tactic

- direct manual proof about predicates is tedious
- could reason about them in the same way as HOL predicates
- we provide an evaluation tactic which performs the conversion
- consists of
  - An evaluation function $[\![\_]\!]_{\_} :: \text{'VALUE WF\_PREDICATE} \Rightarrow \text{'VALUE WF\_BINDING} \Rightarrow \text{bool}$
  - transfer theorems, which prove proof correspondence
  - distribution theorems, e.g. $[\![P \wedge_p Q]\!]b = [\![P]\!]b \wedge [\![Q]\!]b$
- tactic is invoked by utp-pred-tac or utp-pred-auto-tac

Conclusion

- a modular framework for values in UTP
  - important to allow multiple models, e.g. VDM and Z
- domain theory in Isabelle (HOLCF)
- predicate encoding and operators
- predicate evaluation tactic