Conservative Extension in HOL

Roger Bishop Jones

May 21, 2012

Abstract

This note offers an alternative to a proposal by Rob Arthan for a simpler and more powerful replacement for the mechanisms currently provided in the various HOL implementation for defining new constants. [The document is derived from Rob's paper and is mostly exactly the same since I didn't get very far in making the intended changes. My guess is that it is only in the description at the beginning that there may be some differences. The material which it modifies was I think presented by Rob at one of the HOL conferences (probably 2013) without mention of the ideas here, but at the time I write this codicil to the abstract Rob is in preparing final amendments to a version of his paper for publication in a special edition of the Journal of Automated Reasoning and his intention is, as I understand it, to include mention of this possibility in the paper, the details of which we have recently discussed.]

1 The Problem

Neither Rob's proposal nor the more elaborate idea presented here extends in a substantive way what can be done in any of the HOL systems. These proposals allow systems to be specified in ways which are *more abstract* than would otherwise be possible.

One of the criticisms often put forward against a foundationalist approach to mathematics (as exhibited by HOL and by ZFC) is that arbitrary choices are made of ways in which entities such as the various kinds of number are represented, and that these choices go beyond what is essential in the mathematical objects. This complaint is primarily targeted at set theory in which

mathematical entities are "coded up" as various complex sets. From a software engineering point of view this may be thought of as an overspecification of the object of mathematics. The sets used to code numbers have all the desired properties (e.g. comply with the Peano postulates) but have some more (notably, they have members) which are not intended.

Two features of HOL ameliorate this critique. The first is that numbers are drawn from a type of individuals which is known to have sufficient members. The second is that HOL admits definition of types by a mechanism which obscures the true identity of the entities of that type. This method of introducing types ensures that the entities of the new type have the desired properties as established by reasoning about suitable representatives, but their true identity is unknown, and they have none of "collateral" properties posessed by these representatives.

The ability to give abstract definitions of new type constructors was complemented by a facility for introducing new constants by a desired characteristic rather than by explicit identification of values denoted by the constants.

The facility for abstract specification of new constants was introduced shortly after the discovery that the previous definitional extension facility was too liberal, admitting definitions from which contradictions could be derived. The defect in that facility was to allow type variables in the defiens which do not occur in the type of the definendum, and to make `new_specification` safe it was therefore required that no free type variables could be permitted in an extension which did not also appear in the types of the constants introduced by that specification.

The main point of Rob Arthan's proposal was to observe that this constraint is needlessly strong and excludes some useful ways of describing new constants (using universal properties). The liberalisation he proposed hinges on the use of explit witnesses to establish that the desired property is satisfiable (rather than accepting an existentially quantified theorem to establish the existence of such values), and allows the constraint on type variables to be confined to those witnesses, rather than applying to the defining property.

In relation to the motivation for allowing specifications rather than explicit definitions, there is further scope for liberalisation, and it is these further liberalisations which motivate the present proposal.

A typical context in which `new_specification` might be used is in the introduction of the theory of some new kind of entity (e.g. a kind of number). In this one would proceed by first identifying some collection of representatives which could be made into the domain of an structure isomorphic to the

desired new entities by suitable definitions of the required operations over the representatives. The new type constructor is then introduced defined by a bijection between the new types and the chosen representative domains, and the exitence of the required operations over the new type is then proven to justify the introduction of a specification for the required operations over the new type which corresponds to a natural axiomatisation of the new theory (but is known to be conservative over the theory in which it is introduced).

2 Proposed Alternative

The proposal involves the following changes.

1. taking constraints on which constants and type constructors are used out of the logical kernal and decoupling them from the facilities which provide logical extensions, i.e. which introduce constraints on the constants and type constructors.

2. replacing all the existing ways of logically extending a theory except `new_axiom` by a more comprehensive `new_specification`.

The revised `new_specification` is parametrized by:

1. A set of theorems proving that properties delineating representatives for type constructors are satisfiable.

2. A set of theorems which together constitute a conservative extension of the current theory when all their assumptions are discarded.

The theorems in the first set must all be applications in which the function is the defining property for a type and which have no assumptions. The theorems in the second set may have assumptions of one of two forms. The first form asserts the existence of a representation function from objects constructed by one of the new type constructors to its representing type, and asserts explicitly that this is a bijection. The second form assigns a value to one of the new constants.

a theorem of the following form

$$v_1 = t_1, \ldots, v_n = t_n \vdash p$$

where the v_i are variables. If all is well, the revised `new_specification` will then introduce new constants $c_1, \ldots, c_n$ and the following axiom:

$$\vdash P[c_1/v_1, \ldots, c_n/v_n].$$

The revised `new_specification` imposes the following restrictions:

- the theorems which assume a value for some c_i must all assume exactly the same value (literally same term)
- the terms t_i must have no free variables;
- the p must be closed;
- any type variable appearing anywhere in a t_i must appear in the type of the corresponding c_i.

There is no restriction on the type variables appearing in p.

Claim 1 *The revised* `new_specification` *is conservative and hence sound.*

Proof: Let $\Gamma \vdash q$ be a sequent that is provable using the axiom:

$$\vdash p[c_1/v_1, \ldots, c_n/v_n]$$

introduced using the revised `new_specification` and assume that $\Gamma \vdash q$ does not contain any of the c_i. We will show how to transform a proof tree with conclusion $\Gamma \vdash q$ into a proof tree with the same conclusion that does not use the new axiom. First, by simple equality reasoning, derive from the theorem:

$$v_1 = t_1, \ldots, v_n = t_n \vdash p$$

that was passed to `new_specification`, the theorem:

$$\vdash p[t_1/v_1, \ldots, t_n/v_n].$$

Now replace each type instance of a c_i in the proof tree with the corresponding type instance of t_i and wherever a type instance of the axiom $\vdash p[c_1/v_1, \ldots, c_n/v_n]$ is used in the proof tree, replace it with the corresponding type instance of a proof tree for $\vdash p[t_1/v_1, \ldots, t_n/v_n]$. By inspection of the primitive inference rules, if one replaces instances of constants in a correct

inference step by closed terms of the same type in such a way that formulas featuring as assumptions or conclusions of the various sequents involved that were syntactically identical before the replacement remain syntactically identical, then the result is also a correct inference step. As the condition on type variables imposed by the revised `new_specification` guarantees that two instances of a c_i are syntactically identical iff the corresponding instances of t_i are syntactically identical, we have therefore constructed a correct proof tree whose conclusion is $\Gamma \vdash q$. ■

Claim 2 *The revised* `new_specification` *subsumes the functionality of the existing* `new_definition`.

Proof: To define c with axiom $\vdash c = t$, where t has no free variables and contains no type variables that do not appear in its type, apply the revised `new_specification` to the axiom $v = t \vdash v = t$. This is all that is neeeded for the simple form of `new_definition` implemented in ProofPower and all that is needed to define the logical connectives.

For the more general form implemented in HOL 4, assume one wishes to define c with axiom

$$\vdash \forall x_1 \ldots x_n \bullet c\, x_1 \ldots x_n = t,$$

To do this, take the axiom $v = (\lambda x_1 \ldots x_n \bullet t) \vdash v = (\lambda x_1 \ldots x_n \bullet t)$, derive $v = (\lambda x_1 \ldots x_n \bullet t) \vdash \forall x_1 \ldots x_n \bullet v\, x_1 \ldots x_n = t$ from this and then apply the revised `new_specification`. ■

Claim 3 *The revised* `new_specification` *subsumes the functionality of the old* `new_specification`.

Proof: To define $c_1, \ldots, c_n$ with defining axiom $\vdash p[c_1/v_1, \ldots c_n/v_n]$ given the theorem $\vdash \exists v_1 \ldots v_n \bullet p$, first derive the theorem

$$\vdash \exists z \bullet p[\pi_1(z)/v_1, \ldots, \pi_n(z)/v_n]$$

in which the n bound variables $v_1, \ldots v_n$ have been collected into a single n-tuple denoted by the fresh variable z, and where π_i denotes the projection onto the i-th factor. Now derive from that the theorem

$$v_1 = t_1, \ldots, v_n = t_n \vdash p$$

where t_i is $\pi_i(\varepsilon z \bullet p[\pi_1(z)/v_1, \ldots \pi_n(z)/v_n])$. Given this theorem the revised `new_specification` will have the same effect as the old `new_specification` given $\vdash \exists v_1 \ldots v_n \bullet p$. ■

3 Conclusion

Let me assess the proposed alternative against the various observations that led to it:

RJ1 By claim 3, the support for implicit definitions is at least as good with the proposed alternative.

RJ2 By claim 1, the proposed alternative is sound.

RA1 By claim 2, `new_definition` is no longer required. (As noted in the proof of this claim, the special case needed to define the logical connectives does not involve any reasoning about them, so there is no bootstrapping issue).

RA2 The restriction that all the type variables appearing in the defining axiom must appear in the type of all the new constants is relaxed in the proposed alternative. The restriction now applies only to type variables appearing in the witnesses to the consistency of the definition. Defining properties such as initial algebra conditions are supported.

JH1 The proposed revision to `new_specification` is defined solely in terms of equality and primitive language constructs.

MA1 The unintended identities arising as a result of recording definitions in HOL Light will not occur if the revised `new_specification` is adopted as the primitive mechanism for defining constants.

My conclusion is that the proposal is well worth adopting.