Formal Derivation of Proof Rules

Roger Bishop Jones

ICL Defence Systems

VDM-SL PROOF THEORY INVESTIGATIONS

OBJECTIVES

CONSISTENCY, TERMINATION
and
PROOF THEORETIC STRENGTH

STAGES in
FORMAL DEVELOPMENT

DETAILS of
FORMAL THEORIES

CONCLUSIONS from
WORK TO DATE

OBJECTIVES

To FACILITATE the USE OF VDM-SL
in developments involving
FORMAL MACHINE CHECKED PROOF
of
CRITICAL PROPERTIES

To INVESTIGATE the FEASIBILITY
of FORMAL DERIVATION of
PROOF RULES from SEMANTICS

To BROADEN APPLICABILITY
and CONTAIN COMPLEXITY
by DEFERRING SPECIALISATION

CONSISTENCY, TERMINATION and PROOF THEORETIC STRENGTH

We are concerned to ensure the
LOGICAL CONSISTENCY of:

The LOGICAL FOUNDATIONS

User APPLICATION THEORIES

Unless we are assured of their consistency then
NO VALUE
can be attached to any proofs obtained.

APPLICATION THEORIES

Since CONSISTENCY of EXTENSIONS
will NOT be DECIDABLE

it will be NECESSARY to PROVIDE for
their CONSISTENCY to BE PROVABLE
WITHIN the logical SYSTEM

In order that this be possible as often as possible,
a logical system with
high "PROOF THEORETIC STRENGTH"
is needed.

Proof Theoretic Strength
.
Cardinality of Universe

HIGH PROOF THEORETIC STRENGTH

$\Rightarrow$

more SPECIFICATIONS can be SHOWN CONSISTENT

more PROGRAMS can be SHOWN TO TERMINATE

PROOFS can be SHORTER

all of these follow from Goedel's theorems.

OBSERVATIONS on COMPLEXITY

Mike Gordon's HOL logic has:

7 clause abstract syntax

3 primitive type constructors (bool, ind, $\rightarrow$)

2 type inference rules

3 primitive constants (=, $\Rightarrow$, μ)

5 axioms

8 primitive inference rules

It is closely related to Church's simple theory of types.

YET, since its development in 1985, SEVERAL problems in the formulation or implementation of this logic have been found which permit the derivation of contradictions.

VDM-SL will be MORE COMPLEX
by AT LEAST a FACTOR of TEN

HOW DO WE ENSURE that VDM-SL IS SUPPORTED
by SOUND PROOF DEVELOPMENT TOOLS?

HOW DO WE ENSURE that VDM-SL IS SUPPORTED by SOUND PROOF DEVELOPMENT TOOLS?

PROVIDE FORMAL SEMANTICS
in
CLASSICAL SET THEORY (ZFC)

ENSURE that
the UNIVERSE of VDM-SL
is
WELL POPULATED

MECHANICALLY DERIVE PROOF RULES
from
PROOF RULES for SET THEORY

FORMALISATION of SEMANTICS
and
DEVELOPMENT of PROOF THEORY
must be
CONCURRENT and CLOSELY COUPLED

ESTABLISHING the CONSISTENCY of FOUNDATION SYSTEMs

To PROVE the CONSISTENCY of a system
it is necessary to work with a META-LOGIC which has
GREATER PROOF THEORETIC STRENGTH

Since
the CONSISTENCY of the META-LOGIC
may be questioned,
it is desirable to CHOSE a SYSTEM directly RELATED
TO well ESTABLISHED FOUNDATIONS

e.g. ZFC

STAGES in FORMAL DEVELOPMENT

1 formalise ZFC in HOL

2 change PRIMITIVES to FUNCTIONS

3 introduce POLYMORPHISM

4 introduce STRUCTURING

5 TYPES?

REPRESENTATIONS of TYPES(1)

1 ZFC is AXIOMATISED over type ":SET" which is a new PRIMITIVE

2 PURE FUNCTIONS are type ":PPF", a subtype of ":SET"

function_hereditary_DEF
$\vdash$ function_hereditary p =
$(\forall f\bullet$ function f $\wedge$
$\bot_Z \notin_Z$ (image f) $\wedge$
$(\forall x\bullet x \in_Z (\text{field}_Z\ f) \Rightarrow p\ x) \Rightarrow$
p f)
pure_function_DEF
$\vdash$ pure_function s = $(\forall p\bullet$ function_hereditary p $\Rightarrow$ p s)

REPRESENTATIONS of TYPES(2)

3 POLYMORPHIC PURE FUNCTIONS are type ":PPF", and are represented by objects of type ":PF→PF"

4 STRUCTURED POLYMORPHIC FUNCTIONS are type ":SPF", and are represented by ‘REGULAR’ functions of type ":PPF→PPF"

regular
  ⊢ regular ppfun = (∀pf• ∃pfun• ∀ppf•
        REP_PPF(ppfun ppf)pf = pfun(REP_PPF ppf pf))

KEY PRIMITIVES

1 ZFC

membership $\in_Z$, separation Λ_Z

2 PF

application $_f$, abstraction λ_f

3 PPF

application $_p$, abstraction λ_p,
type-vars Tv_p,
type-instantiation $\S_p$,
type-env Te_p

4 SPF

application $_s$, abstraction λ_s,
type-vars Tv_s, individual-vars Iv_s,
type-instantiation $\S_s$, value-instantiation $\S\S_s$,
type-env Te_s, value-env Ie_s

THE AXIOMS of ZERMELO FRAENKEL

There are three main sorts of axiom:

1 LOGICAL axioms (including =)

2 axioms CHARACTERISING SETS

Extensionality and well foundedness.

3 axioms CHARACTERISING ABSTRACTION

The axiom of separation.

4 POPULATING axioms

i.e. axioms which assert the existence of various sets.

This broad pattern is followed by all the foundation systems which we discuss below.

HOL-ZFC PRIMITIVES

Types		":SET"
Constants:		
Membership	$\in_Z$	":SET $\rightarrow$ (SET $\rightarrow$ bool)"
Separation	Λ_Z	":SET $\rightarrow$ ((SET $\rightarrow$ bool) $\rightarrow$ SET)"
Empty set	$\varnothing_Z$	":SET"
Power set	$\mathbb{P}$	":SET $\rightarrow$ SET"
Pair constructor	pair	":SET $\rightarrow$ (SET $\rightarrow$ SET)"
Union	$\cup_Z$	":SET $\rightarrow$ SET"
Natural numbers	$\mathbb{N}$	":SET"
Choice function	μ	":SET $\rightarrow$ SET"

HOL-ZFC DEFINED CONSTANTS

Unit set	unit	":SET $\rightarrow$ SET"
Intersection	$\cap_z$	":SET $\rightarrow$ SET"
Inclusion	$\subseteq_z$	":SET $\rightarrow$ (SET $\rightarrow$ bool)"
Intersection	$\cap_z$	":SET $\rightarrow$ (SET $\rightarrow$ SET)"
Successor	suc	":SET $\rightarrow$ SET"
Transitive	Trans	":SET $\rightarrow$ bool"
Connected	Con	":SET $\rightarrow$ bool"
Ordinal	On	":SET $\rightarrow$ bool"
Successor	Sc	":SET $\rightarrow$ bool"
Natural number	N	":SET $\rightarrow$ bool"

HOL-ZFC AXIOMS

EXT $\vdash \forall x\ y\bullet (\forall z\bullet z \in_Z x \Leftrightarrow z \in_Z y) \Rightarrow (x = y)$

ZF2 $\vdash \forall A\ z\ x\bullet x \in_Z (\Lambda_Z\ z\ A) \Leftrightarrow x \in_Z z \wedge A\ x$

ZF3 $\vdash \varnothing_Z = \Lambda_Z\ \varnothing_Z(\lambda x^1\bullet F)$

ZF4 $\vdash \forall y\ x\bullet x \in_Z (\mathbb{P}\ y) \Leftrightarrow x \subseteq_Z y$

ZF5 $\vdash \forall y\ z\ x\bullet x \in_Z (\text{pair}\ y\ z) \Leftrightarrow (x = y) \vee (x = z)$

ZF6 $\vdash \forall y\ x\bullet x \in_Z (\bigcup_Z y) \Leftrightarrow (\exists z\bullet z \in_Z y \wedge x \in_Z z)$

ZF7 $\vdash \forall x\bullet x \neq \varnothing_Z \Rightarrow (\exists y\bullet y \in_Z x \wedge (y \cap_Z x = \varnothing_Z))$

ZF8 $\vdash \forall x^1\bullet x^1 \in_Z \mathbb{N} \Leftrightarrow N\ x^1$

ZF9 $\vdash \forall f\ r\bullet (\forall x\ y\ z\bullet f\ x\ y \wedge f\ x\ z \Rightarrow (z = y)) \Rightarrow$
$(\exists w\bullet \forall y\bullet y \in_Z w \Leftrightarrow (\exists x\bullet x \in_Z r \wedge f\ x\ y))$

ZF10 $\vdash \forall x^1\bullet x^1 \neq \varnothing_Z \Rightarrow (\mu\ x^1) \in_Z x^1$

HOL-ZFC THEOREMS

ZF_thm19 $\vdash N\ \emptyset_Z$

ZF_thm20 $\vdash \forall x^1 \bullet \emptyset_Z \neq (suc\ x^1)$

ZF_thm22 $\vdash \forall x^1 \bullet N\ x^1 \Rightarrow N(suc\ x^1)$

ZF_thm23 $\vdash \forall x^1\ x^2 \bullet (suc\ x^1 = suc\ x^2) \Rightarrow (x^1 = x^2)$

ZF_thm25

$$\begin{array}{l} \vdash \forall A \bullet \\ \quad A\ \emptyset_Z \wedge (\forall x \bullet N\ x \wedge A\ x \Rightarrow A(suc\ x)) \Rightarrow \\ \quad (\forall x \bullet N\ x \Rightarrow A\ x) \end{array}$$

THE THEORY of "PURE FUNCTIONS"

Types -- ":PF"

Constants --

function_hereditary		":(SET $\rightarrow$ bool) $\rightarrow$ bool"			
pure_function		":SET $\rightarrow$ bool"			
Ω_f	":PF"	$\perp_f$	":PF"	$\cup_f$	":PF $\rightarrow$ PF"
λ_f	":PF $\rightarrow$ ((PF $\rightarrow$ PF) $\rightarrow$ PF)"			Π_f	":PF $\rightarrow$ PF"

Curried Infixes --

$_f$ ":PF $\rightarrow$ (PF $\rightarrow$ PF)"
$\mapsto_f$ ":PF $\rightarrow$ (PF $\rightarrow$ PF)"
$\oplus_f$ ":PF $\rightarrow$ (PF $\rightarrow$ PF)"

Definitions --

function_hereditary_DEF
$\vdash$ function_hereditary p =
($\forall$f• function f $\wedge$
$\perp_Z \notin_Z$ (image f) $\wedge$
($\forall$x• x $\in_Z$ (field$_Z$ f) $\Rightarrow$ p x) $\Rightarrow$
p f)

pure_function_DEF
$\vdash$ pure_function s = ($\forall$p• function_hereditary p $\Rightarrow$ p s)

AXIOMS of PURE FUNCTION THEORY

Of the three main sorts of axiom:

1 LOGICAL axioms (including =)

these remain UNCHANGED

2 axioms CHARACTERISING SETS

These are replaced by comparable axioms for PURE FUNCTIONS (extensionality, well foundedness)

3 axioms CHARACTERISING ABSTRACTION

The axiom of separation is replaced by an axiom of BETA REDUCTION

4 POPULATING axioms

These are changed in detail but play a similar role.

THEOREMS concerning PURE FUNCTIONS

PF1 $\vdash \forall x\ y \bullet (x = y) \Leftrightarrow (\forall z \bullet x\ {}_f\ z = y\ {}_f\ z)$

PF2 $\vdash \forall d\ f\ z \bullet (\lambda_f\ d\ f)\ {}_f\ z =$
$((d\ {}_f\ z = \bot_f) \Rightarrow \bot_f \mid f\ z)$

PF3 $\vdash \forall x \bullet \Omega_f\ {}_f\ x = \bot_f$

PF4 .$\vdash \forall f\ z \bullet (\Pi\ f)\ {}_f\ z =$
$(\forall g \bullet$
$((f\ {}_f\ g = \bot_f) \Rightarrow$
$(z\ {}_f\ g = \bot_f) \mid$
$((f\ {}_f\ g)\ {}_f\ (z\ {}_f\ g)) \neq \bot_f) \Rightarrow$
$T_f \mid$
$\bot_f)$

PF5 $\vdash \forall x\ y\ z \bullet (x \mapsto_f y)\ {}_f\ z = ((z = x) \Rightarrow y \mid \bot_f)$

PF6 $\vdash \forall x\ y\ z \bullet (x \oplus_f y)\ {}_f\ z =$
$((y\ {}_f\ z = \bot_f) \Rightarrow x\ {}_f\ z \mid y\ {}_f\ z)$

PF7 .$\vdash \forall p \bullet$
$(\forall q \bullet (\forall r \bullet ((\mathrm{fie}\ q)\ {}_f\ r) \neq \bot_f \Rightarrow p\ r) \Rightarrow p\ q) \Rightarrow$
$(\forall q \bullet p\ q)$

PF11.$\vdash \forall f\ g \bullet$
$(((\cup_f\ f)\ {}_f\ g) \neq \bot_f \Rightarrow$
$(\exists i \bullet$
$(f\ {}_f\ i) \neq \bot_f \wedge (i\ {}_f\ g = (\cup_f\ f)\ {}_f\ g)) \mid$
$(\forall i \bullet (f\ {}_f\ i) \neq \bot_f \Rightarrow (i\ {}_f\ g = \bot_f)))$

PF13 $\vdash \neg(\bot_f = T_f)$

The Theory ppf136

Types -- ":PPF"

Constants --

regular ":(PPF → PPF) → bool"
λ_p ":PPF → ((PPF → PPF) → PPF)"
Ω_p ":PPF" $\perp_p$ ":PPF"
T_p ":PPF" F_p ":PPF"
if_p ":PPF → (PPF → (PPF → PPF))"
$\cup_p$ ":PPF → PPF" Π_p ":PPF → PPF"
Tv_p ":PPF → PPF" Te_p ":PPF"

Curried Infixes --

$_p$ ":PPF → (PPF → PPF)"
$\mapsto_p$ ":PPF → (PPF → PPF)"
$\oplus_p$ ":PPF → (PPF → PPF)"
$==_p$ ":PPF → (PPF → PPF)"
$\S_p$ ":PPF → (PPF → PPF)"

the AXIOMATISATION of POLYMORPHIC PURE FUNCTIONS

Of the three main sorts of axiom:

1 LOGICAL axioms (including =)

The host logic (HOL) no longer supplies adequate machinery. Equality and conditionals need to be redefined (if_p, $==_p$).

2 axioms CHARACTERISING FUNCTIONS

These are replaced by comparable axioms for POLYMORPHIC PURE FUNCTIONS (extensionality, well foundedness)

3 the axiom of BETA REDUCTION

A variant of BETA REDUCTION is introduced dependent on REGULARITY, TYPE INSTANTIATION is introduced, supported by an analogous axiom.

4 POPULATING axioms

These are changed in detail but play a similar role.

Definitions --

regular

$\vdash$ regular ppfun = ($\forall$pf• $\exists$pfun• $\forall$ppf•
REP_PPF(ppfun ppf)pf = pfun(REP_PPF ppf pf))

Theorems --

PPF1 $\vdash \forall$x y• (x = y) $\Leftrightarrow$ ($\forall$z• x $_p$ z = y $_p$ z)

PPF2 $\vdash \forall$d m• regular m $\Rightarrow$
($\forall$z• (λ_p d m) $_p$ z =
if$_p$ ((d $_p$ z) ==$_p$ $\perp_p$) $\perp_p$ (m z))

PPF3 $\vdash \forall$p• Ω_p $_p$ p = $\perp_p$

PPF5 $\vdash \forall$x y z• (x $\mapsto_p$ y) $_p$ z = if$_p$ (z ==$_p$ x) y $\perp_p$

PPF6 $\vdash \forall$x y z• (x $\oplus_p$ y) $_p$ z =
if$_p$((y $_p$ z) ==$_p$ $\perp_p$)(x $_p$ z)(y $_p$ z)

PPF13 $\vdash \neg(\perp_p = T_p)$

the AXIOMATISATION of STRUCTURED POLYMORPHIC FUNCTIONS

Of the three main sorts of axiom:

1 LOGICAL axioms (including =)

The host logic (HOL) no longer supplies adequate machinery. Equality and conditionals need to be redefined (if_S, $==_S$), host language abstraction is now displaced.

2 axioms CHARACTERISING POLYMORPHIC FUNCTIONS

These are replaced by comparable axioms for STRUCTURED POLYMORPHIC FUNCTIONS (extensionality, well foundedness)

3 the axiom of BETA REDUCTION

The REGULARITY CLAUSE IN the axiom of BETA REDUCTION is ELIMINATED, VALUE INSTANTIATION is introduced, supported by an analogous axiom.

4 POPULATING axioms

These are changed in detail but play a similar role.

The Theory ppf137

Types -- ":SPF"

Constants --

λ_S	":SPF $\rightarrow$ (SPF $\rightarrow$ (SPF $\rightarrow$ SPF))"		
Iv_S	":SPF $\rightarrow$ SPF"		
Ie_S ":SPF"	Tv_S ":SPF $\rightarrow$ SPF"	Te_S ":SPF"	
Ω_S ":SPF"	$\perp_S$ ":SPF"	T_S ":SPF"	
if_S	":SPF $\rightarrow$ (SPF $\rightarrow$ (SPF $\rightarrow$ SPF))"		
$\cup_S$	":SPF $\rightarrow$ SPF"		
Π_S	":SPF $\rightarrow$ SPF"		

Curried Infixes --

$_S$	":SPF $\rightarrow$ (SPF $\rightarrow$ SPF)"
$\S\S_S$	":SPF $\rightarrow$ (SPF $\rightarrow$ SPF)"
$\S_S$	":SPF $\rightarrow$ (SPF $\rightarrow$ SPF)"
$\mapsto_S$	":SPF $\rightarrow$ (SPF $\rightarrow$ SPF)"
$\oplus_S$	":SPF $\rightarrow$ (SPF $\rightarrow$ SPF)"
$==_S$	":SPF $\rightarrow$ (SPF $\rightarrow$ SPF)"

Theorems --

SPF1 $\vdash \forall x\ y \bullet (x = y) \Leftrightarrow (\forall z \bullet x\ _S\ z = y\ _S\ z)$

SPF13 $\vdash \neg(\perp_S = T_S)$

CONCLUSIONS

FORMAL DERIVATION of PROOF RULES

is

DESIRABLE

and

FEASIBLE

but

EXPENSIVE

(c300 terminal hours so far)

WHAT I WOULD DO DIFFERENT
IF I STARTED AGAIN FROM SCRATCH

DIFFERENT formulation of SET THEORY

THEN

NOT PURE FUNCTIONS

but

PURE FUNCTIONS and ‘TYPES’
without $\perp$

this is mathematically nicer
and provides a staging post towards
PURE FUNCTORS and CATEGORIES