

IMP in HOLCF

Tobias Nipkow and Robert Sandner

December 12, 2021

Contents

1	Denotational Semantics of Commands in HOLCF	1
1.1	Definition	1
1.2	Equivalence of Denotational Semantics in HOLCF and Evaluation Semantics in HOL	1
2	Correctness of Hoare by Fixpoint Reasoning	2

1 Denotational Semantics of Commands in HOLCF

theory Denotational imports HOLCF "HOL-IMP.Big_Step" begin

1.1 Definition

definition

```
dlift :: "('a::type) discr -> 'b::pcpo) => ('a lift -> 'b)" where
"dlift f = (LAM x. case x of UU => UU | Def y => f.(Discr y))"
```

```
primrec D :: "com => state discr -> state lift"
```

where

```
"D(SKIP) = (LAM s. Def(undiscr s))"
| "D(X ::= a) = (LAM s. Def((undiscr s)(X := aval a (undiscr s))))"
| "D(c0 ;; c1) = (dlift(D c1) oo (D c0))"
| "D(IF b THEN c1 ELSE c2) =
  (LAM s. if bval b (undiscr s) then (D c1).s else (D c2).s)"
| "D(WHILE b DO c) =
  fix.(LAM w s. if bval b (undiscr s) then (dlift w).(D c).s
    else Def(undiscr s))"
```

1.2 Equivalence of Denotational Semantics in HOLCF and Evaluation Semantics in HOL

```
lemma dlift_Def [simp]: "dlift f.(Def x) = f.(Discr x)"
```

<proof>

```

lemma cont_dlift [iff]: "cont (%f. dlift f)"
  <proof>

lemma dlift_is_Def [simp]:
  "(dlift f.l = Def y) = ( $\exists x. l = \text{Def } x \wedge f \cdot (\text{Discr } x) = \text{Def } y$ )"
  <proof>

lemma eval_implies_D: "(c,s)  $\Rightarrow$  t  $\implies$  D c.(Discr s) = (Def t)"
  <proof>

lemma D_implies_eval: " $\forall s t. D c \cdot (\text{Discr } s) = (\text{Def } t) \longrightarrow (c,s) \Rightarrow t$ "
  <proof>

theorem D_is_eval: "(D c.(Discr s) = (Def t)) = ((c,s)  $\Rightarrow$  t)"
  <proof>

end

```

2 Correctness of Hoare by Fixpoint Reasoning

theory HoareEx **imports** Denotational **begin**

An example from the HOLCF paper by Müller, Nipkow, Oheimb, Slotosch [1]. It demonstrates fixpoint reasoning by showing the correctness of the Hoare rule for while-loops.

type_synonym assn = "state $\Rightarrow$ bool"

definition

hoare_valid :: "[assn, com, assn] $\Rightarrow$ bool" ("|= {(1_)} / (_) / {(1_)}" 50) **where**
 "|= {P} c {Q} = ($\forall s t. P s \wedge D c \cdot (\text{Discr } s) = \text{Def } t \longrightarrow Q t$)"

lemma WHILE_rule_sound:

"|= {A} c {A} $\implies$ |= {A} WHILE b DO c { $\lambda s. A s \wedge \neg \text{bval } b s$ }"
 <proof>

end

References

- [1] O. Müller, T. Nipkow, D. v. Oheimb, and O. Slotosch. HOLCF = HOL + LCF. *J. Functional Programming*, 9:191–223, 1999.