Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Dipl.-Inf. Achim D. Brucker
Dr. Burkhart Wolff

Computer-supported Modeling and Reasoning

http://www.infsec.ethz.ch/education/permanent/csmr/

(rev. 16814)

Submission date: –

HOL: Derived Rules

In the lecture, standard and non-standard models of HOL have been presented in informal notation based on ZF set theory.

On this basis, a small set of axioms is justified, which serve as foundation of HOL. In this exercise, we will prove the basic logical rules of Higher-order logic (HOL) from these axioms and elementary definitions.

1 Background

1.1 Higher-order Logic

We have seen in lecture "HOL: Deriving Rules" how all well-known inference rules for logical connectives and quantifiers can be derived in HOL. We now want to do some of these proofs in Isabelle. Those rules are available by default since they are derived from the eight basic rules once and for all.

Of course, these rules are already proved in the standard Isabelle/HOL library. Nevertheless, do not to use library proofs for them and apply automated tactics only with your own derived rules.

Following general convention, the syntax for function application in HOL is just `f x` instead of `f(x)` as in FOL. function application

2 Isabelle/HOL

2.1 Technicalities

HOL As for FOL you have to tell Isabelle that you want to work in HOL; choose HOL by selecting ⟨`Isabelle/Isar` ▷ ⟨`Logics` ▷ `HOL`⟩⟩. Within Isabelle/HOL the basic theory (on which you build your own theory) is called Main, thus your basic theory file for this exercise should look like: Main

```
theory ex7 = Main:
lemma fun_cong: "f=g ⟹f(x) = g(x)"

end
```

2.2 The Logical Foundation

```
True_def:      "True          ≡((λx::bool. x) = (λx. x))"
All_def :      "∀ P  ≡(P = (λx. True))"
Ex_def:        "∃ P  ≡∀Q. (∀x. P x ⟶Q) ⟶Q"
False_def :    "False         ≡(∀P. P)"
not_def:       "¬ P      ≡P⟶False"
and_def:       "P ∧Q     ≡∀R. (P⟶Q⟶R) ⟶R"
or_def :       "P ∨Q      ≡∀R. (P⟶R) ⟶(Q⟶R) ⟶R"
if_def :       "If  P x y     ≡THE z::'a. (P=True ⟶z=x)
                                   ∧ (P=False ⟶z=y)"

eq_reflection :  "(x=y) ⟹(x≡y)"

refl :         "t = (t::'a)"
subst:         "⟦ s = t; P(s) ⟧ ⟹P(t::'a)"

ext:           "(⋀x::'a. (f x ::'b) = g x) ⟹(λx. f x) = (λx. g x)"

the_eq_trivial :  "(THE x. x = a) = (a::'a)"

impl:          "(P ≡Q) ⟹P⟶Q"
mp:            "⟦ P⟶Q; P ⟧ ⟹ Q"

iff :          "(P⟶Q) ⟶(Q⟶P) ⟶(P=Q)"
True_or_False :  "(P=True) ∨(P=False)"
```

The axiom of the THE-operator seems to be obviously true, but somewhat pointless. In each type τ there is a function assigned to this operator, that chooses out of the set of possible values in the semantic domain of τ the element, that is equal to a. However, since we may write THE x. P x, the THE-operator may be used quite flexibly to define elements that are uniquely defined by a predicate P; in other words: the use of the operator boils down to the proof of uniqueness with respect to P.

THE-operator

2.3 Exercise 29

Derive the following rules:

1. $f = g \Longrightarrow f(x) = g(x) \quad (\mathit{fun\_cong})$
2. $x = y \Longrightarrow f(x) = f(y) \quad (\mathit{arg\_cong})$

2.4 Exercise 30

Derive the following rules presented in the lecture:

1. transitivity and symmetry

$$\begin{aligned} s = t &\Longrightarrow t = s && (\mathit{sym}) \\ [\![r = s; s = t]\!] &\Longrightarrow r = t && (\mathit{trans}) \end{aligned}$$

2. rules about *iff*:

$$\begin{aligned} [\![P \Longrightarrow Q; Q \Longrightarrow P]\!] &\Longrightarrow P = Q && (\mathit{iffI}) \\ [\![P = Q; Q]\!] &\Longrightarrow P && (\mathit{iffD2}) \end{aligned}$$

3. rules about *True*:

$$\begin{aligned} &\mathit{True} && (\mathit{TrueI}) \\ P = \mathit{True} &\Longrightarrow P && (\mathit{eqTrueE}) \\ P &\Longrightarrow P = \mathit{True} && (\mathit{eqTrueI}) \end{aligned}$$

4. rules about $\forall$:

$$\begin{aligned} (\bigwedge x. P\,x) &\Longrightarrow \forall x. P\,x && (\mathit{allI}) \\ (\forall x. P\,x) &\Longrightarrow P\,x && (\mathit{spec}) \end{aligned}$$

5. rules about *False*:

$$\begin{aligned} \mathit{False} &\Longrightarrow P && (\mathit{FalseE}) \\ \mathit{False} = \mathit{True} &\Longrightarrow P && (\mathit{False\_neq\_True}) \\ \mathit{True} = \mathit{False} &\Longrightarrow P && (\mathit{True\_neq\_False}) \end{aligned}$$

6. rules about $\neg$:

$$(P \Longrightarrow \mathit{False}) \Longrightarrow \neg P \quad (\mathit{notI})$$
$$[\![\neg P; P]\!] \Longrightarrow R \qquad (\mathit{notE})$$
$$\neg(\mathit{True} = \mathit{False}) \qquad (\mathit{True\_Not\_False})$$

7. rules about $\exists$:

$$P(x) \Longrightarrow \exists x. P\, x \qquad (\mathit{exI})$$
$$[\![(\exists x. P\, x); \bigwedge x. P\, x \Longrightarrow Q]\!] \Longrightarrow Q \qquad (\mathit{exE})$$

8. rules about $\wedge$:

$$[\![P; Q]\!] \Longrightarrow P \wedge Q \qquad (\mathit{conjI})$$
$$P \wedge Q \Longrightarrow P \qquad (\mathit{conjEL})$$
$$P \wedge Q \Longrightarrow Q \qquad (\mathit{conjER})$$
$$[\![P \wedge Q; [\![P; Q]\!] \Longrightarrow R]\!] \Longrightarrow R \qquad (\mathit{conjE})$$

9. rules about $\vee$:

$$P \Longrightarrow P \vee Q \qquad (\mathit{disjIL})$$
$$Q \Longrightarrow P \vee Q \qquad (\mathit{disjIR})$$
$$[\![P \vee Q; P \Longrightarrow R; Q \Longrightarrow R]\!] \Longrightarrow R \qquad (\mathit{disjE})$$

10. and finaly, exluded middle:

$$P \vee \neg P \quad (\mathit{excluded\ middle})$$

2.5 Exercise 31

Prove the following properties:

$$[\![P\ a; \bigwedge x. P\ x \Longrightarrow x = a]\!] \Longrightarrow (\mathrm{THE}\ x. P\ x) = a \quad (\mathit{the\_equality})$$

2.6 Exercise 32

if −then−else

Prove the following two properties of the if −then−else:

$$Q = \mathit{True} \Longrightarrow (\text{if } Q \text{ then } x \text{ else } y) = x \qquad (\mathit{ite\_then})$$
$$Q = \mathit{False} \Longrightarrow (\text{if } Q \text{ then } x \text{ else } y\ ) = y \qquad (\mathit{ite\_else})$$