VSTTE'10 Competition Q3 in ProofPower-HOL (v2)

Rob Arthan

18th August 2010

1 Introduction

See solution to Q1 for a bit of background.

This document is a literate script containing the specs, "code" and proof commands. See the theory listing towards the end of the document for a collected summary of the definitions and results.

2 Red Tape

SML
```
open_theory"hol";
new_theory "vstte10-3";
set_merge_pcs ["hol1"];
```

3 The Code

We need subscription for lists. This is nice as an infix operator:

SML
```
declare_infix(300, "sub");
```

HOL Constant
```
        $sub : 'a LIST → ℕ → 'a
─────────────────────────────────────────────
 ∀x xs n•
        [] sub n = Arbitrary
∧       (Cons x xs) sub n = if n = 0 then x else xs sub (n-1)
```

We define a general purpose finder and then instantiate it.

HOL Constant
```
        Finder : ('a → BOOL) → 'a LIST → ℕ
─────────────────────────────────────────────
 ∀test x xs•
        Finder test [] = 0
∧       Finder test (Cons x xs) =
        if test x then 0 else Finder test xs + 1
```

HOL Constant

$$\textbf{\textit{ZeroFinder}} : \mathbb{N}\ LIST \rightarrow \mathbb{N}$$

$$ZeroFinder = Finder\ (\$= 0)$$

4 Testing, Testing, ...

ML bindings for the defining theorems:

SML

```
val sub_def = get_spec ⌜$sub⌝;
val finder_def = get_spec ⌜Finder⌝;
val zero_finder_def = get_spec ⌜ZeroFinder⌝;
```

A few tests just to show its all executable (see theory listing towards the end for the results);

SML

```
val sub_test_thm = save_thm ("sub_test_thm",
        rewrite_conv [map_def, sub_def]
        ⌜Map ($sub ["d"; "o"; "g"]) [2;1;1;0]⌝);
val zero_finder_testthm = save_thm ("zero_finder_test_thm",
        rewrite_conv [map_def, zero_finder_def, finder_def]
        ⌜Map ZeroFinder [
                [0]; [0;1]; [1;0;2]; [1;2;3]
        ]⌝);
```

5 Code Properties

Taken together the following tbree theorems give the properties required in the question.

SML

```
set_goal([], ⌜
∀test ns• Finder test ns ≤ Length ns
⌝);
a(REPEAT strip_tac);
a(list_induction_tac ⌜ns⌝ THEN
        asm_rewrite_tac[length_def, finder_def]
        THEN REPEAT strip_tac);
a(cases_tac⌜test x⌝ THEN asm_rewrite_tac[]);
val finder_≤_length_thm = save_pop_thm "finder_≤_length_thm";
```

SML

```
set_goal([], ⌜
∀ns test i• i < Finder test ns ⇒ ¬test (ns sub i)
⌝);
a(strip_tac THEN strip_tac);
a(list_induction_tac ⌜ns⌝ THEN
        asm_rewrite_tac[length_def, finder_def]
        THEN REPEAT strip_tac);
a(POP_ASM_T ante_tac THEN cases_tac⌜test x⌝
        THEN asm_rewrite_tac[]);
a(rewrite_tac[pc_rule1"lin_arith" prove_rule[]
        ⌜∀j•i < j + 1 ⇔ i < j ∨ i = j⌝]
        THEN REPEAT strip_tac);
(* *** Goal "1" *** *)
a(rewrite_tac[sub_def]);
a(cases_tac⌜i = 0⌝ THEN asm_rewrite_tac[]);
a(POP_ASM_T ante_tac THEN
        rewrite_tac[pc_rule1"lin_arith" prove_rule[]
        ⌜¬i = 0 ⇔ 1 ≤ i⌝] THEN rewrite_tac[≤_def]
        THEN REPEAT strip_tac
        THEN all_var_elim_asm_tac1);
a(rewrite_tac[plus_comm_thm]);
a(DROP_NTH_ASM_T 3 bc_thm_tac
        THEN PC_T1 "lin_arith" asm_prove_tac[]);
(* *** Goal "2" *** *)
a(all_var_elim_asm_tac1);
a(rewrite_tac[sub_def]);
a(cases_tac⌜Finder test ns = 0⌝ THEN asm_rewrite_tac[]);
a(POP_ASM_T ante_tac THEN
        rewrite_tac[pc_rule1"lin_arith" prove_rule[]
        ⌜∀i• ¬i = 0 ⇔ 1 ≤ i⌝] THEN rewrite_tac[≤_def]
        THEN REPEAT strip_tac);
a(TOP_ASM_T (rewrite_thm_tac o eq_sym_rule));
a(DROP_NTH_ASM_T 3 bc_thm_tac
        THEN PC_T1 "lin_arith" asm_prove_tac[]);
val less_finder_thm = save_pop_thm "less_finder_thm";
```

SML

```
set_goal([], ⌜
∀ns test• Finder test ns < Length ns ⇒ test (ns sub Finder test ns)
⌝);
a(strip_tac);
a(list_induction_tac ⌜ns⌝ THEN
        asm_rewrite_tac[length_def, finder_def]);
a(REPEAT ∀_tac);
a(cases_tac ⌜test x⌝ THEN asm_rewrite_tac[sub_def]);
val finder_less_length_thm = save_pop_thm "finder_less_length_thm";
```

6 THE THEORY vstte10-3

6.1 Parents

hol

6.2 Constants

$***sub***	$'a\ LIST \to \mathbb{N} \to 'a$
Finder	$('a \to BOOL) \to 'a\ LIST \to \mathbb{N}$
ZeroFinder	$\mathbb{N}\ LIST \to \mathbb{N}$

6.3 Fixity

Right Infix 300:
sub

6.4 Definitions

sub $\vdash \forall\ x\ xs\ n$
$\bullet\ []\ sub\ n = Arbitrary$
$\wedge\ Cons\ x\ xs\ sub\ n$
$= (if\ n = 0\ then\ x\ else\ xs\ sub\ n - 1)$

Finder $\vdash \forall\ test\ x\ xs$
$\bullet\ Finder\ test\ [] = 0$
$\wedge\ Finder\ test\ (Cons\ x\ xs)$
$= (if\ test\ x\ then\ 0\ else\ Finder\ test\ xs + 1)$

ZeroFinder $\vdash ZeroFinder = Finder\ (\$= 0)$

6.5 Theorems

sub_test_thm $\vdash Map\ (\$sub\ [\text{"}d\text{"}; \text{"}o\text{"}; \text{"}g\text{"}])\ [2; 1; 1; 0]$
$= [\text{"}g\text{"}; \text{"}o\text{"}; \text{"}o\text{"}; \text{"}d\text{"}]$

zero_finder_test_thm
$\vdash Map\ ZeroFinder\ [[0]; [0; 1]; [1; 0; 2]; [1; 2; 3]]$
$= [0; 0; 1; 3]$

finder_≤_length_thm
$\vdash \forall\ test\ ns\bullet\ Finder\ test\ ns \leq Length\ ns$

less_finder_thm
$\vdash \forall\ ns\ test\ i\bullet\ i < Finder\ test\ ns \Rightarrow \neg\ test\ (ns\ sub\ i)$

finder_less_length_thm
$\vdash \forall\ ns\ test$
$\bullet\ Finder\ test\ ns < Length\ ns$
$\Rightarrow test\ (ns\ sub\ Finder\ test\ ns)$

7 INDEX