Project: DAZ PROJECT

Title: Calculator Example VCs Proof Scripts

Ref: ISS/HAT/DAZ/WRK513 *Issue:* 1.14 *Date:* 22 July 2011

Status: Informal *Type:* Technical

Author:

Name	*Location*	*Signature*	*Date*
R.D. Arthan	WIN01		
G.M. Prout	HAT Team		

Abstract: This document provides VC proof scripts for the Calculator Example from [2].

Distribution: Library

0 DOCUMENT CONTROL

0.1 Contents List

0.2 Document Cross References

[1] ISS/HAT/DAZ/USR501. *Compliance Tool — User Guide.* Lemma 1 Ltd., `http://www.lemma-one.com`.

[2] ISS/HAT/DAZ/USR503. *Compliance Tool — Proving VCs.* Lemma 1 Ltd., `http://www.lemma-one.com`.

[3] ISS/HAT/DAZ/WRK507. *Calculator Example.* R.D. Arthan, Lemma 1 Ltd., `http://www.lemma-one.com`.

1 INTRODUCTION

This document provides proof scripts for all the VCs generated in the Calculator Example from [2]. The style of the proofs is as advocated in that tutorial. This example was first presented in [3]. Part of the purpose of that document was to demonstrate the insertion of hypertext links in a literate script by the compliance tool. This meant that the the VC proofs were deliberately presented in a non obvious way to illustrate the advantage of these hypertext links. Here, they have been reproved in the order in which they were generated. The proofs have been amended slightly from those in [3] so that they follow the style of proof advocated in [2].

The simplest way to interact with these proofs would be to load the calculator example from [2]. First create the file *usr503.sml* from *usr503.doc* then make yourself a daz database. Instructions on how to do this are in [1]. Then enter the following command in your database:

SML

```
use_file"usr503";
```

2 PREAMBLE

The following Standard ML command sets up the Compliance Tool to process the rest of the script.

SML

```
val _ = map (fn thy => (open_theory thy;
        mapfilter delete_thm (flat (map fst (get_thms thy)))))
        (get_descendants "calc_prelims");
open_theory"calc_prelims";
set_pc"cn1";
```

3 SUBSIDIARY LEMMAS

SML

```
push_pc"z_library1";
set_goal([], ⓩ∀m : NATURAL• m ≥ 0⌝);
a(rewrite_tac[z_get_specⓩNATURAL⌝] THEN REPEAT strip_tac);
val natural_thm = pop_thm();
pop_pc();
```

SML

```
push_pc"z_library1";
set_goal([], ⓩfact 0 = 1 ∧ fact 1 = 1⌝);
a(rewrite_tac[cn_fact_thm, rewrite_rule[cn_fact_thm]
        ((z_∀_elimⓩ0⌝ o ∧_right_elim) cn_fact_thm)]);
val fact_thm  = pop_thm();
pop_pc();
```

SML

```
push_pc"z_library1";
set_goal([], ⓩ∀x: ℤ•  x ** 1 = x⌝);
a(REPEAT strip_tac);
a(rewrite_tac[rewrite_rule[](
    z_∀_elimⓩ(x ≜ x, y ≜ 0)⌝ (∧_right_elim(z_get_specⓩ(_**_)⌝)))]);
val star_star_1_thm = pop_thm();
```

SML

```
set_goal([], ⓩ∀x: ℤ•  x ** 2 = x * x⌝);
a(REPEAT strip_tac);
a(rewrite_tac[star_star_1_thm, rewrite_rule[](
    z_∀_elimⓩ(x ≜ x, y ≜ 1)⌝ (∧_right_elim(z_get_specⓩ(_**_)⌝)))]);
val star_star_2_thm = pop_thm();
pop_pc();
```

4 VC PROOFS

SML

```
open_theory "OPS′body";
set_goal([],get_conjecture"−""vcOPS_1");
a(cn_vc_simp_tac[]);
save_pop_thm"vcOPS_1";
```

SML

```
set_goal([],get_conjecture"−""vcOPS_2");
a(cn_vc_simp_tac[]);
save_pop_thm"vcOPS_2";
```

SML

```
set_goal([],get_conjecture"−""vcOPS_3");
a(cn_vc_simp_tac[]);
save_pop_thm"vcOPS_3";
```

SML

```
set_goal([],get_conjecture"−""vcOPS_4");
a(cn_vc_simp_tac[]);
save_pop_thm"vcOPS_4";
```

SML

```
open_theory "OPSoDIGIT_BUTTON′proc";
set_goal([],get_conjecture"−""vcOPSoDIGIT_BUTTON_1");
a(cn_vc_simp_tac[]);
save_pop_thm"vcOPSoDIGIT_BUTTON_1";
```

SML

```
set_goal([],get_conjecture"-""vcOPSoDIGIT_BUTTON_2");
a(cn_vc_simp_tac[]);
save_pop_thm"vcOPSoDIGIT_BUTTON_2";
```

SML

```
set_goal([],get_conjecture"-""vc3001_1");
a(cn_vc_simp_tac[]);
a(REPEAT strip_tac THEN asm_rewrite_tac[cn_DO_DIGIT_thm]);
save_pop_thm"vc3001_1";
```

SML

```
set_goal([],get_conjecture"-""vc3001_2");
a(cn_vc_simp_tac[]);
a(REPEAT strip_tac THEN asm_rewrite_tac[cn_DO_DIGIT_thm]);
save_pop_thm"vc3001_2";
```

SML

```
open_theory "OPSoOPERATION_BUTTON'proc";
set_goal([],get_conjecture"-""vcOPSoOPERATION_BUTTON_1");
a(cn_vc_simp_tac[]);
save_pop_thm"vcOPSoOPERATION_BUTTON_1";
```

SML

```
set_goal([],get_conjecture"-""vcOPSoOPERATION_BUTTON_2");
a(cn_vc_simp_tac[]);
save_pop_thm"vcOPSoOPERATION_BUTTON_2";
```

SML

```
set_goal([], get_conjecture"-""vc3002_1");
a(cn_vc_simp_tac calc_thms);
a ⇒_tac;
a(asm_rewrite_tac[]);
save_pop_thm"vc3002_1";
```

SML

```
val number_ax = get_axiom"-""Constraint 1";
```

SML

```
set_goal([], get_conjecture"−""vc3002_2");
a(cn_vc_simp_tac calc_thms);
a ⇒_tac;
a(asm_rewrite_tac[]);
a(LIST_DROP_NTH_ASM_T[1,2,3,4,5,6,8,9,10] (MAP_EVERY discard_tac));
a strip_tac;
a(lemma_tac ⓩGVoDISPLAY ∈ NATURAL⌝);
(* *** Goal "1" *** *)
a(ante_tac number_ax);
a(PC_T1"calc_prelims"asm_rewrite_tac[z_get_specⓩNATURAL⌝]);
a(DROP_NTH_ASM_T 2 ante_tac THEN PC_T1 "z_lin_arith" prove_tac[]);
(* *** Goal "2" *** *)
a(ALL_FC_T rewrite_tac[z_get_specⓩFACT⌝]);
save_pop_thm"vc3002_2";
```

SML

```
set_goal([], get_conjecture"−""vc3002_3");
a(cn_vc_simp_tac calc_thms);
a ⇒_tac;
a(asm_rewrite_tac[]);
a(LIST_DROP_NTH_ASM_T[1,2,3,4,5,6,7,9,10,11] (MAP_EVERY discard_tac));
a strip_tac;
a(lemma_tac ⓩGVoDISPLAY ∈ NATURAL⌝);
(* *** Goal "1" *** *)
a(ante_tac number_ax);
a(PC_T1"calc_prelims"asm_rewrite_tac[z_get_specⓩNATURAL⌝]);
a(DROP_NTH_ASM_T 2 ante_tac THEN PC_T1 "z_lin_arith" prove_tac[]);
(* *** Goal "2" *** *)
a(ALL_FC_T rewrite_tac[z_get_specⓩSQRT⌝]);
save_pop_thm"vc3002_3";
```

SML

```
set_goal([], get_conjecture"−""vc3002_4");
a(PC_T1"calc_prelims"cn_vc_simp_tac[]);
a(⇒_tac THEN asm_rewrite_tac[]);
save_pop_thm"vc3002_4";
```

SML

```
set_goal([], get_conjecture"−""vc3002_5");
a(PC_T1"calc_prelims"cn_vc_simp_tac[]);
a(⇒_tac THEN asm_rewrite_tac[]);
save_pop_thm"vc3002_5";
```

SML

```
set_goal([], get_conjecture"-""vc3002_6");
a(PC_T1"calc_prelims"cn_vc_simp_tac[]);
a(⇒_tac THEN asm_rewrite_tac[]);
save_pop_thm"vc3002_6";
```

SML

```
set_goal([], get_conjecture"-""vc3002_7");
a(PC_T1"calc_prelims"cn_vc_simp_tac[]);
a(⇒_tac THEN asm_rewrite_tac[]);
save_pop_thm"vc3002_7";
```

SML

```
set_goal([], get_conjecture"-""vc3002_8");
a(PC_T1"calc_prelims"cn_vc_simp_tac[]);
a(⇒_tac THEN asm_rewrite_tac[]);
save_pop_thm"vc3002_8";
```

SML

```
open_theory "OPSoOPERATION_BUTTONoFACT'func";
set_goal([],get_conjecture"-""vcOPSoOPERATION_BUTTONoFACT_1");
a(cn_vc_simp_tac[]);
a(REPEAT strip_tac THEN all_fc_tac[natural_thm]);
save_pop_thm"vcOPSoOPERATION_BUTTONoFACT_1";
```

SML

```
set_goal([],get_conjecture"-""vcOPSoOPERATION_BUTTONoFACT_2");
a(cn_vc_simp_tac[]);
a(REPEAT strip_tac);
a(all_var_elim_asm_tac1);
save_pop_thm"vcOPSoOPERATION_BUTTONoFACT_2";
```

SML

```
set_goal([],get_conjecture"-""vc1001_1");
a(cn_vc_simp_tac[]);
a(REPEAT strip_tac THEN asm_rewrite_tac[fact_thm]);
save_pop_thm"vc1001_1";
```

SML

```
set_goal([],get_conjecture"-""vc1001_2");
a(cn_vc_simp_tac[]);
a(REPEAT strip_tac);
a(lemma_tac⌜ZM = 0 ∨ M = 1⌝
        THEN1 PC_T1"z_lin_arith"asm_prove_tac[]
        THEN asm_rewrite_tac[fact_thm]);
save_pop_thm"vc1001_2";
```

SML

```
set_goal([],get_conjecture"-""vc1001_3");
a(cn_vc_simp_tac[]);
a(REPEAT strip_tac);
a(asm_ante_tac⌜Z2 ≤ J⌝ THEN PC_T1 "z_lin_arith" prove_tac[]);
save_pop_thm"vc1001_3";
```

SML

```
set_goal([],get_conjecture"-""vc1001_4");
a(cn_vc_simp_tac[]);
save_pop_thm"vc1001_4";
```

SML

```
set_goal([], get_conjecture"-""vc1002_1");
a(cn_vc_simp_tac[]);
a(REPEAT strip_tac THEN all_var_elim_asm_tac1);
a(lemma_tac⌜Z∃K:𝕌• K + 1 = J⌝);
(* *** Goal "1" *** *)
a(z_∃_tac⌜ZJ − 1⌝ THEN PC_T1 "z_lin_arith" prove_tac[]);
(* *** Goal "2" *** *)
a(all_var_elim_asm_tac1);
a(rewrite_tac[z_plus_assoc_thm]);
a(ALL_FC_T rewrite_tac[cn_fact_thm]);
save_pop_thm "vc1002_1";
```

SML

```
open_theory "OPSoOPERATION_BUTTONoSQRT'func";
set_goal([],get_conjecture"-""vcOPSoOPERATION_BUTTONoSQRT_1");
a(cn_vc_simp_tac[]);
save_pop_thm"vcOPSoOPERATION_BUTTONoSQRT_1";
```

SML

```
set_goal([],get_conjecture"-""vcOPSoOPERATION_BUTTONoSQRT_2");
a(cn_vc_simp_tac[]);
a(REPEAT strip_tac THEN all_var_elim_asm_tac1);
save_pop_thm"vcOPSoOPERATION_BUTTONoSQRT_2";
```

SML

```
set_goal([],get_conjecture"-""vc2001_1");
a(cn_vc_simp_tac[]);
save_pop_thm"vc2001_1";
```

SML

```
set_goal([],get_conjecture"-""vc2001_2");
a(cn_vc_simp_tac[]);
save_pop_thm"vc2001_2";
```

SML

```
set_goal([], get_conjecture"-""vc2002_1");
a(cn_vc_simp_tac[]);
a(REPEAT ⇒_tac THEN all_var_elim_asm_tac1);
a(POP_ASM_T discard_tac THEN all_fc_tac[natural_thm]);
a(DROP_NTH_ASM_T 2 discard_tac);
a(asm_rewrite_tac[star_star_2_thm]);
a(z_≤_induction_tacⓩM⌝);
(* *** Goal "1" *** *)
a(rewrite_tac[]);
(* *** Goal "2" *** *)
a(PC_T1 "z_lin_arith" asm_prove_tac[]);
save_pop_thm "vc2002_1";
```

SML

```
set_goal([],get_conjecture"-""vc2002_2");
a(cn_vc_simp_tac[]);
save_pop_thm"vc2002_2";
```

SML

```
set_goal([],get_conjecture"-""vc2002_3");
a(cn_vc_simp_tac[]);
save_pop_thm"vc2002_3";
```

SML

```
set_goal([],get_conjecture"-""vc2003_1");
a(cn_vc_simp_tac[]);
a(REPEAT strip_tac THEN all_var_elim_asm_tac1);
save_pop_thm"vc2003_1";
```

SML

```
set_goal([],get_conjecture"-""vc2003_2");
a(cn_vc_simp_tac[]);
save_pop_thm"vc2003_2";
```

SML

```
set_goal([],get_conjecture"-""vc2003_3");
a(cn_vc_simp_tac[]);
save_pop_thm"vc2003_3";
```

SML

```
set_goal([],get_conjecture"-""vc2004_1");
a(cn_vc_simp_tac[]);
save_pop_thm"vc2004_1";
```

SML

```
set_goal([],get_conjecture"-""vc2004_2");
a(cn_vc_simp_tac[]);
save_pop_thm"vc2004_2";



```