ICMS 2006 Demo

R.D. Arthan
Lemma 1 Ltd.
rda@lemma-one.com

7 September 2006

1 INTRODUCTION

This document contains two proofs in ProofPower of the theorem set as a challenge by Freek Wiedijk at ICMS 2006. Section 2 contains the proof scripts and the rest of the document comprises the customary automatically generated theory listing and index.

The following commands set up a theory to hold the work under the theory of analysis in the ProofPower mathematical case studies:

SML

```
force_delete_theory "icms" handle Fail _ => ();
open_theory "analysis" ;
new_theory "icms";
set_merge_pcs["basic_hol1", "'sets_alg", "'ℤ", "'ℝ"];
```

2 PROOFS

The following ML binding captures the statement to be proved:

SML

```
val main_goal = ⌜
∀f g•
        (∀x y:ℝ•f(x + y) + f(x − y) = ℕℝ 2 * f x * g y)
∧       (∀x•Abs(f x) ≤ ℕℝ 1)
∧       (∃z•¬f z = ℕℝ 0)
⇒       (∀y•Abs(g y) ≤ ℕℝ 1)
⌝;
```

The first proof is the one supplied by Freek. It is broken down here into three lemmas:

SML

```
set_goal([], ⌜
∀f g k x y•
        (∀x y:ℝ•f(x + y) + f(x − y) = ℕℝ 2 * f x * g y)
∧       (∀x•Abs(f x) ≤ k)
⇒       Abs(f x) * Abs(g y) ≤ k
⌝);
a(rewrite_tac[] THEN REPEAT strip_tac
        THEN bc_thm_tac (pc_rule1 "ℝ_lin_arith" prove_rule[]
                ⌜∀a b•ℕℝ 2 * a ≤ ℕℝ 2 * b ⇒ a ≤ b⌝));
a(bc_thm_tac ℝ_≤_trans_thm);
a(∃_tac⌜Abs(f(x + y)) + Abs(f(x + ∼y))⌝ THEN REPEAT strip_tac);
(* *** Goal "1" *** *)
a(bc_thm_tac ℝ_≤_trans_thm);
a(∃_tac⌜Abs(f(x + y) + f(x + ∼y))⌝ THEN REPEAT strip_tac);
(* *** Goal "1.1" *** *)
a(asm_rewrite_tac[ℝ_abs_times_thm]);
(* *** Goal "1.2" *** *)
a(asm_rewrite_tac[ℝ_abs_plus_thm]);
(* *** Goal "2" *** *)
a(bc_thm_tac (pc_rule1 "ℝ_lin_arith" prove_rule[]
                ⌜∀a b•a ≤ k ∧ b ≤ k ⇒ a + b ≤ ℕℝ 2 * k⌝));
a(asm_rewrite_tac[]);
val lemma_1_thm = save_pop_thm "lemma_1_thm";
```

SML

```
set_goal([], ⌜
∀f•     (∀x•Abs(f x) ≤ ℕℝ 1)
⇒       ∃k•     (∀x• Abs(f x) ≤ k)
        ∧       (∀b• (∀x•Abs(f x) ≤ b) ⇒ k ≤ b)
⌝);
a(REPEAT strip_tac);
a(lemma_tac⌜¬{y | ∃x•y = Abs(f x)} = {}⌝);
(* *** Goal "1" *** *)
a(PC_T1 "sets_ext1" once_rewrite_tac[] THEN
        REPEAT strip_tac THEN rewrite_tac[]);
a(prove_tac[]);
(* *** Goal "2" *** *)
a(LEMMA_T⌜∃ a• ∀ x• x ∈ {y | ∃x•y = Abs(f x)} ⇒ x ≤ a⌝ asm_tac);
(* *** Goal "2.1" *** *)
a(∃_tac⌜ℕℝ 1⌝ THEN rewrite_tac[] THEN REPEAT strip_tac THEN
        all_var_elim_asm_tac1);
a(asm_rewrite_tac[]);
```

```
(* *** Goal "2.2" *** *)
a(∃_tac⌜Sup {y | ∃x•y = Abs(f x)}⌝ THEN REPEAT strip_tac);
(* *** Goal "2.2.1" *** *)
a(bc_thm_tac ℝ_≤_sup_bc_thm);
a(asm_rewrite_tac[]);
a(REPEAT strip_tac);
a(POP_ASM_T bc_thm_tac);
a(prove_tac[]);
(* *** Goal "2.2.2" *** *)
a(bc_thm_tac ℝ_sup_≤_bc_thm);
a(asm_rewrite_tac[]);
a(REPEAT strip_tac);
a(asm_rewrite_tac[]);
val lemma_2_thm = save_pop_thm "lemma_2_thm";
```

SML

```
set_goal([], ⌜
∀afx agy k•
        afx * agy ≤ k
∧       ℕℝ 0 < k
∧       ℕℝ 1 < agy
⇒       afx ≤ k * agy ⁻¹
∧       k * agy ⁻¹ < k
⌝);
a(REPEAT ∀_tac THEN ⇒_tac);
a(lemma_tac⌜ℕℝ 0 < agy ∧ ¬agy = ℕℝ 0⌝
        THEN1 PC_T1 "ℝ_lin_arith" asm_prove_tac[]);
a(rewrite_tac[ℝ_≤_¬_less_thm]);
a(ALL_FC_T1 fc_⇔_canon once_rewrite_tac[ℝ_times_mono_⇔_thm]);
a(rewrite_tac[∀_elim⌜k⌝ ℝ_times_order_thm]);
a(ALL_FC_T rewrite_tac[ℝ_recip_clauses]);
a(REPEAT strip_tac THEN1 PC_T1 "ℝ_lin_arith" asm_prove_tac[]);
a(bc_thm_tac(rewrite_rule[] (list_∀_elim[⌜k⌝, ⌜ℕℝ 1⌝, ⌜agy⌝] ℝ_times_mono_thm)));
a(REPEAT strip_tac);
val lemma_3_thm = save_pop_thm "lemma_3_thm";
```

SML

```
set_goal([], main_goal);
a(contr_tac);
a(POP_ASM_T (strip_asm_tac o rewrite_rule[ℝ_¬_≤_less_thm]));
a(all_fc_tac[lemma_2_thm]);
a(lemma_tac⌜Abs(f z) > ℕℝ 0⌝
        THEN1 (rewrite_tac[ℝ_abs_def]
```

```
                THEN cases_tac⌜ℕℝ 0 ≤ f z⌝
                THEN asm_rewrite_tac[]
                THEN PC_T1 "ℝ_lin_arith" asm_prove_tac[]));
a(lemma_tac⌜ℕℝ 0 < k⌝
        THEN1 (spec_nth_asm_tac 3 ⌜z⌝
                THEN PC_T1 "ℝ_lin_arith" asm_prove_tac[]));
a(lemma_tac⌜ Abs(f z) * Abs(g y) ≤ k ⌝ THEN1
        (bc_thm_tac lemma_1_thm THEN asm_rewrite_tac[]));
a(lemma_tac⌜ k * Abs(g y) ⁻¹ < k ⌝ THEN1
        all_fc_tac[lemma_3_thm]);
a(lemma_tac⌜∀x•Abs(f x) ≤ k * Abs (g y) ⁻¹⌝ THEN1 REPEAT strip_tac);
(* *** Goal "1" *** *)
a(lemma_tac⌜ Abs(f x) * Abs(g y) ≤ k ⌝ THEN1
        (bc_thm_tac lemma_1_thm THEN asm_rewrite_tac[]));
a(all_fc_tac[lemma_3_thm]);
(* *** Goal "2" *** *)
a(LIST_DROP_NTH_ASM_T[6]all_fc_tac);
a(PC_T1 "ℝ_lin_arith" asm_prove_tac[]);
val icms_thm = save_pop_thm "icms_thm";
```

The second proof was pointed out by Tom Hales during the ICMS session. It is based on the following lemma.

SML

```
set_goal([], ⌜
∀f g•
        (∀x y:ℝ•f(x + y) + f(x − y) = ℕℝ 2 * f x * g y)
∧       (∀x•Abs(f x) ≤ ℕℝ 1)
⇒       (∀m:ℕ; x y•Abs(f x)*Abs(g y)^m ≤ ℕℝ 1)
⌝);
a(REPEAT ∀_tac THEN ⇒_tac THEN ∀_tac THEN induction_tac⌜m:ℕ⌝);
(* *** Goal "1" *** *)
a(asm_rewrite_tac[]);
(* *** Goal "2" *** *)
a(REPEAT strip_tac THEN asm_rewrite_tac[ℝ_ℕ_exp_def, ℝ_abs_times_thm]);
a(rewrite_tac[ℝ_times_assoc_thm1]);
a(LEMMA_T⌜Abs(f x) * Abs(g y) = (1/2) * Abs(ℕℝ 2 * f x * g y)⌝
        rewrite_thm_tac
        THEN1 (rewrite_tac[ℝ_abs_times_thm]
                THEN conv_tac(RIGHT_C ℝ_anf_conv)
                THEN strip_tac));
a(DROP_NTH_ASM_T 3 (rewrite_thm_tac o
                conv_rule(ONCE_MAP_C eq_sym_conv)));
a(rewrite_tac [conv_rule(ONCE_MAP_C eq_sym_conv) ℝ_abs_times_thm,
```

```
        conv_rule(ONCE_MAP_C eq_sym_conv) ℝ_abs_ℝ_ℕ_exp_thm,
        ℝ_times_assoc_thm,
        pc_rule1 "ℝ_lin_arith" prove_rule[]
        ⌜∀x y•(1/2) * x ≤ y ⇔ x ≤ ℕℝ 2 * y⌝]);
a(rewrite_tac [ℝ_times_plus_distrib_thm]);
a(bc_thm_tac ℝ_≤_trans_thm
        THEN ∃_tac ⌜Abs (f (x + y) * g y ^ m) + Abs(f (x + ~ y) * g y ^ m)⌝);
a(rewrite_tac [ℝ_abs_plus_thm]);
a(bc_thm_tac (pc_rule1 "ℝ_lin_arith" prove_rule[]
                ⌜∀a b•a ≤ ℕℝ 1 ∧ b ≤ ℕℝ 1 ⇒ a + b ≤ ℕℝ 2⌝)
        THEN asm_rewrite_tac[ℝ_abs_times_thm, ℝ_abs_ℝ_ℕ_exp_thm]);
val TH_lemma_1_thm = save_pop_thm "TH_lemma_1_thm";
```

SML

```
set_goal([], main_goal);
a(contr_tac);
a(POP_ASM_T (strip_asm_tac o rewrite_rule[ℝ_¬_≤_less_thm]));
a(lemma_tac⌜Abs(f z) > ℕℝ 0 ∧ ¬Abs(f z) = ℕℝ 0⌝
        THEN1 (rewrite_tac[ℝ_abs_def]
                THEN cases_tac⌜ℕℝ 0 ≤ f z⌝
                THEN asm_rewrite_tac[]
                THEN PC_T1 "ℝ_lin_arith" asm_prove_tac[]));
a(lemma_tac⌜∃m:ℕ•Abs(f z) ⁻¹ < Abs(g y) ^ m⌝
        THEN1 (bc_thm_tac ℝ_ℕ_exp_tends_to_infinity_thm THEN REPEAT strip_tac));
a(swap_nth_asm_concl_tac 1);
a(ante_tac(∀_elim⌜Abs(f z)⌝ ℝ_times_mono_⇔_thm));
a(asm_rewrite_tac[]);
a(STRIP_T once_rewrite_thm_tac);
a(ALL_FC_T rewrite_tac[ℝ_recip_clauses]);
a(rewrite_tac[ℝ_¬_less_≤_thm]);
a(ALL_FC_T rewrite_tac[TH_lemma_1_thm]);
val TH_icms_thm = save_pop_thm "TH_icms_thm";
```

3 THE THEORY icms

3.1 Parents

analysis

3.2 Theorems

lemma_1_thm $\vdash \forall\ f\ g\ k\ x\ y$
$\bullet\ (\forall\ x\ y \bullet\ f\ (x + y) + f\ (x - y) = \mathbb{NR}\ 2 * f\ x * g\ y)$
$\wedge\ (\forall\ x \bullet\ Abs\ (f\ x) \leq k)$
$\Rightarrow Abs\ (f\ x) * Abs\ (g\ y) \leq k$

lemma_2_thm $\vdash \forall\ f$
$\bullet\ (\forall\ x \bullet\ Abs\ (f\ x) \leq \mathbb{NR}\ 1)$
$\Rightarrow (\exists\ k$
$\bullet\ (\forall\ x \bullet\ Abs\ (f\ x) \leq k)$
$\wedge\ (\forall\ b \bullet\ (\forall\ x \bullet\ Abs\ (f\ x) \leq b) \Rightarrow k \leq b))$

lemma_3_thm $\vdash \forall\ afx\ agy\ k$
$\bullet\ afx * agy \leq k \wedge \mathbb{NR}\ 0 < k \wedge \mathbb{NR}\ 1 < agy$
$\Rightarrow afx \leq k * agy^{-1} \wedge k * agy^{-1} < k$

icms_thm $\vdash \forall\ f\ g$
$\bullet\ (\forall\ x\ y \bullet\ f\ (x + y) + f\ (x - y) = \mathbb{NR}\ 2 * f\ x * g\ y)$
$\wedge\ (\forall\ x \bullet\ Abs\ (f\ x) \leq \mathbb{NR}\ 1)$
$\wedge\ (\exists\ z \bullet\ \neg\ f\ z = \mathbb{NR}\ 0)$
$\Rightarrow (\forall\ y \bullet\ Abs\ (g\ y) \leq \mathbb{NR}\ 1)$

TH_lemma_1_thm
$\vdash \forall\ f\ g$
$\bullet\ (\forall\ x\ y \bullet\ f\ (x + y) + f\ (x - y) = \mathbb{NR}\ 2 * f\ x * g\ y)$
$\wedge\ (\forall\ x \bullet\ Abs\ (f\ x) \leq \mathbb{NR}\ 1)$
$\Rightarrow (\forall\ m\ x\ y \bullet\ Abs\ (f\ x) * Abs\ (g\ y)\ \hat{}\ m \leq \mathbb{NR}\ 1)$

TH_icms_thm $\vdash \forall\ f\ g$
$\bullet\ (\forall\ x\ y \bullet\ f\ (x + y) + f\ (x - y) = \mathbb{NR}\ 2 * f\ x * g\ y)$
$\wedge\ (\forall\ x \bullet\ Abs\ (f\ x) \leq \mathbb{NR}\ 1)$
$\wedge\ (\exists\ z \bullet\ \neg\ f\ z = \mathbb{NR}\ 0)$
$\Rightarrow (\forall\ y \bullet\ Abs\ (g\ y) \leq \mathbb{NR}\ 1)$

4 INDEX