1 Series I

theory *SeriesI*
imports
Complex-Main
HOL−Analysis.Analysis
begin

1.1 Problem 1

Solve the equation in the integers:

theorem *problem1*:
fixes $x\ y :: int$
assumes $x \neq 0$ **and** $y \neq 0$
shows $1\ /\ x^2 + 1\ /\ (x*y) + 1\ /\ y^2 = 1$
$\longleftrightarrow x = 1 \wedge y = -1 \vee x = -1 \wedge y = 1$
(**is** $?eqn \longleftrightarrow ?sols$)
proof
— Unfortunately, removing the conversions between int and real takes a few lines
let $?x = \textit{real-of-int}\ x$ **and** $?y = \textit{real-of-int}\ y$
assume $?eqn$
then have $1\ /\ ?x^2 + 1\ /\ (?x*?y) + 1\ /\ ?y^2 = 1$ **by** *auto*
hence $?x^2*?y^2\ /\ ?x^2 + ?x^2*?y^2\ /\ (?x*?y) + ?x^2*?y^2\ /\ ?y^2 = ?x^2*?y^2$
by *algebra*
hence $?x^2 + ?x*?y + ?y^2 = ?x^2 * ?y^2$ **using** ‹$x \neq 0$› ‹$y \neq 0$›
by (*simp add*: *power2-eq-square*)
hence *inteq*: $x^2 + x*y + y^2 = x^2 * y^2$
using *of-int-eq-iff* **by** *fastforce*

define g **where** $g = gcd\ x\ y$
then have $g \neq 0$ **and** $g > 0$ **using** ‹$x \neq 0$› ‹$y \neq 0$› **by** *auto*
define $x'\ y'$ **where** $x' = x\ div\ g$ **and** $y' = y\ div\ g$
then have $x' * g = x$ **and** $y' * g = y$ **using** *g-def* **by** *auto*
from *inteq* **and** *this* **have** $g^2 * (x'^2 + x' * y' + y'^2) = x'^2 * y'^2 * g\hat{}4$
by *algebra*
hence *reduced*: $x'^2 + x' * y' + y'^2 = x'^2 * y'^2 * g^2$ **using** ‹$g \neq 0$› **by** *algebra*

hence $x'\ dvd\ y'^2$ **and** $y'\ dvd\ x'^2$
by *algebra+*
moreover have $coprime\ x'\ (y'^2)\ coprime\ (x'^2)\ y'$
unfolding *x′-def y′-def g-def*
using *assms div-gcd-coprime* **by** *auto*
ultimately have $is\text{-}unit\ x'\ is\text{-}unit\ y'$
unfolding *coprime-def* **by** *auto*
hence *abs1*: $|x'| = 1 \wedge |y'| = 1$ **using** *assms* **by** *auto*
then consider (*same-sign*) $x' = y'$ | (*diff-sign*) $x' = -y'$ **by** *fastforce*
thus $?sols$
proof *cases*
case *same-sign*

```
    then have x' * y' = 1
      using abs1 and zmult-eq-1-iff by fastforce
    hence g^2 = 3
      using abs1 same-sign and reduced by algebra
    hence 1^2 < g^2 and g^2 < 2^2 by auto
    hence 1 < g and g < 2
      using ‹g > 0› and power2-less-imp-less by fastforce+
    hence False by auto
    thus ?sols by auto
  next
    case diff-sign
    then have x' * y' = −1
      using abs1
      by (smt mult-cancel-left2 mult-cancel-right2)
    hence g^2 = 1
      using abs1 diff-sign and reduced by algebra
    hence g = 1 using ‹g > 0›
      by (smt power2-eq-1-iff)
    hence x = x' and y = y'
      unfolding x'-def and y'-def by auto
    thus ?sols using abs1 and diff-sign by auto
  qed
next
  assume ?sols
  then show ?eqn by auto
qed
```

1.2 Problem 2

Prove that a sequence is bounded, converges, and find the limit.

```
context
  fixes a :: real
  assumes a-bounds: 0 < a a < 1
begin
fun c :: nat ⇒ real where
c 0 = a / 2 |
c (Suc n) = (a + (c n)^2) / 2

abbreviation x1 ≡ 1 − sqrt (1 − a)
abbreviation x2 ≡ 1 + sqrt (1 − a)

lemma c-pos: 0 < c n
  using a-bounds
  by (induction n, auto, smt zero-less-power)

lemma c-bounded: c n < x1
proof (induction n)
  case 0
  have (1 − a/2)^2 = 1 − a + (a/2)^2
```

```
    by (simp add: power2-diff)
  hence 1 − a < (1 − a/2)^2 using a-bounds by auto
  hence sqrt (1 − a) < 1 − a/2
    using a-bounds and real-less-lsqrt by auto
  thus ?case by auto
next
  case (Suc n)
  then have (c n)^2 < (1 − sqrt (1−a))^2 using c-pos
    by (smt power-less-imp-less-base real-sqrt-abs)
  also have ... = 2 − 2 ∗ sqrt (1−a) − a
    using a-bounds by (simp add: power2-diff)
  finally have (a + (c n)^2)/2 < 1 − sqrt (1−a) by auto
  then show ?case by auto
qed

lemma c-incseq: incseq c
proof (rule incseq-SucI)
  fix n
  from c-bounded have c n < x1 by auto
  have c n < x1 c n < x2
    using c-bounded
    by (smt a-bounds real-sqrt-lt-0-iff)+
  moreover have (c n)^2 − 2∗c n + a = (c n − x1)∗(c n − x2)
    using a-bounds
    by (auto simp add: algebra-simps power2-eq-square)
  ultimately have (c n)^2 − 2∗c n + a > 0
    by (smt nonzero-mult-div-cancel-right zero-le-divide-iff)
  thus c n ≤ c (Suc n) by auto
qed

theorem problem2: c ⟶ x1
proof −
  obtain L where c ⟶ L
    using c-incseq c-bounded incseq-convergent
    by (metis less-imp-le)
  then have (λn. c (Suc n)) ⟶ L
    using LIMSEQ-Suc by blast
  hence (λn. (a + (c n)^2) / 2 ∗ 2) ⟶ L∗2
    using tendsto-mult-right by fastforce
  hence (λn. a + (c n)^2) ⟶ L∗2 by (simp del: distrib-right-numeral)
  hence (λn. a + (c n)^2 − a) ⟶ L∗2 − a
    using tendsto-diff LIMSEQ-const-iff by blast
  hence (λn. (c n)^2) ⟶ L∗2 − a
    by auto
  moreover from ‹c ⟶ L›
  have (λn. (c n)^2) ⟶ L^2
    unfolding power2-eq-square
    using tendsto-mult by blast
  ultimately have L∗2 − a = L^2
```

by (*rule LIMSEQ-unique*)
hence $L^2 - 2*L + a = 0$ **by** *auto*
moreover have $L^2 - 2*L + a = (L - x1)*(L - x2)$
using *a-bounds*
by (*auto simp add*: *algebra-simps power2-eq-square*)
ultimately have $L = x1 \vee L = x2$
by *auto*
moreover from *c-bounded* **and** $\langle c \longrightarrow L \rangle$ **have** $L \leq x1$
by (*meson LIMSEQ-le-const2 le-less-linear less-imp-triv*)
moreover from *a-bounds* **have** $x1 < x2$ **by** *auto*
ultimately have $L = x1$ **by** *auto*
thus *?thesis* **using** $\langle c \longrightarrow L \rangle$ **by** *auto*
qed

end

end