Contents

Title 1

Table of Contents ii

Theory pimGreI 1

101 — Introduction — [1.2] 1

102 — A First Lemma — [1.3] 1

102.1 — Two-Two — [2.4] . . . 1

...Le.102.1.1....(two-two)......$2 + 2 = 4$. ...[2.4.a] . . . 1

...Prf.102.1.1.1......By definition of the natural numbers, I suppose. $\mathbf{\Pi}(\mathbf{\Lambda})$. ...[2.4.a] . . 1

103 — Main Notations — [4.4] 1

103.1 — apply-inverseG — [4.4] . . . 1

...Le.103.1.1....(apply-inverseG)......If $f(x) = u$, and for every x in P, $g(f(x)) = x$, and x is in P, then $g(u) = x$. ...[5.4.a] . . . 1

...Prf.103.1.1.1......By auto, for some underlying reason. $\mathbf{\Pi}(\mathbf{\Lambda})$. ...[5.4.a] . . . 1

103.2 — idG — [6.1] . . . 1

...Df.103.2.1....(idG)......The identity function idG, is defined to be $\mathrm{idG}(x) = x$. ...[6.2.a] 1

103.3 — nondecreasing-on — [6.3] . . . 2

...Le.103.3.1....(nondecreasing-on)...... ...[6.3.a] . . . 2

104 — Automatic Proofs — [7.3] 2

104.1 — idG-apply — [7.4] . . . 2

...Le.104.1.1....(idG-apply)......$\mathrm{idG}(x) = x$. ...[7.4.a] . . . 2

...Prf.104.1.1.1......By simple application of of the definition of idG. $\mathbf{\Pi}(\mathbf{\Lambda})$. ...[7.4.a] . . 2

...Le.104.1.2....(id-applyMetis)......Here, it's shown that id-apply can be proved with metis. Isabelle can't tell that $x = \mathrm{id}(x)$ is equivalent to $\mathrm{id}(x) = x$. ...[8.3.a] . . . 2

105 — Interactive Proof — [9.3] 2

105.1 — sum-square — [10.2] . . . 2

...Le.105.1.1....(sum-square)......We have the equality, $(a + b)^2 = a^2 + 2ab + b^2$. ...[9.3.a] 2

...Prf.105.1.1.1......We have $x^2 = xx$, so we can expand each square in the equation above. ...[9.3.a] [...] . . . 2

105.2 — expression-nonneg — [11.2] . . . 3

...Le.105.2.1....(expression-nonneg)......We have $x^2 + 6x + 9 \geq 0$. ...[11.2.a] . . . 3

...Prf.105.2.1.1......Factoring, we get $x^2 + 6x + 9 = (x + 3)^2$. ...[11.2.a] [...] . . . 3

106 — Assumptions and Local Variables — [13.3] 3

106.1 — interior-ball — [13.4] . . . 3

...Le.106.1.1....(interior-ball)......The element x is in the interior of S if and only if there exists an $e > 0$ such that $B(x, e) \subseteq S$. ...[13.4.a] . . . 3

...Le.106.1.2....(mem-interior-O)......Proof of i06MvTES.thy `mem-interior-O`. ...[15.1.a] 4

106.2 — Fix, Equivalent Interior Def — [15.2] . . . 4

...Le.106.2.1....(st)......Proving something about set equality. I don't know what what the notation means. ...[15.2.a] 4
...Le.106.2.2....(interior-def2)......The interior of S is equal to the set of all x such that there exists an $e > 0$ with $B(x, e) \subseteq S$. ...[15.3.a] 4
...Le.106.2.3....(interior-def3)......Using `fix`, I prove RHS and LHS are subsets of each other, but I don't know how to get equality yet. ...[15.3.b] 4

107 — Introducing New Notations and Concepts — [16.1] **5**
107.1 — Defining New Notations — [16.1] 5

[—pimGreI.thy—] **6**

[—pimGreI.thy <...>—] **9**

Theory pimGreI

```
(*header{*Isabelle Primer for Mathematicians*}*)
theory pimGreI
imports "~~/src/HOL/Multivariate_Analysis/Multivariate_Analysis"
begin
sledgehammer_params [provers = "e cvc3 metis metis_full_types metis_no_types
  spass z3 leo2 satallax z3_tptp", slicing=true, verbose=true,
  isar_proof=true]
```

101 — Introduction — [1.2]

102 — A First Lemma — [1.3]

102.1 — Two-Two — [2.4]

Λ.Lemma. 102.1.1 (*two-two*). $2 + 2 = 4$. ❐

```
lemma two_two:
  "(2::nat)+2=4"
```

Π.Proof. 102.1.1.1. By definition of the natural numbers, I suppose. **Π(Λ)**.

```
  apply auto
  done
```

103 — Main Notations — [4.4]

103.1 — apply-inverseG — [4.4]

Λ.Lemma. 103.1.1 (*apply-inverseG*). If $f(x) = u$, and for every x in P, $g(f(x)) = x$, and x is in P, then $g(u) = x$. ❐

```
lemma apply_inverseG:
  "f x = u ==> (⋀x. x ∈ P ==> g (f x) = x)
           ==> x ∈ P
           ==> g u = x"
```

Π.Proof. 103.1.1.1. By auto, for some underlying reason. **Π(Λ)**.

```
  by auto
```

103.2 — idG — [6.1]

Δ.Definition. 103.2.1 (*idG*). The identity function idG, is defined to be $\text{idG}(x) = x$. ❐

```
definition idG::
  "'a => 'a" where "idG = (λx. x)"
```

103.3 — nondecreasing-on — [6.3]

Λ.LEMMA. 103.3.1 (*nondecreasing-on*). ❐

```
definition nondecreasing_on::
  "real set => (real => real) => bool"
    where "nondecreasing_on S f <->
    (∀x∈S. ∀y∈S. x ≤ y --> f x ≤ f y)"
```

104 — Automatic Proofs — [7.3]

104.1 — idG-apply — [7.4]

Λ.LEMMA. 104.1.1 (*idG-apply*). $\mathrm{idG}(x) = x$. ❐

```
lemma idG_apply:
  "idG y = y"
```

Π.PROOF. 104.1.1.1. By simple application of of the definition of idG. **Π(Λ)**.

```
  apply (simp add: idG_def)
  --"Here is the one-line proof of piMisGre's apply-id:"
    --"using Fun.id-apply by auto"
  --"I can't use it because Fun.id-apply is proved with ‘by (simp add: id-def)’."
  --"My lemma says that idG(y)=y, not that id(y)=y."
  done
```

Λ.LEMMA. 104.1.2 (*id-applyMetis*). Here, it's shown that id-apply can be proved with metis. Isabelle can't tell that $x = \mathrm{id}(x)$ is equivalent to $\mathrm{id}(x) = x$. ❐

```
lemma id_applyMetis:
  "x = id x"
  apply (metis Fun.id_apply)
  done
```

105 — Interactive Proof — [9.3]

105.1 — sum-square — [10.2]

Λ.LEMMA. 105.1.1 (*sum-square*). We have the equality, $(a + b)^2 = a^2 + 2ab + b^2$. ❐

```
lemma sum_square:
  "(a+b)^2 = a^2+(2::real)*a*b+b^2"
```

Π.PROOF. 105.1.1.1. We have $x^2 = xx$, so we can expand each square in the equation above.

```
  apply (simp add: Nat_Numeral.power2_eq_square)
```

This gives $(a+b)(a+b) = aa+2ab+bb$, which merely needs to be simplified with basic algebra. **Π(Λ)**.

```
  apply (simp add: Groups.algebra_simps)
  done
```

105.2 — expression-nonneg — [11.2]

Λ.Lemma. 105.2.1 (*expression-nonneg*). We have $x^2 + 6x + 9 \geq 0$. ❐

```
lemma expression_nonneg:
  "x^2 + (6::real)*x + 9 >=0"
```

Π.Proof. 105.2.1.1. Factoring, we get $x^2 + 6x + 9 = (x + 3)^2$.

```
  proof-
  have aux1: "x^2 + (6::real)*x + 9 = (x+3)^2"
    using sum_square [of x 3]
    by auto
```

We also have $a^2 \geq 0$, so $(x + 3)^2 \geq 0$.

```
  have aux2: "(x+3)^2 ≥ 0"
    using Rings.zero_le_square [of "x+3"]
    by auto
```

By *aux1* and *aux2*, the result is proved. **Π(Λ)**.

```
  thus ?thesis using aux1 aux2
    by auto
  qed
```

106 — Assumptions and Local Variables — [13.3]

106.1 — interior-ball — [13.4]

Λ.Lemma. 106.1.1 (*interior-ball*). The element x is in the interior of S if and only if there exists an $e > 0$ such that $B(x, e) \subseteq S$. ❐

```
lemma interior_ball:
    "x ∈ (interior S) ⟷ (∃e > 0. (ball x e) ⊆ S)"
  proof-
  {
  assume "x ∈ (interior S)" (*
                  [π] ⟶ (∃e > 0. (ball x e) ⊆ S) *)
    then obtain T where T_def: "x ∈ T ∧ open T ∧ T ⊆ S"
                    using interior_def by auto (*
                  [1] Df...interior S = ⋃{T. open T ∧ T ⊆ S}
                  [␣] S° is the union of all open sets in S. Because x is
                      in this union, there exists an open T ⊆ S, such that
                      x ∈ T. *)
    hence "∃e > 0. (ball x e) ⊆ T"
                    using open_contains_ball by auto (*
                  [2] Le...open_contains_ball:
                        open S ⟷ (∀x∈S. ∃e>0. ball x e ⊆ S)
                  [␣] Open sets contain open balls. Because T is open and x
                      is in T, then T contains an open ball B(x,e) such that
                      B(x,e) ⊆ T. *)
    hence "∃e > 0. (ball x e) ⊆ S"
                    using T_def by auto
    } note LeftToRight = this
```

```
  {
  assume "∃e > 0. (ball x e) ⊆ S" (*
                  [π] ⟶ x ∈ (interior S) *)
    then obtain e where e_def: "e > 0 ∧ (ball x e) ⊆ S"
                      by auto
    def T ≡ "ball x e"
    hence "x ∈ T ∧ open T ∧ T ⊆ S"
                      using open_ball e_def by auto (*
                  [3] Le...open_ball[intro, simp]: open (ball x e)
                  [␣] Balls are open, so T = B(x,e) is open, and x ∈ T ⊆ S.
                      *)
    hence "x ∈ interior S"
                      using interior_def by auto
    } note RightToLeft = this
  thus ?thesis
                      using LeftToRight by auto
  qed
```

Λ.Lemma. 106.1.2 (*mem-interior-O*). Proof of i06MvTES.thy `mem-interior-0`. ❐

```
lemma mem_interior_0:
    "x ∈ interior S ⟷ (∃e>0. ball x e ⊆ S)"
                      using open_contains_ball_eq [where S="interior S"]
                      by (simp add: open_subset_interior)
```

106.2 — Fix, Equivalent Interior Def — [15.2]

Λ.Lemma. 106.2.1 (*st*). Proving something about set equality. I don't know what what the notation means. ❐

```
lemma st:
    "(S::'a set) = T"
  proof-
    {fix x assume "x:S" hence "x:T" sorry} note imp1 = this
    {fix x assume "x:T" hence "x:S" sorry}
  from this show ?thesis
                      using imp1 by auto
  qed
```

Λ.Lemma. 106.2.2 (*interior-def2*). The interior of S is equal to the set of all x such that there exists an $e > 0$ with $B(x, e) \subseteq S$. ❐

```
lemma interior_def2:
    "interior S = {x. ∃e>0. (ball x e) ⊆ S}"
                      using interior_ball[of _ S] by auto
                      (* [of _ S] Specifies fixed S, arbitrary x *)
```

Λ.Lemma. 106.2.3 (*interior-def3*). Using `fix`, I prove RHS and LHS are subsets of each other, but I don't know how to get equality yet. ❐

```
lemma interior_def_LHS:
    "interior S ⊆ {x. ∃e>0. (ball x e) ⊆ S}"
  proof-
```

```
    {
    fix y assume "y ∈ interior S"
      hence "∃e>0. (ball y e) ⊆ S"
                            using interior_ball by auto
      hence "y ∈ {x. ∃e>0. ball x e ⊆ S}"
                            by auto
      } note LHS = this
    thus ?thesis
                            using LHS by auto
    qed
  lemma interior_def_RHS:
      "{x. ∃e>0. (ball x e) ⊆ S} ⊆ interior S"
    proof-
    {
    fix y assume "y ∈ {x. ∃e>0. (ball x e) ⊆ S}"
      hence "∃e>0. ball y e ⊆ S"
                            by auto
      hence "y ∈ interior S"
                            using interior_ball by auto
      } note RHS = this
    thus ?thesis
                            using RHS by auto
    qed
```

107 — Introducing New Notations and Concepts — [16.1]

107.1 — Defining New Notations — [16.1]

```
  end
```

[—pimGreI.thy—]

```
(*header{*Isabelle Primer for Mathematicians*}*)
theory pimGreI
imports "~~/src/HOL/Multivariate_Analysis/Multivariate_Analysis"
begin
sledgehammer_params [provers = "e cvc3 metis metis_full_types metis_no_types
  spass z3 leo2 satallax z3_tptp", slicing=true, verbose=true,
  isar_proof=true]
lemma two_two:
  "(2::nat)+2=4"
  apply auto
  done
lemma apply_inverseG:
  "f x = u ==> (⋀x. x ∈ P ==> g (f x) = x)
            ==> x ∈ P
            ==> g u = x"
  by auto
definition idG::
  "'a => 'a" where "idG = (λx. x)"
definition nondecreasing_on::
  "real set => (real => real) => bool"
    where "nondecreasing_on S f <->
    (∀x∈S. ∀y∈S. x ≤ y --> f x ≤ f y)"
lemma idG_apply:
  "idG y = y"
  apply (simp add: idG_def)
  --"Here is the one-line proof of piMisGre's apply-id:"
    --"using Fun.id-apply by auto"
  --"I can't use it because Fun.id-apply is proved with ‘by (simp add: id-def)’."
  --"My lemma says that idG(y)=y, not that id(y)=y."
  done
lemma id_applyMetis:
  "x = id x"
  apply (metis Fun.id_apply)
  done
lemma sum_square:
  "(a+b)^2 = a^2+(2::real)*a*b+b^2"
  apply (simp add: Nat_Numeral.power2_eq_square)
  apply (simp add: Groups.algebra_simps)
  done
lemma expression_nonneg:
  "x^2 + (6::real)*x + 9 >=0"
  proof-
  have aux1: "x^2 + (6::real)*x + 9 = (x+3)^2"
    using sum_square [of x 3]
    by auto
  have aux2: "(x+3)^2 ≥ 0"
    using Rings.zero_le_square [of "x+3"]
    by auto
  thus ?thesis using aux1 aux2
    by auto
  qed
lemma interior_ball:
    "x ∈ (interior S) ⟷ (∃e > 0. (ball x e) ⊆ S)"
  proof-
  {
  assume "x ∈ (interior S)" (*
                  [π] ⟶ (∃e > 0. (ball x e) ⊆ S) *)
```

```
  then obtain T where T_def: "x ∈ T ∧ open T ∧ T ⊆ S"
            using interior_def by auto (*
          [1] Df...interior S = ⋃{T. open T ∧ T ⊆ S}
          [␣] S° is the union of all open sets in S. Because x is
            in this union, there exists an open T ⊆ S, such that
            x ∈ T. *)
  hence "∃e > 0. (ball x e) ⊆ T"
            using open_contains_ball by auto (*
          [2] Le...open_contains_ball:
             open S ⟷ (∀x∈S. ∃e>0. ball x e ⊆ S)
          [␣] Open sets contain open balls. Because T is open and x
            is in T, then T contains an open ball B(x,e) such that
            B(x,e) ⊆ T. *)
  hence "∃e > 0. (ball x e) ⊆ S"
            using T_def by auto
  } note LeftToRight = this
 {
 assume "∃e > 0. (ball x e) ⊆ S" (*
          [π] ⟶ x ∈ (interior S) *)
  then obtain e where e_def: "e > 0 ∧ (ball x e) ⊆ S"
            by auto
  def T ≡ "ball x e"
  hence "x ∈ T ∧ open T ∧ T ⊆ S"
            using open_ball e_def by auto (*
          [3] Le...open_ball[intro, simp]: open (ball x e)
          [␣] Balls are open, so T = B(x,e) is open, and x ∈ T ⊆ S.
            *)
  hence "x ∈ interior S"
            using interior_def by auto
  } note RightToLeft = this
 thus ?thesis
            using LeftToRight by auto
 qed
lemma mem_interior_0:
  "x ∈ interior S ⟷ (∃e>0. ball x e ⊆ S)"
            using open_contains_ball_eq [where S="interior S"]
            by (simp add: open_subset_interior)
lemma st:
  "(S::'a set) = T"
 proof-
  {fix x assume "x:S" hence "x:T" sorry} note imp1 = this
  {fix x assume "x:T" hence "x:S" sorry}
 from this show ?thesis
            using imp1 by auto
 qed
lemma interior_def2:
  "interior S = {x. ∃e>0. (ball x e) ⊆ S}"
            using interior_ball[of _ S] by auto
            (* [of _ S] Specifies fixed S, arbitrary x *)
lemma interior_def_LHS:
  "interior S ⊆ {x. ∃e>0. (ball x e) ⊆ S}"
 proof-
 {
 fix y assume "y ∈ interior S"
  hence "∃e>0. (ball y e) ⊆ S"
            using interior_ball by auto
  hence "y ∈ {x. ∃e>0. ball x e ⊆ S}"
            by auto
  } note LHS = this
```

```
  thus ?thesis
                      using LHS by auto
  qed
lemma interior_def_RHS:
    "{x. ∃e>0. (ball x e) ⊆ S} ⊆ interior S"
  proof-
  {
  fix y assume "y ∈ {x. ∃e>0. (ball x e) ⊆ S}"
    hence "∃e>0. ball y e ⊆ S"
                      by auto
    hence "y ∈ interior S"
                      using interior_ball by auto
    } note RHS = this
  thus ?thesis
                      using RHS by auto
  qed
end
```

[—pimGreI.thy <...>—]

```
(*header{*Isabelle Primer for Mathematicians*}*)
theory pimGreI
imports "~~/src/HOL/Multivariate_Analysis/Multivariate_Analysis"
begin
sledgehammer_params [provers = "e cvc3 metis metis_full_types metis_no_types
  spass z3 leo2 satallax z3_tptp", slicing=true, verbose=true,
  isar_proof=true]
lemma two_two:
  "(2::nat)+2=4"
  apply auto
  done
lemma apply_inverseG:
  "f x = u ==> (\<And>x. x \<in> P ==> g (f x) = x)
          ==> x \<in> P
          ==> g u = x"
  by auto
definition idG::
  "'a => 'a" where "idG = (\<lambda>x. x)"
definition nondecreasing_on::
  "real set => (real => real) => bool"
    where "nondecreasing_on S f <->
    (\<forall>x\<in>S. \<forall>y\<in>S. x \<le> y --> f x \<le> f y)"
lemma idG_apply:
  "idG y = y"
  apply (simp add: idG_def)
  --"Here is the one-line proof of piMisGre's apply-id:"
    --"using Fun.id-apply by auto"
  --"I can't use it because Fun.id-apply is proved with 'by (simp add: id-def)'."
  --"My lemma says that idG(y)=y, not that id(y)=y."
  done
lemma id_applyMetis:
  "x = id x"
  apply (metis Fun.id_apply)
  done
lemma sum_square:
  "(a+b)^2 = a^2+(2::real)*a*b+b^2"
  apply (simp add: Nat_Numeral.power2_eq_square)
  apply (simp add: Groups.algebra_simps)
  done
lemma expression_nonneg:
  "x^2 + (6::real)*x + 9 >=0"
  proof-
  have aux1: "x^2 + (6::real)*x + 9 = (x+3)^2"
    using sum_square [of x 3]
    by auto
  have aux2: "(x+3)^2 \<ge> 0"
    using Rings.zero_le_square [of "x+3"]
    by auto
  thus ?thesis using aux1 aux2
    by auto
  qed
lemma interior_ball:
    "x \<in> (interior S) \<longleftrightarrow> (\<exists>e > 0. (ball x e) \<subseteq> S)"
  proof-
  {
  assume "x \<in> (interior S)" (*
                    [\<pi>] \<longrightarrow> (\<exists>e > 0. (ball x e) \<subseteq> S) *)
    then obtain T where T_def: "x \<in> T \<and> open T \<and> T \<subseteq> S"
                        using interior_def by auto (*
                    [1] Df...interior S = \<Union>{T. open T \<and> T \<subseteq> S}
                    [\<spacespace>] S\<^sup>\<circ> is the union of all open sets in S. Because x is
                        in this union, there exists an open T \<subseteq> S, such that
                        x \<in> T. *)
    hence "\<exists>e > 0. (ball x e) \<subseteq> T"
                        using open_contains_ball by auto (*
                    [2] Le...open_contains_ball:
                          open S \<longleftrightarrow> (\<forall>x\<in>S. \<exists>e>0. ball x e \<subseteq> S)
                    [\<spacespace>] Open sets contain open balls. Because T is open and x
                        is in T, then T contains an open ball B(x,e) such that
                        B(x,e) \<subseteq> T. *)
    hence "\<exists>e > 0. (ball x e) \<subseteq> S"
                        using T_def by auto
    } note LeftToRight = this
  {
  assume "\<exists>e > 0. (ball x e) \<subseteq> S" (*
                    [\<pi>] \<longrightarrow> x \<in> (interior S) *)
    then obtain e where e_def: "e > 0 \<and> (ball x e) \<subseteq> S"
                        by auto
    def T \<equiv> "ball x e"
    hence "x \<in> T \<and> open T \<and> T \<subseteq> S"
                        using open_ball e_def by auto (*
                    [3] Le...open_ball[intro, simp]: open (ball x e)
                    [\<spacespace>] Balls are open, so T = B(x,e) is open, and x \<in> T \<subseteq> S.
                        *)
    hence "x \<in> interior S"
                        using interior_def by auto
    } note RightToLeft = this
  thus ?thesis
                        using LeftToRight by auto
  qed
lemma mem_interior_0:
    "x \<in> interior S \<longleftrightarrow> (\<exists>e>0. ball x e \<subseteq> S)"
                        using open_contains_ball_eq [where S="interior S"]
                        by (simp add: open_subset_interior)
lemma st:
    "(S::'a set) = T"
  proof-
    {fix x assume "x:S" hence "x:T" sorry} note imp1 = this
```

```
    {fix x assume "x:T" hence "x:S" sorry}
  from this show ?thesis
                      using imp1 by auto
  qed
lemma interior_def2:
    "interior S = {x. \<exists>e>0. (ball x e) \<subseteq> S}"
                      using interior_ball[of _ S] by auto
                      (* [of _ S] Specifies fixed S, arbitrary x *)
lemma interior_def_LHS:
    "interior S \<subseteq> {x. \<exists>e>0. (ball x e) \<subseteq> S}"
  proof-
  {
  fix y assume "y \<in> interior S"
    hence "\<exists>e>0. (ball y e) \<subseteq> S"
                      using interior_ball by auto
    hence "y \<in> {x. \<exists>e>0. ball x e \<subseteq> S}"
                      by auto
    } note LHS = this
  thus ?thesis
                      using LHS by auto
  qed
lemma interior_def_RHS:
    "{x. \<exists>e>0. (ball x e) \<subseteq> S} \<subseteq> interior S"
  proof-
  {
  fix y assume "y \<in> {x. \<exists>e>0. (ball x e) \<subseteq> S}"
    hence "\<exists>e>0. ball y e \<subseteq> S"
                      by auto
    hence "y \<in> interior S"
                      using interior_ball by auto
    } note RHS = this
  thus ?thesis
                      using RHS by auto
  qed
end
```