Assignment 05

By Martin Lundfall, Denis Erfurt (k0025944)

May 24, 2016

Contents

theory *LundfallErfurtEx05*
imports *Main*
begin

Denis Erfurt - k0025944

Exercise 1

```
theorem ex1:
fixes f :: bool ⇒ bool
shows f (f (f n)) = f n
proof −
  have ¬ n ∨ n by (rule excluded-middle)
  then show ?thesis
  proof
    assume 1: ¬ n
    have ¬ f n ∨ f n by (rule excluded-middle)
    then show ?thesis
    proof
      assume 2: ¬ f n
      then have ¬ f(f(n)) using 1 by simp
      then have 4: ¬ f(f(f(n))) using 2 by simp
      from 2 4 show ?thesis by simp
    next
      assume 5: f n
      have ¬ f(f(n)) ∨ f(f(n)) by (rule excluded-middle)
      then show ?thesis
      proof
        assume ¬ f(f(n))
        then have f(f(f(n))) using 1 5 by simp
        thus ?thesis using 5 by simp
      next
        assume f(f(n))
        then have f(f(f(n))) using 5 by simp
        thus ?thesis using 5 by simp
      qed
```

```
    qed
  next
    assume 6: n
    have ¬ f n ∨ f n by (rule excluded-middle)
    then show ?thesis
    proof
      assume 7: ¬ f n
      have ¬ f(f(n))∨ f(f(n)) by (rule excluded-middle)
      thus ?thesis
      proof
        assume ¬ f(f(n))
        then have ¬ f(f(f(n))) using 7 by simp
        thus ?thesis using 7 by simp
      next
        assume f(f(n))
        then have ¬ f(f(f(n))) using 6 7 by simp
        thus ?thesis using 7 by simp
      qed
    next
      assume 8: f n
      then have f(f(n)) using 6 by simp
      then have f(f(f(n))) using 6 8 by simp
      thus ?thesis using 8 by simp
    qed
  qed
qed
```

Exercise 2

```
abbreviation leibnizEq :: 'a ⇒ 'a ⇒ bool (infixl =L 42 ) where
a =L b ≡ ∀ P. P a ⟶ P b

lemma refl:
assumes a =L b
shows ∀ P. P a ⟷ P b
proof −
  {
    fix P
    {
      assume 1: P b
      {
        assume 2: ¬ (P a)
        from assms have 3: ¬ (P a) ⟶ ¬ (P b) by (rule allE)
        from 3 2 have 4: ¬ P b by (rule mp)
        from 4 1 have False by (rule notE)
      }
      from this have P a by (rule ccontr)
    } note RtoL = this
    {
      assume 5: P a
```

```
      from assms have 6: P a ⟶ P b by (rule allE)
      from 6 5 have P b by (rule mp)
    } note LtoR = this
    from LtoR RtoL have P a ⟷ P b by (rule iffI)
  }
  from this have ∀ P . P a ⟷ P b by (rule allI)
thus ?thesis .
qed

lemma N2:
assumes a =L b
shows a = b
proof –
    from assms have 2: a = a ⟶ a = b by (rule allE)
    have 3: a = a by simp
    from 2 3 have 4: a = b by (rule mp)
thus ?thesis .
qed

lemma N3:
assumes a = b
shows a =L b
proof –
  {
    fix P
    {
      assume 1: P a
      from assms 1 have P b by simp
    }
    from this have P a ⟶ P b by (rule impI)
  } from this have ∀ P. P a ⟶ P b by (rule allI)
thus ?thesis .
qed
```

Exercise 3

```
typedecl bird

consts call :: bird ⇒ bird ⇒ bird (infix · 51)
definition mockingbird where mockingbird M ≡ ∀ x. M · x = x · x
definition composesWith where composesWith C A B ≡ ∀ x. A · (B · x) = C ·
x
definition isfond where isfond A B ≡ A · B = B

axiomatization where
  C1: ∃ C. composesWith C A B and
  C2: ∃ M. mockingbird M

theorem first–rumor: ∀ x. ∃ y. isfond x y
proof –
```

```
  {
    fix x
    from C2 obtain M where 1: mockingbird M by (rule exE)
    from 1 have 2: ∀ z . M · z = z · z unfolding mockingbird-def .
    from C1 obtain C where 4: composesWith C x M by (rule exE)
    then have 5: ∀ y. x · (M · y) = C · y unfolding composesWith-def .
    have C · C = C · C by simp
    from 2 have M · C = C · C by (rule allE)
    from 5 have 6: x · (M · C) = C · C by (rule allE)
    from 2 5 have x · (M · C) = M · C by simp
    then have isfond x (M · C) unfolding isfond-def .
    then have ∃ y. isfond x y by (rule exI)
  }
  from this have ∀ x. ∃ y. isfond x y by (rule allI)
  thus ?thesis .
qed

end
```