```
theory primes-mod-four
imports Main
        HOL−Computational-Algebra.Primes
begin

lemma aux1:
  fixes m k :: nat
  assumes (m ∗ k) mod 4 = 3
  shows m mod 4 = 3 ∨ k mod 4 = 3
  by (smt One-nat-def add-Suc-right assms mod-double-modulus mod-mod-trivial
      mod-mult-right-eq mult.right-neutral mult-0-right mult-2-right
      not-mod2-eq-Suc-0-eq-0 numeral-2-eq-2 numeral-3-eq-3
      numeral-Bit0 one-add-one zero-le zero-less-Suc zero-neq-numeral)

— an alternative proof
lemma aux1':
  fixes m k :: nat
  assumes (m ∗ k) mod 4 = 3
  shows m mod 4 = 3 ∨ k mod 4 = 3
proof (rule ccontr)
  assume ¬ (m mod 4 = 3 ∨ k mod 4 = 3)
  moreover have m mod 4 = 0 ∨ m mod 4 = 1 ∨ m mod 4 = 2 ∨ m mod 4 = 3
    by linarith
  moreover have k mod 4 = 0 ∨ k mod 4 = 1 ∨ k mod 4 = 2 ∨ k mod 4 = 3
    by linarith
  moreover have (m ∗ k) mod 4 = ((m mod 4) ∗ (k mod 4)) mod 4
    by (simp add: mod-mult-eq)
  ultimately show False
    using ‹(m ∗ k) mod 4 = 3› by auto
qed

lemma aux2:
  fixes n :: nat
  shows n mod 4 = 3 ⟶ (∃ p. prime p ∧ p dvd n ∧ p mod 4 = 3)
proof (induct n rule: less-induct)
  case (less n)
  then have IH: ⋀ m. m < n ⟹ m mod 4 = 3 ⟶
                (∃ p. prime p ∧ p dvd m ∧ p mod 4 = 3) by simp
  show n mod 4 = 3 ⟶ (∃ p. prime p ∧ p dvd n ∧ p mod 4 = 3)
  proof (clarify)
    assume h: n mod 4 = 3
    show (∃ p. prime p ∧ p dvd n ∧ p mod 4 = 3)
    proof cases
      assume prime n
      with h show ?thesis by auto
    next
      assume ¬ prime n
      moreover from h have n ≥ 2 by linarith
      ultimately obtain m k where m < n and k < n and n = m ∗ k
```

```
      by (metis Suc-1 dvd-def dvd-imp-le le-neq-implies-less
            less-le-trans mult.commute mult.right-neutral
            nat-mult-eq-cancel-disj prime-nat-naiveI zero-less-Suc)
    have m mod 4 = 3 ∨ k mod 4 = 3
      using ‹n = m * k› aux1 h by blast
    show ∃ p. prime p ∧ p dvd n ∧ p mod 4 = 3
      using IH ‹k < n› ‹m < n› ‹m mod 4 = 3 ∨ k mod 4 = 3› ‹n = m * k›
            prime-dvd-mult-eq-nat by blast
   qed
  qed
qed

theorem infinite-primes-three-mod-four: infinite {p :: nat. prime p ∧ p mod 4 =
3}
proof
  let ?S = {p :: nat. prime p ∧ p mod 4 = 3}
  assume fS: finite ?S
  let ?u = 4 * (∏ x ∈ ?S. x) − 1
  have h1: (∏ x ∈ ?S. x) ≥ 1
    by (metis (no-types, lifting) mem-Collect-eq prime-ge-1-nat prod-ge-1)
  hence h2: (∏ x ∈ ?S. x) = (∏ x ∈ ?S. x) − 1 + 1
    by linarith
  have ?u mod 4 = 3
    by (subst h2) (simp add: ring-distribs)
  then obtain p where prime p and p dvd ?u and p mod 4 = 3
    using aux2 by blast
  have p ∉ ?S
  proof
    assume p ∈ ?S
    hence p dvd 4 * (∏ x ∈ ?S. x)
      by (simp add: dvd-prod-eqI fS)
    with ‹p dvd ?u› have p dvd 1
      by (metis (no-types, lifting) dvd-diffD1 h1 less-one
          mult-eq-0-iff not-le zero-neq-numeral)
    thus False
      using ‹prime p› not-prime-unit by blast
  qed
  moreover with ‹prime p› ‹p mod 4 = 3› have p ∈ ?S by auto
  ultimately show False by simp
qed

end
```