```
theory Problem-5
  imports HOL−Analysis.Analysis
begin
```

0.1 Problem 5

Real numbers $M, a_1, a_2, \ldots, a_{10}$ are given. Prove that, if $a_1x_1 + a_2x_2 + \cdots + a_{10}x_{10} \leq M$ for all x_i such that $|x_i| = 1$, then

$$\sqrt{a_1^2 + a_2^2 + \cdots + a_{10}^2} \leq M.$$

```
lemma sqr-sum-ineq:
  list-all (λx. x ≥ 0) xs ⟹ sum-list (map power2 xs) ≤ (sum-list xs)^2
  for xs :: real list
proof (induction xs)
  case Nil
  then show ?case by auto
next
  case (Cons x xs)
  note IH = ‹list-all (λx. x ≥ 0) xs ⟹ sum-list (map power2 xs) ≤ (sum-list
xs)^2›
  note nonneg = ‹list-all (λx. x ≥ 0) (x # xs)›
  then have x ≥ 0 and nonneg': list-all (λx. x ≥ 0) xs by auto
  hence sum-list xs ≥ 0 using sum-list-nonneg unfolding list-all-def by auto

  have sum-list (map power2 (x # xs)) = x^2 + sum-list (map power2 xs) by auto
  also have ... ≤ x^2 + (sum-list xs)^2 using IH and nonneg' by auto
  also have ... ≤ x^2 + 2*x*(sum-list xs) + (sum-list xs)^2
    using ‹x ≥ 0› and ‹sum-list xs ≥ 0› by auto
  also have ... = (x + sum-list xs)^2 by algebra
  also have ... = (sum-list (x # xs))^2 by auto
  finally show sum-list (map power2 (x # xs)) ≤ (sum-list (x # xs))^2.
qed

definition sgn' :: real ⇒ real where
sgn' x = (if x ≥ 0 then 1 else −1)

lemma [simp]: x * sgn' x = |x|
  unfolding sgn'-def by auto

lemma [simp]: |sgn' x| = 1
  unfolding sgn'-def by auto

theorem problem5:
  fixes M :: real and as :: real list
  assumes *: ⋀xs. list-all (λx. |x| = 1) xs ⟹ sum-list (map2 (*) as xs) ≤ M
  shows sqrt (sum-list (map power2 as)) ≤ M
proof −
```

```
  define xs where xs = map sgn' as
  then have list-all (λx. |x| = 1) xs unfolding list-all-def by auto
  with * [of xs] have sum-abs-below-M: sum-list (map abs as) ≤ M
   unfolding xs-def by (auto simp add: map2-map-map [where f=id, simplified])
  moreover have sum-abs-nonneg: sum-list (map abs as) ≥ 0
    using sum-list-abs abs-ge-zero order-trans by blast
  ultimately have M ≥ 0 by auto

  have [simp]: power2 ∘ abs = (power2 :: 'a ⇒ ('a :: linordered-idom))
    by auto
  have list-all (λx. x ≥ 0) (map abs as) unfolding list-all-def by auto
  from sqr-sum-ineq [OF this]
  have sum-list (map power2 as) ≤ (sum-list (map abs as))²
    by auto
  also have ... ≤ M² using sum-abs-below-M sum-abs-nonneg by auto
  finally have sum-list (map power2 as) ≤ M².
  with ‹M ≥ 0› show sqrt (sum-list (map power2 as)) ≤ M
    by (metis abs-of-nonneg real-sqrt-abs real-sqrt-le-iff)
qed

end
```