Descartes' Rule of Signs

Manuel Eberl

March 17, 2025

Abstract

In this work, we formally proved Descartes Rule of Signs, which relates the number of positive real roots of a polynomial with the number of sign changes in its coefficient list.

Our proof follows the simple inductive proof given by Arthan [1], which was also used by John Harrison in his HOL Light formalisation. We proved most of the lemmas for arbitrary linearly-ordered integrity domains (e.g. integers, rationals, reals); the main result, however, requires the intermediate value theorem and was therefore only proven for real polynomials.

Contents

1 Sign changes and Descartes' Rule of Signs

theory *Descartes-Sign-Rule*
imports
  Complex-Main
  HOL−Computational-Algebra.Polynomial
begin

lemma *op-plus-0*: $((+)\ (0 :: 'a :: monoid\text{-}add)) = id$
  ⟨*proof*⟩

lemma *filter-dropWhile*:
  $filter\ (\lambda x.\ \neg P\ x)\ (dropWhile\ P\ xs) = filter\ (\lambda x.\ \neg P\ x)\ xs$
  ⟨*proof*⟩

1.1 Polynomials

A real polynomial whose leading and constant coefficients have opposite non-zero signs must have a positive root.

lemma *pos-root-exI*:
  assumes $poly\ p\ 0 * lead\text{-}coeff\ p < (0 :: real)$
  obtains x **where** $x > 0\ poly\ p\ x = 0$
⟨*proof*⟩

Substitute X with aX in a polynomial $p(X)$. This turns all the $X - a$ factors in p into factors of the form $X - 1$.

definition *reduce-root* **where**
  $reduce\text{-}root\ a\ p = pcompose\ p\ [:0,\ a:]$

lemma *reduce-root-pCons*:
  $reduce\text{-}root\ a\ (pCons\ c\ p) = pCons\ c\ (smult\ a\ (reduce\text{-}root\ a\ p))$
  ⟨*proof*⟩

lemma *reduce-root-nonzero* [*simp*]:
  $a \neq 0 \Longrightarrow p \neq 0 \Longrightarrow reduce\text{-}root\ a\ p \neq (0 :: 'a :: idom\ poly)$
  ⟨*proof*⟩

1.2 List of partial sums

We first define, for a given list, the list of accumulated partial sums from left to right: the list *psums xs* has as its i-th entry $\sum_{j=0}^{i} \mathrm{xs}_i$.

fun *psums* **where**
  $psums\ [] = []$
| $psums\ [x] = [x]$
| $psums\ (x\#y\#xs) = x\ \#\ psums\ ((x+y)\ \#\ xs)$

lemma *length-psums* [*simp*]: $length\ (psums\ xs) = length\ xs$
  ⟨*proof*⟩

lemma *psums-Cons*:
  $psums\ (x\#xs) = (x :: 'a :: semigroup\text{-}add)\ \#\ map\ ((+)\ x)\ (psums\ xs)$
  ⟨*proof*⟩

lemma *last-psums*:
  $(xs :: 'a :: monoid\text{-}add\ list) \neq [] \Longrightarrow last\ (psums\ xs) = sum\text{-}list\ xs$
  ⟨*proof*⟩

lemma *psums-0-Cons* [*simp*]:
  $psums\ (0\#xs :: 'a :: monoid\text{-}add\ list) = 0\ \#\ psums\ xs$
  ⟨*proof*⟩

lemma *map-uminus-psums*:
  fixes $xs :: 'a :: ab\text{-}group\text{-}add\ list$

shows *map uminus* (*psums xs*) = *psums* (*map uminus xs*)
⟨*proof*⟩

lemma *psums-replicate-0-append*:
psums (*replicate n* (*0* :: *'a* :: *monoid-add*) @ *xs*) =
replicate n 0 @ *psums xs*
⟨*proof*⟩

lemma *psums-nth*: $n < \mathit{length}\ xs \Longrightarrow \mathit{psums}\ xs\ !\ n = (\sum i{\le}n.\ xs\ !\ i)$
⟨*proof*⟩

1.3 Sign changes in a list

Next, we define the number of sign changes in a sequence. Intuitively, this is the number of times that, when passing through the list, a sign change between one element and the next element occurs (while ignoring all zero entries).

We implement this by filtering all zeros from the list of signs, removing all adjacent equal elements and taking the length of the resulting list minus one.

definition *sign-changes* :: (*'a* :: {*sgn*,*zero*} *list*) ⇒ *nat* **where**
sign-changes xs = *length* (*remdups-adj* (*filter* (λ*x*. *x* ≠ *0*) (*map sgn xs*))) − *1*

lemma *sign-changes-Nil* [*simp*]: *sign-changes* [] = *0*
⟨*proof*⟩

lemma *sign-changes-singleton* [*simp*]: *sign-changes* [*x*] = *0*
⟨*proof*⟩

lemma *sign-changes-cong*:
assumes *map sgn xs* = *map sgn ys*
shows *sign-changes xs* = *sign-changes ys*
⟨*proof*⟩

lemma *sign-changes-Cons-ge*: *sign-changes* (*x* # *xs*) ≥ *sign-changes xs*
⟨*proof*⟩

lemma *sign-changes-Cons-Cons-different*:
fixes *x y* :: *'a* :: *linordered-idom*
assumes *x* ∗ *y* < *0*
shows *sign-changes* (*x* # *y* # *xs*) = *1* + *sign-changes* (*y* # *xs*)
⟨*proof*⟩

lemma *sign-changes-Cons-Cons-same*:
fixes *x y* :: *'a* :: *linordered-idom*
shows *x* ∗ *y* > *0* ⟹ *sign-changes* (*x* # *y* # *xs*) = *sign-changes* (*y* # *xs*)
⟨*proof*⟩

lemma *sign-changes-0-Cons* [*simp*]:
  sign-changes (*0* # *xs* :: *'a* :: *idom-abs-sgn list*) = *sign-changes xs*
  ⟨*proof*⟩

lemma *sign-changes-two*:
  fixes $x\ y$ :: *'a* :: *linordered-idom*
  shows *sign-changes* [*x*,*y*] =
      (*if* $x > 0 \wedge y < 0 \vee x < 0 \wedge y > 0$ *then 1 else 0*)
  ⟨*proof*⟩

lemma *sign-changes-induct* [*case-names nil sing zero nonzero*]:
  assumes $P\ []\ \bigwedge x.\ P\ [x]\ \bigwedge xs.\ P\ xs \Longrightarrow P\ (0\#xs)$
    $\bigwedge x\ y\ xs.\ x \neq 0 \Longrightarrow P\ ((x + y)\ \#\ xs) \Longrightarrow P\ (x\ \#\ y\ \#\ xs)$
  shows *P xs*
⟨*proof*⟩

lemma *sign-changes-filter*:
  fixes *xs* :: *'a* :: *linordered-idom list*
  shows *sign-changes* (*filter* ($\lambda x.\ x \neq 0$) *xs*) = *sign-changes xs*
  ⟨*proof*⟩

lemma *sign-changes-Cons-Cons-0*:
  fixes *xs* :: *'a* :: *linordered-idom list*
  shows *sign-changes* (*x* # *0* # *xs*) = *sign-changes* (*x* # *xs*)
  ⟨*proof*⟩

lemma *sign-changes-uminus*:
  fixes *xs* :: *'a* :: *linordered-idom list*
  shows *sign-changes* (*map uminus xs*) = *sign-changes xs*
⟨*proof*⟩

lemma *sign-changes-replicate*: *sign-changes* (*replicate n x*) = *0*
  ⟨*proof*⟩

lemma *sign-changes-decompose*:
  assumes $x \neq$ (*0* :: *'a* :: *linordered-idom*)
  shows *sign-changes* (*xs* @ *x* # *ys*) =
    sign-changes (*xs* @ [*x*]) + *sign-changes* (*x* # *ys*)
⟨*proof*⟩

If the first and the last entry of a list are non-zero, its number of sign changes is even if and only if the first and the last element have the same sign. This will be important later to establish the base case of Descartes' Rule. (if there are no positive roots, the number of sign changes is even)

lemma *even-sign-changes-iff*:
  assumes *xs* ≠ ([] :: *'a* :: *linordered-idom list*) *hd xs* ≠ *0 last xs* ≠ *0*
  shows *even* (*sign-changes xs*) ⟷ *sgn* (*hd xs*) = *sgn* (*last xs*)
⟨*proof*⟩

1.4 Arthan's lemma

context
begin

We first prove an auxiliary lemma that allows us to assume w.l.o.g. that the first element of the list is non-negative, similarly to what Arthan does in his proof.

private lemma *arthan-wlog* [*consumes 3*, *case-names nonneg lift*]:
fixes $xs :: 'a :: linordered\text{-}idom\ list$
assumes $xs \neq []\ \ last\ xs \neq 0\ \ x + y + sum\text{-}list\ xs = 0$
assumes $\bigwedge x\ y\ xs.\ xs \neq [] \Longrightarrow last\ xs \neq 0 \Longrightarrow$
$x + y + sum\text{-}list\ xs = 0 \Longrightarrow x \geq 0 \Longrightarrow P\ x\ y\ xs$
assumes $\bigwedge x\ y\ xs.\ xs \neq [] \Longrightarrow P\ x\ y\ xs \Longrightarrow P\ (-x)\ (-y)\ (map\ uminus\ xs)$
shows $P\ x\ y\ xs$
⟨*proof*⟩

We now show that the α and β in Arthan's proof have the necessary properties: their difference is non-negative and even.

private lemma *arthan-aux1*:
fixes $xs :: 'a :: \{linordered\text{-}idom\}\ list$
assumes $xs \neq []\ \ last\ xs \neq 0\ \ x + y + sum\text{-}list\ xs = 0$
defines $v \equiv \lambda xs.\ int\ (sign\text{-}changes\ xs)$
shows $v\ (x\ \#\ y\ \#\ xs) - v\ ((x + y)\ \#\ xs) \geq$
$v\ (psums\ (x\ \#\ y\ \#\ xs)) - v\ (psums\ ((x + y)\ \#\ xs)) \wedge$
$even\ (v\ (x\ \#\ y\ \#\ xs) - v\ ((x + y)\ \#\ xs) -$
$(v\ (psums\ (x\ \#\ y\ \#\ xs)) - v\ (psums\ ((x + y)\ \#\ xs))))$
⟨*proof*⟩

Now we can prove the main lemma of the proof by induction over the list with our specialised induction rule for *sign-changes*. It states that for a non-empty list whose last element is non-zero and whose sum is zero, the difference of the sign changes in the list and in the list of its partial sums is odd and positive.

lemma *arthan*:
fixes $xs :: 'a :: linordered\text{-}idom\ list$
assumes $xs \neq []\ \ last\ xs \neq 0\ \ sum\text{-}list\ xs = 0$
shows $sign\text{-}changes\ xs > sign\text{-}changes\ (psums\ xs) \wedge$
$odd\ (sign\text{-}changes\ xs - sign\text{-}changes\ (psums\ xs))$
⟨*proof*⟩

end

1.5 Roots of a polynomial with a certain property

The set of roots of a polynomial p that fulfil a given property P:

definition $roots\text{-}with\ P\ p = \{x.\ P\ x \wedge poly\ p\ x = 0\}$

The number of roots of a polynomial p with a given property P, where multiple roots are counted multiple times.

definition *count-roots-with* $P\ p = (\sum x \in$ *roots-with* $P\ p.$ *order* $x\ p)$

abbreviation *pos-roots* $\equiv$ *roots-with* $(\lambda x.\ x > 0)$
abbreviation *count-pos-roots* $\equiv$ *count-roots-with* $(\lambda x.\ x > 0)$

lemma *finite-roots-with* [*simp*]:
  $(p :: 'a ::$ *linordered-idom poly*$) \neq 0 \Longrightarrow$ *finite* (*roots-with* $P\ p$)
  ⟨*proof*⟩

lemma *count-roots-with-times-root*:
  assumes $p \neq 0$ P $(a :: 'a ::$ *linordered-idom*$)$
  shows *count-roots-with* P $([:a, -1:] * p) =$ *Suc* (*count-roots-with* $P\ p$)
⟨*proof*⟩

1.6 Coefficient sign changes of a polynomial

abbreviation (*input*) *coeff-sign-changes* $f \equiv$ *sign-changes* (*coeffs* f)

We first show that when building a polynomial from a coefficient list, the coefficient sign sign changes of the resulting polynomial are the same as the same sign changes in the list.

Note that constructing a polynomial from a list removes all trailing zeros.

lemma *sign-changes-coeff-sign-changes*:
  assumes *Poly* $xs = (p :: 'a ::$ *linordered-idom poly*$)$
  shows *sign-changes* $xs =$ *coeff-sign-changes* p
⟨*proof*⟩

By applying *reduce-root* a, we can assume w.l.o.g. that the root in question is 1, since applying root reduction does not change the number of sign changes.

lemma *coeff-sign-changes-reduce-root*:
  assumes $a > (0 :: 'a ::$ *linordered-idom*$)$
  shows *coeff-sign-changes* (*reduce-root* $a\ p$) $=$ *coeff-sign-changes* p
⟨*proof*⟩

Multiplying a polynomial with a positive constant also does not change the number of sign changes. (in fact, any non-zero constant would also work, but the proof is slightly more difficult and positive constants suffice in our use case)

lemma *coeff-sign-changes-smult*:
  assumes $a > (0 :: 'a ::$ *linordered-idom*$)$
  shows *coeff-sign-changes* (*smult* $a\ p$) $=$ *coeff-sign-changes* p
  ⟨*proof*⟩

context
begin

We now show that a polynomial with an odd number of sign changes contains a positive root. We first assume that the constant coefficient is non-zero. Then it is clear that the polynomial's sign at 0 will be the sign of the constant coefficient, whereas the polynomial's sign for sufficiently large inputs will be the sign of the leading coefficient.

Moreover, we have shown before that in a list with an odd number of sign changes and non-zero initial and last coefficients, the initial coefficient and the last coefficient have opposite and non-zero signs. Then, the polynomial obviously has a positive root.

private lemma *odd-coeff-sign-changes-imp-pos-roots-aux*:
  assumes [*simp*]: $p \neq (0 :: real\ poly)$ $poly\ p\ 0 \neq 0$
  assumes *odd* (*coeff-sign-changes p*)
  obtains x **where** $x > 0$ $poly\ p\ x = 0$
⟨*proof*⟩

We can now show the statement without the restriction to a non-zero constant coefficient. We can do this by simply factoring p into the form $p \cdot x^n$, where n is chosen as large as possible. This corresponds to stripping all initial zeros of the coefficient list, which obviously changes neither the existence of positive roots nor the number of coefficient sign changes.

lemma *odd-coeff-sign-changes-imp-pos-roots*:
  assumes $p \neq (0 :: real\ poly)$
  assumes *odd* (*coeff-sign-changes p*)
  obtains x **where** $x > 0$ $poly\ p\ x = 0$
⟨*proof*⟩

end

1.7 Proof of Descartes' sign rule

For a polynomial $p(X) = a_0 + \ldots + a_n X^n$, we have $[X^i](1-X)p(X) = (\sum_{j=0}^{i} a_j)$.

lemma *coeff-poly-times-one-minus-x*:
  fixes $g :: 'a :: linordered\text{-}idom\ poly$
  shows $coeff\ g\ n = (\sum i \leq n.\ coeff\ (g * [:1, -1:])\ i)$
  ⟨*proof*⟩

We apply the previous lemma to the coefficient list of a polynomial and show: given a polynomial $p(X)$ and $q(X) = (1-X)p(X)$, the coefficient list of $p(X)$ is the list of partial sums of the coefficient list of $q(X)$.

lemma *Poly-times-one-minus-x-eq-psums*:
  fixes $xs :: 'a :: linordered\text{-}idom\ list$

assumes [*simp*]: $length\ xs = length\ ys$
assumes $Poly\ xs = Poly\ ys * [:1,\ -1:]$
shows $ys = psums\ xs$
⟨*proof*⟩

We can now apply our main lemma on the sign changes in lists to the coefficient lists of a nonzero polynomial $p(X)$ and $(1-X)p(X)$: the difference of the changes in the coefficient lists is odd and positive.

lemma *sign-changes-poly-times-one-minus-x*:
fixes g :: $'a$ :: *linordered-idom poly* **and** a :: $'a$
assumes *nz*: $g \neq 0$
defines $v \equiv$ *coeff-sign-changes*
shows $v\ ([:1,\ -1:] * g) - v\ g > 0 \wedge odd\ (v\ ([:1,\ -1:] * g) - v\ g)$
⟨*proof*⟩

We can now lift the previous lemma to the case of $p(X)$ and $(a - X)p(X)$ by substituting X with aX, yielding the polynomials $p(aX)$ and $a \cdot (1-X) \cdot p(aX)$.

lemma *sign-changes-poly-times-root-minus-x*:
fixes g :: $'a$ :: *linordered-idom poly* **and** a :: $'a$
assumes *nz*: $g \neq 0$ **and** *pos*: $a > 0$
defines $v \equiv$ *coeff-sign-changes*
shows $v\ ([:a,\ -1:] * g) - v\ g > 0 \wedge odd\ (v\ ([:a,\ -1:] * g) - v\ g)$
⟨*proof*⟩

Finally, the difference of the number of coefficient sign changes and the number of positive roots is non-negative and even. This follows straightforwardly by induction over the roots.

lemma *descartes-sign-rule-aux*:
fixes p :: *real poly*
assumes $p \neq 0$
shows *coeff-sign-changes* $p \geq$ *count-pos-roots* $p \wedge$
even (*coeff-sign-changes* p − *count-pos-roots* p)
⟨*proof*⟩

The main theorem is then an obvious consequence

theorem *descartes-sign-rule*:
fixes p :: *real poly*
assumes $p \neq 0$
shows $\exists\, d.$ *even* $d \wedge$ *coeff-sign-changes* p = *count-pos-roots* $p + d$
⟨*proof*⟩

end

References

[1] R. D. Arthan. Descartes' rule of signs by an easy induction. 2007.