Formal Verification of the Interaction Between Semi-Algebraic Sets and Real Analytic Functions

Anonymous Author(s)

Abstract

Semi-algebraic sets and real analytic functions are fundamental concepts in Real Algebraic Geometry and Real Analysis, respectively. These concepts interact in the study of Differential Equations, where the real analytic solution to a differential equation is known to enter or exit a semi-algebraic set in a predicable way. Motivated to enhance the capability to reason about differential equations in the Prototype Verification System (PVS), a formalization of multivariate polynomials, semi-algebraic sets, and real analytic functions is developed. The favorable way that a real analytic function enters and exits a semi-algebraic set is proven. It is further shown that if the function is assumed to be smooth, a slightly weaker assumption than real analytic, these favorable interactions with semi-algebraic sets may fail.

Keywords: formal verification, PVS, semi-algebraic sets

1 Introduction

Differential equations are a powerful tool for modeling the evolution of continuous states in dynamical systems [15, 34]. While a variable is modeled as a solution to a differential equation, semi-algebraic (SA) sets can be used to define environment constraints and control properties of the variable. In the context of safety-critical applications, the interaction between a solution to a differential equation and a semi-algebraic set is crucial for verifying the safety properties of the given system. In particular, the way that a solution to a differential equation leaves and enters an SA set can inform when a safety violation has occurred or ended. One specific example is two aircraft maintaining a safe distance from one another [11].

Differential dynamic logic (DDL) is a logic that allows formal reasoning about hybrid systems, using properties of solutions of differential equations, in some cases without having needed the explicit solution[26, 27]. Under modest assumptions, the solution of a differential equation is guaranteed to be a real analytic function (see, e.g. [9], Chapter 1.D, or [35], Chapter 9.37), and so reasoning about how a solution to a differential equation interacts with its domain can often be reduced to reasoning about the behavior of real analytic functions and SA sets [18, 32].

This work focuses on the formal specification and verification of the interactions between SA sets and real analytic functions in the Prototype Verification System (PVS) [24, 25]. The main motivation is the eventual implementation of a formally verified version of DDL in PVS that allows users to reason about cyber-physical systems using DDL interactively in PVS. To do this, the deduction rules for DDL must be formally verified in PVS, and as noted above, these involve reasoning about real analytic functions and SA sets. SA sets are defined using collections of multivariate polynomial constraints, allowing a wide variety of sets to be defined. The formalization provided in this paper allows for reasoning about general SA sets and particular user-specified instantiations of these sets.

A formal specification of the theory of real analytic functions is also developed. Real analytic functions can be written in terms of their power series, which includes functions like polynomials, trigonometric functions, exponential and logarithmic functions, and products, sums, and compositions of such functions. Importantly, even functions that are not explicitly specified may be known to be real analytic, the motivating example being solutions to many differential equations.

The way that a real analytic function interacts with the boundary of an SA set is known to have certain favorable geometric properties, which are important for reasoning in DDL. An example of this behavior is shown in Figure 1, where the analytic solution of a differential equation leaves an SA set for a complete interval of time before entering again. These interactions are formally verified in PVS. It is also shown that relaxing the assumption of real analytic to smooth (i.e., infinitely differentiable) removes the favorable interactions that are guaranteed between real analytic functions and SA sets. This offers practical insight regarding the subtle difference between real analytic and smooth functions. Furthermore, the challenges of implementing this theory in PVS give educational insight into proofs and allows further development of the growing NASA PVS library. [1]

Certified Programs and Proofs '21, Jan 18–19, 2021,

[1] https://github.com/nasa/pvslib

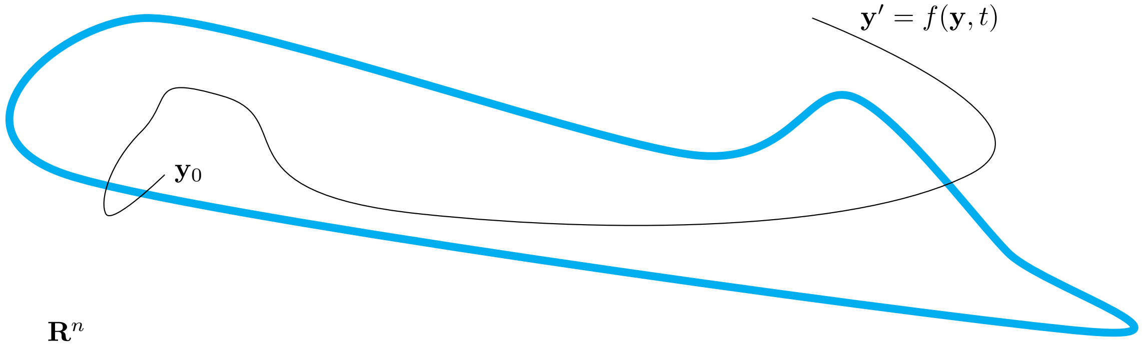

Figure 1. An analytic solution to a differential equation moves in and out of a semi-algebraic set

The remaining sections are organized as follows. Section 2 gives a brief review of multivariate polynomials and SA sets. Section 3 introduces real analytic functions and describes the way they interact with SA sets. Related work is discussed in Section 4. Conclusions and future directions are discussed in Section 5.

The mathematics presented in Sections 2 and 3 have all been specified and verified in PVS by the authors, except that in a few cases, some important concept or theorem is taken from NASA's PVS library (NASALib). This will be explicitly noted when needed.

2 Polynomials & Semi-Algebraic Sets

Real Algebraic Geometry is a branch of mathematics concerned with the study of SA sets. An SA set is a set of points that satisfy a finite sequence of multivariate polynomial equalities and inequalities, or a union of such sets [2]. As noted in the introduction, SA sets are also important to the theory of general real-valued functions, particularly how a real analytic function behaves on such a set. This section discusses a formalization of multivariate polynomials and semi-algebraic sets in PVS.

2.1 Multivariate polynomials over the reals

When mathematicians consider multivariate polynomials over the reals, it is often unclear what kind of formal objects they are referring to. Such a polynomial may be considered a member of the polynomial ring $\mathbb{R}[X_0, X_1, \ldots, X_{m-1}]$ with m indeterminants and real coefficients, or for inductive reasons a member of the ring $(\mathbb{R}[X_0, X_1, \ldots, X_{m-2}])[X_{m-1}]$) with a single indeterminant and polynomial coefficients, or as a function from $\mathbb{R}^m$ into $\mathbb{R}$, or many other possible definitions. Indeed, the fact that polynomials can be considered in different settings is part of what makes them so useful.

From the formalization standpoint, one particular representation for such polynomials has to be chosen, and translations or interpretations for any of the other definitions have to be specified and justified. The author Zippel, in [36], identifies three decision points with respect to choosing how polynomial might be represented. Expanded vs. recursive representation concerns whether coefficients are real numbers and multiple variables are allowed (expanded), or a single variable has (recursively) multivariate polynomials as coefficients. Variable sparse vs. variable dense refers to whether the representation includes variables with exponent zero (dense) or excludes them (sparse) in a monomial definition. Degree sparse vs. degree dense refers to whether all monomials up to a given multidegree are included in the representation by using a zero coefficient (dense) or if only those with non-zero coefficient are recorded (sparse). For this formalization, an expanded, (essentially) variable dense, and (essentially) degree sparse representation was chosen, as described below.

The polynomials considered here are real-linear combinations of *monomials*, expressions of the form $\mathbf{X}^{\boldsymbol{\alpha}} := X_0^{\alpha_0} \cdots X_{m-1}^{\alpha_{m-1}}$, where $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \ldots \alpha_{m-1}) \in \mathbb{N}^m$, and $m \in \mathbb{N}$. The number of entries in α is called the *dimension* of the monomial, i.e., $\dim(\mathbf{X}^{\boldsymbol{\alpha}}) = m$, while the *degree* of the of monomial is $\deg(\mathbf{X}^{\boldsymbol{\alpha}}) = \sum_{i=0}^{m-1} \alpha_i$. A *monomial* is then $c\mathbf{X}^{\boldsymbol{\alpha}}$, where $c \in \mathbb{R}$. The implementation of this is done with record datatype

```
monomial: TYPE =
  [# C := real, alpha := list[nat] #],
```

with a particular instantiation of this type taking the form

```
  m = (# C := c, alpha := L #).
```

The symbol " ' " is field accessor of a record, i.e., `m'alpha = L` . This is equivalent to the dot notation in programming languages like Java.

Note in particular that the implementation above allows for coefficients to be zero, hence not *forcing* degree sparsity.

A multivariate polynomial function has the form

$$p = \sum_{k=0}^{n} c_k \mathbf{X}^{\boldsymbol{\alpha}(k)}, \tag{1}$$

where $n \in \mathbb{N}$ is finite, $c_k \in \mathbb{R}$, and $\boldsymbol{\alpha}(k) \in \mathbb{N}^m$ for each $i \in \mathbb{N}_{\leq n}$. The implementation represents this simply as a list,

```
MultPoly: TYPE = list[monomial].
```

This intentionally allows for different expressions to be given for the same polynomial. For example, consider the (syntactically distinct) expressions

$$
\begin{aligned}
p_1 &= X_0^2 + X_0, \\
p_2 &= X_0 + X_0^2, \\
p_3 &= X_0 X_1^0 + X_0^2, \\
p_4 &= X_0 + 3X_0^2 + (-2)X_0^2, \text{ and} \\
p_5 &= X_0 + X_0^2 + 0X_0^3,
\end{aligned}
\tag{2}
$$

represented in PVS as

```
p1: MultPoly =(: (# C:=1, alpha:=(: 2 :) #),
        (# C:=1, alpha:=(: 1 :) #) :)
p2: MultPoly =(: (# C:=1, alpha:=(: 1 :) #),
  (# C:=1, alpha:=(: 2 :) #) :)
p3: MultPoly =(: (# C:=1, alpha:=(: 1, 0 :) #),
  (# C:=1, alpha:=(: 2 :) #) :)
p4: MultPoly =(: (# C:=1, alpha:=(: 1 :) #),
  (# C:=3, alpha:=(: 2 :) #),
  (# C:=-2, alpha:=(: 2 :) #) :)
p5: MultPoly =(: (# C:=1, alpha:=(: 1 :) #),
  (# C:=1, alpha:=(: 2 :) #),
  (# C:=0, alpha:=(: 3 :) #) :).
```

These expressions are different, and yet are meant to express the same polynomial. Indeed, considered as *functions*, these are the same, and simple algebraic manipulation can turn any one into the other. This general form of polynomials allows for the easy definition of ring operations on polynomials (addition is just list concatenation), but in order to unambiguously define the dimension and degree of a polynomial, a standard form is defined.

Definition 2.1. A multivariate polynomial representation given by Equation (1) is said to be in *standard form* when the following properties hold:

1. The dimension of each monomial in the expression is the same, and some term uses the last variable non-trivially. That is, there exists $m \in \mathbb{N}$ such that $\dim(\boldsymbol{\alpha}_k) = m$ for all $k \in \mathbb{N}_{\leq n}$, and there exists $n_0 \in \mathbb{N}_{\leq n}$ with $\alpha_m(n_0) > 0$.
2. For $i \neq j \in \mathbb{N}_{\leq n}$, $\boldsymbol{\alpha}_i \neq \boldsymbol{\alpha}_j$. In other words, each exponent vector $\boldsymbol{\alpha}$ can appear at most one time in (1).
3. The coefficient $c_k \neq 0$ for each $k \in \mathbb{N}_{\leq n}$ (note that the identically zero polynomial is the empty list).
4. The monomial terms in the expression (1) are ordered by some total order on the monomials in m variables.

In the PVS formalization, each of these properties is defined using a predicate on a polynomial p. In addition, functions are defined that operate on a general polynomial and give it the corresponding property. Property 1 is defined using the predicate `minlength?(p)`, and bestowed by applying `cut(p)`, which removes trailing zeroes from exponents, and `lift(p)`, which pad exponents with zeroes to be the longest exponent length occurring in the polynomial. Property 2 is defined using the predicate `simplified?(p)`, and bestowed by `simplify(p)`. Property 3 is defined using the predicate `allnonzero?(p)`, and bestowed by `allnonzero(p)`. Property 4, with respect to the *graded lexographical* (GL) ordering described below, is defined using the predicate `is_sorted?(p)`, and bestowed by `mv_sort(p)`. Using these functions, Definition 2.1 is specified as a single predicate `mv_standard_form?(p)` that holds when all 4 predicates hold, and the corresponding function `mv_standard_form(p)` gives all four properties to the polynomial p.

The particular monomial ordering chosen for sorting monomials is the *graded lexographical* ordering. The ordering sorts first by the total degree of the monomial (graded), and breaks ties comparing the degrees of individual variables in order (lexicographic). Specifically, $\mathbf{X}^{\boldsymbol{\alpha}(0)} < \mathbf{X}^{\boldsymbol{\alpha}(1)}$ exactly when :

1. $\deg\left(\mathbf{X}^{\boldsymbol{\alpha}(0)}\right) < \deg\left(\mathbf{X}^{\boldsymbol{\alpha}(1)}\right)$, or
2. $\deg\left(\mathbf{X}^{\boldsymbol{\alpha}(0)}\right) = \deg\left(\mathbf{X}^{\boldsymbol{\alpha}(1)}\right)$ and

$$\exists j \in \mathbb{N}_{\leq m-1} \left(\boldsymbol{\alpha}_j(0) < \boldsymbol{\alpha}_j(1) \wedge \forall i \in \mathbb{N}_{<j}\ \boldsymbol{\alpha}_i(0) = \boldsymbol{\alpha}_i(1)\right).$$

As an example, the four monomials in the ring $\mathbb{R}[X_0, X_1, X_2]$ below are listed in increasing GL order.

$$X_1 X_2,\ X_0^2 X_1^2,\ X_0^2 X_1^1 X_0^1,\ X_0^4.$$

Given a polynomial whose representation is in standard form, the degree and dimension are each well-defined. The dimension is the length of the longest exponent (or in fact *any* exponent due to the `lift` function), and the degree is the maximum degree (or the degree of the last monomial, due to `mv_sort`). Functions for polynomial addition, scalar and polynomial multiplication, and polynomial exponentiation are specified, which, by definition, preserve standard form.

Multivariate polynomials so far defined have the structure of a ring, and hence can be combined and manipulated, but cannot yet be used as functions from $\mathbb{R}^m \to \mathbb{R}$. To do so, an evaluation function on polynomials is defined. In fact, two forms of evaluation are defined. *Partial* evaluation takes a list of indices and a list of values, and evaluates only those variables listed in the indices, using the corresponding values. This returns a polynomial of the *same* dimension as the original, where the evaluated variables have exponent zero. *Full* evaluation takes a list of values, at least as long as the dimension of the polynomial, and replaces the variables with the corresponding values ,ignoring values past the dimension of the polynomial, returning a real number.

The main purpose of the evaluation function is for use in defining the semi-algebraic sets of Section 2.2. A secondary use of the evaluation function is in proving the uniqueness of the standard form defined above.

Theorem 2.2. *Given a function σ that returns the standard form of a polynomial as in Definition 2.1, and p_1, p_2 polynomial expressions of the form* (1),

$$\sigma(p_1) = \sigma(p_2)$$

if and only if for all $\mathbf{x} \in \mathbb{R}^m$,

$$p_1(\mathbf{x}) = p_2(\mathbf{x}).$$

2.2 Semi-algebraic sets

Given a dimension m, a semi-algebraic set is a subset $S \subseteq \mathbb{R}^n$ defined by satisfying a finite collection multivariate polynomial relations , or a finite union of such sets. This corresponds to satisfying the disjunction (or join) of the conjunction (or meet) of a collection of polynomial relations. A boolean formula in this form is said to be in *disjunctive normal form*. In some situations, it is more convenient to consider the a general form of a quantifier-free boolean formula over multivariate polynomial relations built using the boolean operators $\vee, \wedge, \neg$, and $\Rightarrow$. Noting that every such quantifier-free boolean formula can be written in disjunctive normal form [5], the restricted definition as the *join* of *meets* is chosen without loss of generality. The technical definition and formalization details are developed below.

An *atomic polynomial formula* over the variables $\mathbf{X} := X_0, \cdots, X_{m-1}$ is defined as $p \triangleright 0$ where p is a polynomial in $\mathbb{R}[\mathbf{X}]$ and $\triangleright \in \{\geq, >, \leq, <\}$. [2] The implementation of this in PVS done with record datatype

```
atomic_poly: TYPE =
  [# poly:(mv_standard_form?), ineq:INEQ #],
```

where

```
INEQ: TYPE = { ff: [real,real -> bool] |
  (ff = <= ) OR (ff = >= ) OR
  (ff = < ) OR(ff = > ) }.
```

Note, in the type `INEQ` above the expression a higher order equality is used to compare functions, where the inequalities `<=`, `>=`, `<`, and `>` are functions that return the truth value of the inequality based on the two real operands.

The formulas to be considered are expressed as

$$\varphi = \bigvee_{i=1}^{I} \bigwedge_{j=1}^{J_i} p_{ij} \triangleright 0, \text{ where } \triangleright \in \{\geq, >, \leq, <\}, \tag{3}$$

and a subset S of $\mathbb{R}^m$ is a *semi-algebraic set*, if there is a quantifier free polynomial formula φ such that

$$S = \{\mathbf{x} \in \mathbb{R}^m \mid \varphi(\mathbf{x}) \text{ is true}\}.$$

In the formalization, the conjunction of atomic polynomial formulas is specified simply as a list,

```
meeting TYPE =list[atomic_poly],
```

and a disjunction of these conjunctions is specified as

```
joining: TYPE =list[meeting].
```

[2] The functions $=$ and $\neq$ are excluded for simplicity of the embedding of SA sets. Note that they can be described with the relations allowed.

Of course, the atomic polynomials and lists of them have no inherent meaning, being just lists. To define an SA set, evaluation functions must be defined. The functions `atom_eval`, `meet`, and `join` are defined successively to take a point $\mathbf{x} \in \mathbb{R}^m$ and return the truth value of an atomic polynomial formula, the *meet* of such formulas, and the *join* of *meets* evaluated at the point.

A semi-algebraic set $S(\varphi)$ defined by φ is then specified in PVS by

```
semi_alg(j:joining)(n:nat | n >= meet_max(j)):
  set[VectorN(n)] =
  { x:VectorN(n) | join(j)(x) }.
```

One of the most important basic properties of semi-algebraic sets is that they are closed under finite set operations. The following theorem expresses this.

Theorem 2.3. *For two semi algebraic sets S_1 and S_2, the following properties hold:*

1. *The union $S_1 \cup S_2$ is an SA set.*
2. *The intersection $S_1 \cap S_2$ is an SA set.*
3. *The compliment $\neg S_1$ is an SA set.*

This theorem is clear intuitively (union is join, intersection is meet, and complement is negation), but due to the formalization definition, the formal proof requires translating the conjunction and disjunction of two `joining` expressions in disjunctive normal form into another expression that is in disjunctive normal form. For union this is as simple as concatenating the to two lists using the append function:

```
union_join: LEMMA
  FORALL(j1,j2:joining,
  x: list[real] | length(x) >=
  max(meet_max(j1),meet_max(j2))):
    (join(j1)(x) OR join(j2)(x)) =
    join(append(j1,j2))(x).
```

For intersection, the formula for the conjunction of two `joining` expressions in disjunctive normal form (3) is given by

```
cap_join(j1,j2:joining): RECURSIVE joining =
  IF j1 = null THEN null
  ELSIF j2 = null THEN null
  ELSE append(append_to_each(car(j1),j2),
  cap_join(cdr(j1),j2))
  ENDIF
  MEASURE length(j1).
```

Here, the `append_to_each` function takes each conjunction in the second `joining` and appends it to each of the conjunctions in the first `joining`. This is because distributing a conjunction over disjunctions has the following form

$$\left(\bigvee_{i=1}^{I} \bigwedge_{j=1}^{J_i} p_{ij} \triangleright 0\right) \wedge (q \triangleright 0) = \bigvee_{i=1}^{I} \bigwedge_{j=1}^{J_i+1} w_{ij},$$

where

$$w_{i,j} = \begin{cases} p_{i,j} & j \leq J_i \\ q & j = J_i + 1. \end{cases}$$

Using the cap_join function, it can be shown that the conjunction of two disjunctive normal form expressions can be written in disjunctive normal form.

```
intersect_join: LEMMA
  FORALL(j1,j2:joining,
  x:list[real] | length(x) >=
  max(meet_max(j1),meet_max(j2))):
  (join(j1)(x) AND join(j2)(x)) =
  join(cap_join(j1,j2))(x).
```

Noting that the complement of an SA set is given by the negation of the corresponding formula, consider the negation of (3) which can be written

$$\neg\varphi = \bigwedge_{i=1}^{I} \bigvee_{j=1}^{J_i} p_{ij} \neg\triangleright\, 0 \text{ where } \neg\triangleright \in \{\geq, >, \leq, <\}. \qquad (4)$$

Here $\neg\triangleright$ is defined according to the following table:

$\triangleright$	$\neg\triangleright$
$\geq$	$<$
$\leq$	$>$
$>$	$\leq$
$<$	$\geq$

The expression in equation (4) is transformed into disjunctive normal form by repeated use of the cap_join function:

```
not_join(j:joining): RECURSIVE joining =
  IF j=null THEN(: (: :) :)
  ELSE
  cap_join(negative_atom_meet(car(j)),
  not_join(cdr(j)))
  ENDIF
  MEASURE length(j).
```

The equivalence is expressed by

```
not_join: LEMMA
  FORALL(j:joining, x:list[real] |
  length(x) >=  meet_max(j)):
  (NOT join(j)(x)) = join(not_join(j))(x).
```

As noted above, the proofs here could have been made simpler by allowing for more general boolean expressions in the definition of semi-algebraic sets. On the other hand, this would have incurred an overhead cost in the original specification, as well as in the evaluation functions. The design choice of using only formulas in disjunctive normal form allows for a much cleaner representation, at the cost of some tedious proofs.

3 Real-analytic Functions

For an open set $D \subseteq \mathbb{R}$, A real function $f : D \to \mathbb{R}$ is said to be *real analytic* at a point $c \in D$ when there exists a real sequence $\{a_k\}_{k=0}^{\infty}$ and an $r \in \mathbb{R}_{>0}$ such that

$$f(x) = \sum_{k=0}^{\infty} a_k (x - c)^k \qquad \forall x \in (c_0 - r, c_0 + r). \qquad (5)$$

Furthermore, f is analytic on a $V \subseteq D$ if it is analytic at each $x \in V$. In PVS, the sequence $\{a_k\}$ and real number r in (5) are defined by the predicate

```
analytic_parts?(c0:real,f:[real->real])
(M:posreal, ak:sequence[real]): bool =
  FORALL(x:real| abs(x-c0) < M):
  convergent?(powerseries(ak)(x-c0)) AND
  f(x) = inf_sum(powerseq(ak,x-c0)),
```

Using this predicate, an analytic function $f : \mathbb{R} \to \mathbb{R}$ at a point c_0 is defined by

```
analytic?(c0:real)(f:[real -> real]): bool =
  EXISTS(r:posreal, ak:sequence[real]):
  analytic_parts?(c0,f)(r,ak).
```

For a function $f : D \to \mathbb{R}$, where D is open, the definition in (5) is equivalent to

```
analytic?(c0:real)(lift(D,f))
```

where lift(D, f) trivially extends the domain of f to all of $\mathbb{R}$, i.e.,

```
lift(D:(open?),f:[D -> real])(x:real): real =
  IF D(x) THEN f(x) ELSE 0 ENDIF.
```

In (5), the number r is called the the *radius of convergence* of f at x. If there is not a r such that (5) holds, the maximal radius of convergence is said to be 0, while if (5) holds for all $r \in \mathbb{R}_{\geq 0}$, the maximal radius of convergence is said to be infinity. In all other cases there is an $r_{\max} \in \mathbb{R}$ which is called the maximal radius of convergence.

From the definition in (5), it is clear that the infinite sum $\sum_{k=0}^{\infty} a_k (x - c_0)^k$, $x \in (c_0 - r, c_0 + r)$ converges. Using standard properties of convergent series, it can be shown that analyticity is closed under addition and scalar multiplication. To show that the product of two analytic functions is analytic, the following lemma is required.

Lemma 3.1 (Absolute Convergence). *Suppose $f : D \to R$ is analytic at a point $c_0 \in D$, as stated in* (5). *For each $x \in (c - r, c + r)$, the sum*

$$A = \sum_{k=0}^{\infty} \left| a_k (x - c)^k \right|$$

converges.

Lemma 3.1 shows that if a function is analytic, then the series representation of the function converges absolutely. This lemma has been previously proven in NASALib's series library, so the proof will not be presented here.

With the lemma above, enough machinery is available to show that being analytic at a point is closed under summation, scalar multiplication, and multiplication.

Theorem 3.2. *Suppose $f : D \to \mathbb{R}$ and $g : D \to \mathbb{R}$ are analytic at a point $c_0 \in D$ with radius of convergence r_f and r_g respectively. i.e.,*

$$f(x) = \sum_{k=0}^{\infty} a_k (x - c_0)^k \qquad \forall x \in (c_0 - r_f, c_0 + r_f) \tag{6}$$

$$g(x) = \sum_{k=0}^{\infty} b_k (x - c_0)^k \qquad \forall x \in (c_0 - r_g, c_0 + r_g)$$

and let $r_{min} = \min(r_f, r_g)$, then the following statements hold:

1. *$f + g$ is analytic with radius of convergence r_{min},*
2. *$c \cdot f$ is analytic with radius of convergence r_f, and*
3. *$f \cdot g$ is analytic with radius of convergence r_{min}*

$$(g * f)(x) = \sum_{k=0}^{\infty} \operatorname{conv}(k, a, b) (x - c_0)^k ,$$

where $\operatorname{conv}(k, a, b)$ is the kth convolution of the sequences a and b

$$\operatorname{conv}(k, a, b) = \sum_{i=0}^{k} a_i b_{k-1}.$$

Proof. Parts 1 and 2 follow from basic convergence properties of series. For 3, let $x \in (c_0 - r_f, c_0 + r_f)$,

$$S_n = \sum_{k=0}^{n} conv(k, a, b) (x - c_0)^k , \quad \text{and} \tag{7}$$

$$R_n = \sum_{k=n+1}^{\infty} b_k (x - c_0)^k . \tag{8}$$

By using (6) and (3), S_n can be re-written as

$$S_n = g(x) \sum_{k=0}^{n} a_k (x - c_0)^k - \sum_{k=0}^{n} a_k (x - c_0)^k R_{n-k}. \tag{9}$$

By using equation (6) the first term in this expression converges

$$\lim_{n\to\infty} g(x) \sum_{k=0}^{n} a_k (x - c_0)^k = g(x) f(x).$$

It remains to show that

$$\lim_{n\to\infty} \sum_{k=0}^{n} a_k (x - c_0)^k R_{n-k} = 0.$$

Let $\epsilon > 0$, Choose $N_0 \in \mathbb{N}$ such that for all $N \geq N_0$, $|R_N| < \epsilon/(2A)$, where $A = \sum_{k=0}^{\infty} \left| a_k (x - c)^k \right|$ is finite from Lemma 3.1. This N_0 exists since $\lim_{n\to\infty} R_n = 0$.

Choose $N_1 \in \mathbb{N}$ such that for $N \geq N_1$, $\left| a_N (x - c_0)^N \right| < \epsilon/(2N_0 R)$, where $R = \max_{i \in \mathbb{R}_{\leq N_0}} |R_i|$. This exists since $a_n (x - c_0)^n \to 0$.

Now, let $N \geq N_0 + N_1$. Using the triangle inequality

$$\begin{aligned} &\left| \sum_{k=0}^{N} a_k (x - c_0)^k R_{N-k} \right| \\ \leq &\left| \sum_{k=0}^{N-N_0} a_k (x - c_0)^k R_{N-k} \right| + \left| \sum_{k=N-N_0+1}^{N} a_k (x - c_0)^k R_{N-k} \right| . \end{aligned}$$

The first summation has the bound

$$\begin{aligned} \left| \sum_{k=0}^{N-N_0} a_k (x - c_0)^k R_{N-k} \right| &\leq \sum_{k=0}^{N-N_0} \left| a_k (x - c_0)^k \right| |R_{N-k}| \\ &\leq \frac{\epsilon}{2A} \sum_{k=0}^{N-N_0} \left| a_k (x - c_0)^k \right| \\ &\leq \frac{\epsilon}{2}. \end{aligned}$$

The second summation has the bound

$$\begin{aligned} \left| \sum_{k=N-N_0+1}^{N} a_k (x - c_0)^k R_{N-k} \right| &\leq \sum_{k=N-N_0+1}^{N} \left| a_k (x - c_0)^k \right| |R_{N-k}| \\ &\leq \sum_{k=N-N_0+1}^{N} \frac{\epsilon |R_{N-k}|}{2N_1 R} \\ &\leq \sum_{k=N-N_0+1}^{N} \frac{\epsilon}{2N_0} \\ &= \frac{\epsilon N_0}{2N_0} \leq \frac{\epsilon}{2}. \end{aligned} \tag{10}$$

Therefore

$$\left| \sum_{k=0}^{N} a_k (x - c_0)^k R_{N-k} \right| \leq \epsilon,$$

and thus

$$\lim_{n\to\infty} \sum_{k=0}^{n} a_k (x - c_0)^k R_{n-k} = 0.$$

The result is shown. □

This proof above has the same general structure as the proof in [16] (Ch. 1, page 4-5). The largest departure is the introduction of $N_2 \in \mathbb{N}$, which guarantees the bound shown in (10) for $N \geq N_0 + N_1$. In the original proof, the summation in (10) is said to converge to 0 "by holding N_0 fixed as letting N go to infinity." This combination of an ϵ based argument and a limit based argument is not easily translated into PVS, so a clearer ϵ argument was constructed.

Additionally, implementation proof of Theorem 3.2 in PVS required non-trivial manipulations of finite sums. A finite sum in PVS is defined using the `sigma` function defined in the real number library of NASALib,

```
sigma(low, high, F): RECURSIVE real =
  IF low > high THEN 0
  ELSE F(high) + sigma(low, high-1, F) ENDIF
```

```
  MEASURE (LAMBDA low, high, F:
abs(high+1-low)).
```

To get from the definition of S_n in (7) to the form in (9) required a number of intermediate lemmas, including

```
sig_a_pull_conv: LEMMA
  FORALL (c0:real, a, b :sequence[real],
  x:real, n:nat, i:below(j+1)):
  sigma(i, n, LAMBDA (k: nat):
  sigma(i, k, convlf(k, a, b)) * (x- c0) ^k)
  =
  sigma(i, n, LAMBDA(k:nat): a(k)*
  sigma(i,n, LAMBDA(m:nat): IF k<=m THEN
  b(m-k)*(x-c0)^m ELSE 0 ENDIF)) ,
```

The Lemma `sig_a_pull_conv` required inducting on the quantity `n-i` in PVS, and allowed writing S_n as

$$S_n = \sum_{k=0}^{n} a_k \sum_{m=k}^{n} b_{m-k}(x - c_0)^m,$$

one of the intermediate steps between (7) and (9). These manipulations are done almost automatically by a mathematician at a blackboard, but can be difficult when doing a formal proof. From Theorem 3.2 the following useful lemma can be shown

Lemma 3.3. *For a function $f : D \to \mathbb{R}$ that is analytic at a point c_0 with radius of convergence $r \in \mathbb{R}^+$. For $k \in \mathbb{N}$ and $c \in \mathbb{R}$ the function*

$$g(x) = cf(x)^k$$

is analytic at c_0 with radius of convergence $r \in \mathbb{R}^+$.

This section focused primarily on analytic functions whose codomain is $\mathbb{R}$. This definition can be extended to a function f, with codomain in $\mathbb{R}^n$, for $n \in \mathbb{N}$ in the following way. f is analytic at a point $\mathbf{c}_0$ means that each of its sub-functions $\{f_i\}_{i=1}^n$ are analytic at $\mathbf{c}_0$, where

$$f(x) = \begin{bmatrix} f_0(x) \\ \vdots \\ f_{n-1}(x). \end{bmatrix}. \tag{11}$$

The radius of convergence of $f(x)$ is the minimal of all the radii of convergence of the f_i functions. In PVS this definition uses the `nth` function:

```
analytic?(n:nat,c0:real)
  (f:[real -> VectorN(n)]): bool =
  FORALL(i:below(n)): analytic?(c0)(nth(f,i)).
```

3.1 Analytic vs. Smooth

The way a real analytic function interacts with SA sets is preferable to the way a smooth function might interact with an SA set. To describe the difference, first this section investigates establishes the difference between the two classes of functions. A function f is *smooth* at a a point c_0 means that $f^{(n)}(c_0)$ exists for all $n \in \mathbb{N}$. The following theorem establishes that every analytic function is smooth.

Theorem 3.4. *Suppose $f : D \to R$ is analytic at a point $c_0 \in D$ with radius of convergence r, given in* (5)*. Then f is smooth on the interval $(c_0 - r, c_0 + r)$. Furthermore:*

$$a_k = \frac{f^{(k)}(c_0)}{k!}$$

and

$$f^{(n)}(x) = \sum_{k=0}^{\infty} \prod_{i=0}^{n-1} (k + n - i) a_k x^k.$$

This theorem was already established in the series library in NASALib so it is stated without proof.

From Theorem 3.4 it can be shown that the power series representation of an analytic function is unique:

$$f(x) = \sum_{k=0}^{\infty} f^{(n)}(c)\,(x - c)^k \qquad \forall x \in (c_0 - M, c_0 + M)\,. \tag{12}$$

Although an analytic function is smooth, the converse is not necessarily true. Take

$$sm(x) = \begin{cases} e^{-1/x}\sin(1/x) & x > 0 \\ 0 & x \le 0. \end{cases} \tag{13}$$

This function is clearly smooth for $x \neq 0$. Showing that $sm(x)$ is smooth at $x = 0$, but not analytic[3] requires a few helpful lemmas.

Lemma 3.5. *For $x > 0$, $n \in \mathbb{N}$, and $sm(x)$ defined in* (13)

1. *There are sequences of polynomials $\{p_n\}$ and $\{q_n\}$ such that the nth derivative of s at x is given by*

$$sm^{(n)}(x) = \frac{e^{-1/x}\,(p_n(x)\sin(1/x) + q_n(x)\cos(1/x))}{x^{2n}}. \tag{14}$$

2. *The limit of $sm^{(n)}(x)$ towards 0 from the right hand side is zero, i.e.,*

$$\lim_{x \to 0^+} sm^{(n)}(x) = 0. \tag{15}$$

The proof of (14) in Lemma 3.5 in PVS uses induction on n. The polynomial sequences $\{p_n\}$ and $\{q_n\}$ are defined recursively with $p_0(x) = 1$ and $q_0(x) = 0$, and for $n \in \mathbb{N}_{\ge 1}$

$$p_n(x) = p_{n-1}(x) + p'_{n-1}(x) + q_{n-1}(x) - 2nxp_n(x) \quad \text{and}$$
$$q_n(x) = q_{n-1}(x) - p_{n-1}(x) + q'_{n-1}(x) - 2nxq_{n-1}(x)$$

where $p'_{n-1}(x)$ and $q'_{n-1}(x)$ are the derivatives of $p_{n-1}(x)$ and $q_{n-1}(x)$, respectively. This required a proof that a single variate polynomial is differentiable in PVS, which was straightforward using the differentiation rules already present in the analysis library of NASALib. In fact, once $\{p_n\}$ and $\{q_n\}$ were defined in PVS, the inductive proof showing (14) made repeated use of the chain, quotient, product, and power rules already available in the analysis library.

[3] There are other, simpler, smooth but not analytic functions, but this choice will serve in the next section.

The proof of (15) in Lemma 3.5 first required showing there exists a $C_n \in \mathbb{R}$ such that, for $0 \le x \le 1$

$$|sm^{(n)}(x)| \le C_n \left| \frac{e^{-1/x}}{x^{2n}} \right|. \tag{16}$$

This result follows from the continuity of $h(x) = p_n(x)\sin(1/x) + q_n(x)\cos(1/x)$ on the interval $[0, 1]$. Using (16) and

$$\begin{aligned} \lim_{x\to 0^+} \frac{e^{-1/x}}{x^{2n}} &= \lim_{x\to\infty} \frac{x^{2n}}{e^x} \\ &= 0 \end{aligned}$$

gives the desired result. Typically, one would use induction and L'Hôpital's rule to show

$$\lim_{x\to\infty} \frac{x^{2n}}{e^x} = 0.$$

NASALib does not have L'Hôpital's rule, so a different proof for (3.1) had to be found that uses properties of the natural log, exponential function, and existing analysis rules had to be used. The proof is described as follows. For all $x \ge 0$, note that

$$\frac{x^{2n}}{e^x} = \frac{1}{e^{x-2n\ln(x)}}.$$

The function $h_1(x) = x - 2n\ln(x)$ is less than or equal to $h_2(x) = \frac{1}{2}(x-4n)+(4n-2n\ln(4n))$ for all $x \ge 4n$. This can be seen since $h_1(4n) = h_2(4n)$ and $h_1'(x) \le h_2'(x)$ for all $x \ge 4n$. Therefore for $x \ge 4n$

$$0 \le \left|e^{-h_1(x)}\right| \le \left|e^{-h_2(x)}\right|.$$

Since $\lim_{x\to\infty} e^{-h_2(x)} = 0$, $\lim_{x\to\infty} e^{-h_1(x)} = 0$, and the result is shown.

Lemma 3.5 part 1 establishes the value of $sm^{(n)}(x)$ for $x > 0$. For $x < 0$, $sm^{(n)}(x) = 0$. Also $sm^{(n)}(x)$ is continuous for $x \ne 0$, and Lemma 3.5 part 2 establishes that $sm^{(n)}(x)$ is continuous at $x = 0$. The next theorem establishes that the nth derivative of of sm at $x = 0$ is $sm^{(n)}(x) = 0$, showing smoothness at $x = 0$.

Theorem 3.6. *For function sm defined in* (13)*, the following statement holds*

1. *s is smooth, with $sm^{(n)}(0) = 0$ for each $n \in \mathbb{N}$,*
2. *sm is not analytic at $x = 0$.*

The proof of Theorem 3.6 part 1 was done by induction. The crux of the argument was the following equalities

$$\begin{aligned} sm^{(n)}(0) &= \lim_{h\to 0} \frac{sm^{(n-1)}(h) - sm^{(n-1)}(0)}{h} \\ &= \lim_{h\to 0} sm^{(n)}\left(c_{(h)}\right) \\ &= \lim_{h\to 0} sm^{(n)}(h)). \\ &= 0, \end{aligned}$$

Where the existence of $c_h \in (0, h)$ is given by the Mean Value Theorem. The conditions of the Mean Value Theorem are satisfied since $sm^{(n-1)}$ is differentiable on the open interval $(0, h)$ and continuous, on the interval $[0, h]$.

The Mean Value Theorem in NASALib's analysis library required that $sm^{(n)}$ be differentiable on the closed interval $[0, h]$, which could not be assumed, since it is exactly what is trying to be proven. This required a new Mean Value Theorem to be specified with the slightly weaker assumptions on the function:

```
mean_value_gen: THEOREM
  FORALL(f:[real->real], a:real,
  b:bb:real|bb>a):
  (derivable?[open_interval(a,b)](f) AND
  continuous?[closed_interval(a,b)](f)) IMPLIES
  EXISTS (c:real): a <  c AND c <  b AND
  deriv(f, c) * (b - a) = f(b) - f(a).
```

As a result, this more general version of the Mean Value Theorem was proven and has been added to NASALib.

For the proof of part 2 of Theorem 3.6, the proof was by contradiction. If sm was analytic at 0, by Theorem 3.4 then there would be some $r \in \mathbb{R}_{\ge 0}$ such that

$$f(x) = \sum_{k=1}^{\infty} \frac{f^{(k)}(0)}{k!} x^k, \quad \forall x \in (-r, r)\,.$$

Using part 1 of this theorem this would mean $f(x) = 0$ on the interval $(-r, r)$. This is a contradiction since $f(x) = e^{-1/x}\sin(1/x)$, for all $x > 0$, and is therefore not the zero function in any neighborhood around $x = 0$. This is a fact that a mathematician would accept without proof, but PVS required the following reasoning. For $n \in \mathbb{N}$ and

$$x_n = \frac{2}{\pi(4*n+1)},$$

$$sm(x_n) = e^{\frac{\pi(4*n+1)}{2}} \sin\left(\tfrac{\pi*4n+1}{2}\right) = e^{\frac{\pi(4*n+1)}{2}} > 0. \text{ for all } n \in \mathbb{N}.$$

Since

$$\lim_{n\to\infty} x_n = 0,$$

sm is not the zero in any open interval around $x = 0$.

Below is the PVS definition of sm, and the PVS theorem stating that it is smooth everywhere, but not analytic at 0.

```
sm(x:real): real = IF x <= 0 THEN 0
  ELSE exp(- 1 / x) * sin(1/x) ENDIF
```

```
smooth_not_analytic: THEOREM
  smooth?(sm) AND NOT analytic(0)(sm).
```

3.2 Semi-algebraic Sets and Analytic Functions

This section investigates the interaction between SA sets and real analytic functions. The goal is to show, in a sense that will be made precise, that a real analytic function leaves (or enters) an SA set at a single point, or for an interval.

First, the following lemma discusses the behavior of an analytic function around a root. This will be key to showing the favorable properties of an analytic function entering and leaving an SA set.

Lemma 3.7. *For an analytic function f at a point t with radius of convergence r, the following properties hold:*

1. *If $f(t) > 0$ then there exists an $\epsilon \in \mathbb{R}_{>0}$ such that $f(x) > 0$ for all $x \in (t-\epsilon, t+\epsilon)$*
2. *If $f(t) < 0$ then there exists an $\epsilon \in \mathbb{R}_{>0}$ such that $f(x) < 0$ for all $x \in (t-\epsilon, t+\epsilon)$*
3. *If $f(t) = 0$ then there exists an $\epsilon \in \mathbb{R}_{>0}$ such that either*
   a. *$f(x) = 0$ for all $x \in (t-\epsilon, t+\epsilon)$, or*
   b. *$f(x) \neq 0$ for $x \neq t$ and $x \in (t-\epsilon, t+\epsilon)$.*

Proof. Parts 1 and 2 follow from the fact that f is continuous. For part 3 the proof is by contradiction. Assume that $f(t) = 0$ and f is not all zero on any open interval around t. Also assume that there is a sequence $\{t_k\}_{k=1}^{\infty}$ such that $t_k \in \left(t - \frac{1}{k}, t + \frac{1}{k}\right)$, $f(t_k) = 0$ and $t_k \neq t$. Since f is analytic it takes the form in (5). Since f is non-zero on $(c_0 - t, c_0 + t)$ there must be an $n \in \mathbb{N}$ such that $f^{(n)}(t) \neq 0$. Assume that n is the minimal number that has this property. By Taylor's remainder theorem there exists a ψ_k between t and t_k, i.e., $|\psi_k - t| \leq |t_k - t|$ such that

$$\begin{aligned} f(t_k) &= \sum_{i=1}^{n-1} f^{(i)}(t)(x-\alpha)^i + f^{(n)}(\psi_k)(t-t_k) \\ &= f^{(n)}(\psi_k)(t-t_k). \end{aligned}$$

This implies $f^{(n)}(\psi_k) = 0$ since $t_k \neq 0$. Furthermore $\psi_k \to t$ since $t_k \to t$. Since f is analytic, $f^{(n)}(t)$ is continuous this means $f^{(n)}(t) = 0$, which contradiction that n is the minimal number such that $f^{(n)}(t) \neq 0$. The result is shown. □

Parts 1 and 2 of the proof above required basic properties of continuity that were found in NASALib's analysis library. Part 3 required Taylor's theorem, which was also in NASALib's analysis library.

To study the the way an analytic function comes into contact with an SA set, it is necessary to study the interaction of between the function composed with a multivariate polynomial. The next lemma shows the composition of an analytic function with a multivariate polynomial is analytic.

Lemma 3.8. *For a function $f : \mathbb{D} \to \mathbb{R}^n$, analytic at a point $c_0 \in \mathbb{R}$, the following statements are true*

1. *For any monomial $m : \mathbb{R}^n \to \mathbb{R}$. the composition $m \circ f$ is analytic.*
2. *Furthermore, for any polynomial $p : \mathbb{R}^n \to R$, the composition $p \circ f$ is analytic.*

Proof of both parts 1 and 2 of Lemma 3.8 were proved using induction. For part 1, this was done using the recursion, for a monomial $m : \mathbb{R}^n \to \mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}^n$,

$$m \circ f(x) = (\hat{m} \circ \hat{f}(x)) \cdot (c(f_0(x))^k), \tag{17}$$

where c is the coefficient of the monomial m, f_0 is the first of the functions that f is comprised of (defined in (11)), and where $\hat{m} : \mathbb{R}^{n-1} \to \mathbb{R}$ and $\hat{f} : \mathbb{R} \to \mathbb{R}^{n-1}$ are the original monomial m and function f projected on the last $n-1$ entries. In PVS , $\hat{m}$ and $\hat{f}$ are defined as

```
hat(m:mm:monomial| cons?(mm‘alpha)):
  {mm:monomial | length(mm‘alpha) =
    length(m‘alpha) - 1 } =
  (# C:= 1 , alpha := cdr[nat](m‘alpha) #)
hat(n:posnat)(f:[real -> VectorN(n)]):
  [real -> VectorN(n-1)] =
  LAMBDA(x:real): cdr(f(x)),
```

with the property in (17) specified by the lemma

```
eval_hat_equiv: LEMMA
  FORALL(n:posnat, m:monomial |
  length(m‘alpha)=n, f:[real->VectorN(n)]):
  (LAMBDA(x:real): full_eval(m)(f(x)))
  =
  (LAMBDA(x:real): m‘C * car(f(x)) ^
car[nat](m‘alpha) *
  full_eval(hat(m))(hat(n)(f)(x))).
```

With the recursion in (17) verified, the rest of the proof of Lemma (3.8), part 1 follows from applying Theorem 3.2, part 3 and Lemma 3.3.

Part 2 of Lemma (3.8) follows from the fact that that the polynomial p is the finite sum of $n \in \mathbb{N}$ monomials

$$p = m_1 + m_2 + \cdots + m_n,$$

and the composition $p \circ f(x)$ is nothing more that the sum of f composed with monomials

$$p \circ f(x) = m_1 \circ f + m_2 \circ f + \cdots + m_n \circ f.$$

By an induction argument that uses Lemma 3.2 part 1, this proof was shown in PVS.

Lemma 3.8 is very helpful, because it allows reasoning about $p \circ f$ directly as an analytic function, instead of as the composition of an analytic function and a multivariate polynomial. The next lemma describes the behavior of an analytic function around an SA set created by a conjunction of atomic polynomial formulas, at any point in the function's domain.

Lemma 3.9. *For a connected $D \subset \mathbb{R}$, a function $f : D \to \mathbb{R}^n$ that is analytic on D, and φ be a conjunction of atomic polynomial formulas $\{p_j\}_{j=1}^n$,*

$$\varphi = \bigwedge_{j=1}^{J} p_j \triangleright 0 \text{ where } \triangleright \in \{\geq, >, \leq, <\}. \tag{18}$$

For $x_0 \in D$ there exists an $\epsilon > 0$ such that either

1. *for all $0 < t < \epsilon$, $f(x_0 + t) \in S(\varphi)$, or*
2. *for all $0 < t < \epsilon$, $f(x_0 + t) \notin S(\varphi)$.*

Because of the result in Lemma 3.8, this can be proven as a simple extension of Lemma 3.7. For each p_j in the conjunction (18), there is an ϵ_j such that there are no roots of $p_i \circ f$ on $(x_0, x_0 + \epsilon)$ for any $i \leq n$. From this, it was straightforward to show that there exists an $\epsilon_{\min} > 0$ such that no $p_i \circ f$ in for $i \in \mathbb{N}_{\leq J}$ has a root on the interval $(x_0, x_0 + \epsilon_{\min})$:

```
min_eps LEMMA
  FORALL (m:meeting,x0:real,
  f:(analytic?(atom_max(m),x0))):
  EXISTS(eps_min:posreal):
  FORALL(i:below(length(m)), t:real):
  (x0 < t AND t < x0 + eps_min) IMPLIES
  full_eval(nth(m,i)`poly)(f(x0 + t)) /= 0.
```

With the existence of this $\epsilon_{\min}$, it is clear that the truth value of φ in (18) is constant on the interval $(x_0, x_0 + \epsilon_{\min})$, finishing the proof.

With Lemma 3.9 above, the main result of the paper is ready to be shown. The next two theorems classify how an analytic function can leave or enter an SA set.

Theorem 3.10. *For a connected $D \subset \mathbb{R}$, a function $f : D \to \mathbb{R}^n$, that is analytic on D, a semi algebraic set $S(\varphi)$ where φ is defined in Equation (3), and a $x_0 \in \mathbb{R}$ such that $f(x_0) \in S(\varphi)$. Then one of the following cases is true*

1. *$f(x) \in S$ for all $x \geq x_0$,*
2. *for $x^* = \inf\{x \in D | x > x_0, f(x) \notin S(\varphi)\}$, $f(x^*) \notin S$, and there exists an ϵ such that $f(x^* + t) \in S(\varphi)$ for all $0 < t < \epsilon$, or*
3. *for $x^* = \inf\{x \in D | x > x_0, f(x) \notin S(\varphi)\}$, there exists an ϵ such that $f(x^* + t) \notin S(\varphi)$ for all $0 < t < \epsilon$.*

Note that if the first condition is not satisfied,

$$t^* = \inf\{x \in D | x > x_0, f(x) \notin S(\varphi)\}$$

exists. By using Lemma (3.9), an $\epsilon_{\min}$ can be found such that for each $i \in \mathbb{N}_{\leq I}$ the conjunction

$$\bigwedge_{j=1}^{J} p_{ij} \triangleright 0$$

has a constant truth value on the interval $(x^*, x^* + \epsilon_{\min})$. The result follows from this. In PVS the theorem is specified as

```
clean_exit: THEOREM
  FORALL(j:joining, x0:real,
  f:(analytic?(meet_max(j),x0))):
  semi_alg(j)(meet_max(j))(f(x0)) IMPLIES (
  % Condition 1
  (FORALL(x:real): x >=  x0
  IMPLIES semi_alg(j)(meet_max(j))(f(x)) OR
  % Condition 2
  EXISTS(eps:posreal):
  FORALL(t:real): inf({xx:real |
  NOT semi_alg(j)(meet_max(j))(f(xx))}) < t
  AND t <  inf({xx:real |
  NOT semi_alg(j)(meet_max(j))(f(xx))}) + t
  IMPLIES semi_alg(j)(meet_max(j))(f(t)) OR
  %  Condition 3
  EXISTS(eps:posreal): FORALL(t:real):
  inf({xx:real |
```

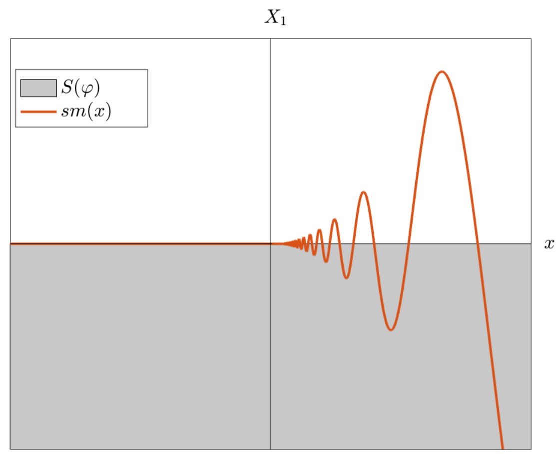

Figure 2. A visualization of Example 3.12. The function sm defined in Equation (13) is smooth, not analytic, and has infinity many points inside and outside of the SA set $S(\varphi)$ around $x = 0$, violating the conclusion of Theorem 3.10.

```
  NOT semi_alg(j)(meet_max(j))(f(xx))}) < t AND
  t> inf(xx:real |
  NOT semi_alg(j)(meet_max(j))(f(xx))) + t
  IMPLIES NOT semi_alg(j)(meet_max(j))(f(t)).
```

Theorem 3.11. *For a connected $D \subset \mathbb{R}$, a function $f : D \to \mathbb{R}^n$, where that is analytic on D, a semi algebraic set $S(\varphi)$ where φ is defined in Equation (3), and a $x_0 \in \mathbb{R}$ such that $f(x_0) \notin S(\varphi)$. Then one of the following cases is true*

1. *$f(x) \notin S(\varphi)$ for all $x \geq x_0$,*
2. *for $x^* = \inf\{x \in D | x > x_0, f(x) \in S(\varphi)\}$ $f(x^*) \in S$ and there exists an ϵ such that $f(x^* + t) \notin S(\varphi)$ for all $0 < t < \epsilon$, or*
3. *for $x^* = \inf\{x \in D | x > x_0, f(x) \in S(\varphi)\}$ there exists an ϵ such that $f(x^* + t) \in S$ for all $0 < t < \epsilon$.*

A proof of Theorem 3.11 can be found by applying Theorem 3.10 with f and S^c. These theorems show that an analytic function leaves or enters an SA set in a "clean" way, i.e., at a a single point, or for a complete interval of time. When the assumption that f is weakened from analytic to smooth, this result does not hold, as shown in the following example.

Example 3.12. Consider the SA set $S(\varphi)$ where $\varphi = (X_1 \leq 0)$, and the function $sm : \mathbb{R} \to \mathbb{R}$ is defined in Equation (13), see Figure 2. Using Theorem 3.6, sm is smooth, but not analytic. For all $x \leq 0$, $sm(x) \in S(\varphi)$. Furthermore, $x^* = \inf\{x \in \mathbb{R} | sm(x) \notin S(\varphi)\} = 0$ since for $x_n = \frac{1}{\pi(n+1)}$, $sm(x_n) = 0 \in S$ and $x_n \to 0$. On the other hand, for $y_n = \frac{2}{\pi(4n+1)}$, $sm(y_n) = e^{-1/y_n} \notin S$. Because of the infinite oscillations around the origin, the conclusions in Theorem 3.10 are not satisfied, i.e., for all $\epsilon > 0$ there exists $0 < x_1, x_2 < \epsilon$ such that $x_1 \in S(\varphi)$

and $x_2 \notin S(\varphi)$. In PVS, this counter example is shown in the lemma below

```
% Define variables
p1:(mv_standard_form?) =
  (: (# C:=1, alpha:=(: 1 :) #) :)
atom1: atomic_poly =
  (# poly :=p1, ineq:= <= #)
SA: set[VectorN(1)] =
  semi_alg( (: (: atom1 :) :))(2)
% Smoothness is not enough for "clean break"
not_clean_break: LEMMA
  inf(xx:real | NOT SA((: sm(xx) :))) = 0 AND
  EXISTS(xn,yn:sequence[real]):
  convergence(xn,0) AND convergence(yn,0) AND
  FORALL(i:nat): SA((: sm(xn(i)) :)) AND
    xn(i) > 0 AND
    NOT SA((: sm(xn(i)) :) ) AND
    yn(i) > 0
```

4 Related Work

The development of analytic functions and SA sets in PVS is a part of an ongoing project to implement a differential dynamic logic (DDL) in PVS. The purpose of this formalization is to help reason about hybrid systems, i.e., systems that have both discrete variables and continuous variables, the latter defined by solutions to ordinary differential equations, without having to explicitly solve the differential equations in some cases [28–30]. An example of an implementation of DDL is a theorem prover called KeYmaera X, which is a formal verification tool to interactively and formally reason about hybrid systems [10]. To verify the soundness of DDL, it has been formalized in both Isabelle and Coq [3].

Often, solving the differential equation explicitly is overly cumbersome or not feasible, so it is easier to reason about the solution without finding it. The deduction that the solution of an ODE is analytic is possible with general assumptions about the underlying ODEs. DDL allows this reasoning but requires knowledge of how such a function interacts with constraints modeled as SA sets. There has been significant research done on reasoning about differential invariants in DDL, where the domain of the differential equation and a set of system constraints are modeled as SA sets. Of particular interest is how such a solution leaves and enters a set of constraints, motivating this work. [12, 31–33]

Although the interactions between analytic functions and SA sets have been studied (e.g., [19]), to the best of the author's knowledge, there is no known formalization of these behaviors. A constructive formalization of SA sets was undertaken in Coq, to specify and formally verify the cylindrical algebraic decomposition (CAD) algorithm, which takes a set of polynomials and decomposes their domain space into SA sets, where the sign of each polynomial is constant [7, 8]. This is one of the most fundamental and important algorithms in real algebraic geometry. In addition to the CAD implementation [20, 21], multivariate polynomials have been implemented and used in Coq several ways [1, 4, 6]. In Isabelle/Hol, formalization of multivariate polynomials [13] and the CAD algorithm [17] are active areas of research. Implementation of univariate polynomials was done in the formalization of Sturm's theorem in HolLight [14] and in the PVS implementation of Sturm's and Tarski's theorems [23]. Multivariate Bernstein polynomials have also been formalized in PVS [22], which is a powerful tool for approximating continuous functions.

5 Conclusions and Future Work

This paper describes the formalization of multivariate polynomials with a sparse representation and semi-algebraic sets in PVS, as well as real analytic functions and their interactions with SA sets.

The primary goal of this work is to eventually formalize a version of DDL that can be used in an interactive way in PVS. To this end, there is much interesting work to be done. The theory of differential equations must be formalized including, at the least, the existence and uniqueness theorems which guarantee a real analytic solution to a differential equation exists. The soundness of the differential rules in DDL will also need to be shown, which will depend on the theory of differential equations.

With respect to the SA set formalization there are several directions that the research can be extended. The current embedding in PVS assumes the an SA set is already in disjunctive normal form. An extension that allows conditional statements of polynomial formulas would add to the expressiveness of the library, and and implementation of a disjunctive normal form transformation would make this extension fit into the theory that has been established in this paper.

Additionally, one of the fundamental theorems in real algebraic geometry is the Tarski-Seidenberg Theorem, which says that every *quantified* formula over multivariate polynomial constraints is equivalent to a *quantifier-free* formula used to define semi-algebraic sets. A proof of this theorem, as well as specification and proof of CAD methods for quantifier elimination, are long-term goals for the PVS formalization. As noted in Section 4, this is an on-going area of research in many theorem provers.

References

[1] Sophie Bernard, Yves Bertot, Laurence Rideau, and Pierre-Yves Strub. 2016. Formal proofs of transcendence for e and pi as an application of multivariate and symmetric polynomials. In *Proceedings of the 5th ACM SIGPLAN Conference on Certified Programs and Proofs*. 76–87.

[2] Jacek Bochnak, Michel Coste, and Marie-Françoise Roy. 2013. *Real algebraic geometry*. Vol. 36. Springer Science & Business Media.

[3] Brandon Bohrer, Vincent Rahli, Ivana Vukotic, Marcus Völp, and André Platzer. 2017. Formally verified differential dynamic logic. In *Proceedings of the 6th ACM SIGPLAN Conference on Certified Programs and*

Proofs. 208–221.

[4] Cyril Cohen. 2013. Pragmatic quotient types in Coq. In *International Conference on Interactive Theorem Proving*. Springer, 213–228.

[5] Brian A Davey and Hilary A Priestley. 2002. *Introduction to lattices and order*. Cambridge university press.

[6] Maxime Dénès, Anders Mörtberg, and Vincent Siles. 2012. A refinement-based approach to computational algebra in Coq. In *International Conference on Interactive Theorem Proving*. Springer, 83–98.

[7] Boris Djalal. 2018. A constructive formalisation of Semi-algebraic sets and functions. In *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs*. 240–251.

[8] Boris Djalal. 2018. *Formalisations en Coq pour la décision de problèmes en géométrie algébrique réelle*. Ph.D. Dissertation. Côte d'Azur.

[9] Gerald B Folland. 1995. *Introduction to partial differential equations*. Vol. 102. Princeton university press.

[10] Nathan Fulton, Stefan Mitsch, Jan-David Quesel, Marcus Völp, and André Platzer. 2015. KeYmaera X: An axiomatic tactical theorem prover for hybrid systems. In *International Conference on Automated Deduction*. Springer, 527–538.

[11] Khalil Ghorbal, Jean-Baptiste Jeannin, Erik Zawadzki, André Platzer, Geoffrey J Gordon, and Peter Capell. 2014. Hybrid theorem proving of aerospace systems: Applications and challenges. *Journal of Aerospace Information Systems* 11, 10 (2014), 702–713.

[12] Khalil Ghorbal, Andrew Sogokon, and André Platzer. 2017. A hierarchy of proof rules for checking positive invariance of algebraic and semi-algebraic sets. *Computer Languages, Systems & Structures* 47 (2017), 19–43.

[13] Florian Haftmann, Andreas Lochbihler, and Wolfgang Schreiner. 2014. Towards abstract and executable multivariate polynomials in Isabelle. In *Isabelle Workshop*, Vol. 201.

[14] John Harrison. 1997. Verifying the accuracy of polynomial approximations in HOL. In *International Conference on Theorem Proving in Higher Order Logics*. Springer, 137–152.

[15] Hassan K Khalil and Jessy W Grizzle. 2002. *Nonlinear systems*. Vol. 3. Prentice hall Upper Saddle River, NJ.

[16] Steven G Krantz and Harold R Parks. 2002. *A primer of real analytic functions*. Springer Science & Business Media.

[17] Wenda Li. 2019. *Towards justifying computer algebra algorithms in Isabelle/HOL*. Ph.D. Dissertation. University of Cambridge.

[18] Jiang Liu, Naijun Zhan, and Hengjun Zhao. 2011. Computing semi-algebraic invariants for polynomial dynamical systems. In *Proceedings of the ninth ACM international conference on Embedded software*. 97–106.

[19] Jiang Liu, Naijun Zhan, and Hengjun Zhao. 2011. Computing semi-algebraic invariants for polynomial dynamical systems. In *Proceedings of the ninth ACM international conference on Embedded software*. 97–106.

[20] Assia Mahboubi. 2006. Programming and certifying a CAD algorithm in the Coq system. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[21] Assia Mahboubi. 2007. Implementing the cylindrical algebraic decomposition within the Coq system. *Mathematical Structures in Computer Science* 17, 1 (2007), 99.

[22] César Muñoz and Anthony Narkawicz. 2013. Formalization of Bernstein polynomials and applications to global optimization. *Journal of Automated Reasoning* 51, 2 (2013), 151–196.

[23] Anthony Narkawicz, César Muñoz, and Aaron Dutle. 2015. Formally-verified decision procedures for univariate polynomial computation based on Sturm's and Tarski's theorems. *Journal of Automated Reasoning* 54, 4 (2015), 285–326.

[24] Sam Owre, John M Rushby, and Natarajan Shankar. 1992. PVS: A prototype verification system. In *International Conference on Automated Deduction*. Springer, 748–752.

[25] Sam Owre and Natarajan Shankar. 2008. A brief overview of PVS. In *International Conference on Theorem Proving in Higher Order Logics*. Springer, 22–27.

[26] André Platzer. 2008. Differential dynamic logic for hybrid systems. *Journal of Automated Reasoning* 41, 2 (2008), 143–189.

[27] André Platzer. 2018. *Logical foundations of cyber-physical systems*. Vol. 662. Springer.

[28] André Platzer and Jan-David Quesel. 2008. KeYmaera: A hybrid theorem prover for hybrid systems (system description). In *International Joint Conference on Automated Reasoning*. Springer, 171–178.

[29] André Platzer and Yong Kiam Tan. 2018. Differential equation axiomatization: The impressive power of differential ghosts. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*. 819–828.

[30] Jan-David Quesel, Stefan Mitsch, Sarah Loos, Nikos Aréchiga, and André Platzer. 2016. How to model and prove hybrid systems with KeYmaera: a tutorial on safety. *International Journal on Software Tools for Technology Transfer* 18, 1 (2016), 67–91.

[31] Andrew Sogokon, Khalil Ghorbal, Paul B Jackson, and André Platzer. 2016. A method for invariant generation for polynomial continuous systems. In *International Conference on Verification, Model Checking, and Abstract Interpretation*. Springer, 268–288.

[32] Andrew Sogokon and Paul B Jackson. 2015. Direct formal verification of liveness properties in continuous and hybrid dynamical systems. In *International Symposium on Formal Methods*. Springer, 514–531.

[33] Andrew Sogokon, Stefan Mitsch, Yong Kiam Tan, Katherine Cordwell, and André Platzer. 2019. Pegasus: A framework for sound continuous invariant generation. In *International Symposium on Formal Methods*. Springer, 138–157.

[34] Brian L Stevens, Frank L Lewis, and Eric N Johnson. 2015. *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons.

[35] Morris. Tenenbaum and Harry Pollard. 1963. *Ordinary differential equations: an elementary textbook for students of mathematics, engineering, and the sciences*. Dover Publications.

[36] Richard Zippel. 1993. *Effective Polynomial Computation*. Springer US.