Abstract Partial Cylindrical Algebraic Decomposition I: The Lifting Phase*

Grant Olney Passmore[1,2] and Paul B. Jackson[2]
grant.passmore@cl.cam.ac.uk, pbj@inf.ed.ac.uk

[1] Clare Hall, University of Cambridge
[2] LFCS, University of Edinburgh

Abstract. Though decidable, the theory of real closed fields (**RCF**) is fundamentally infeasible. This is unfortunate, as automatic proof methods for nonlinear real arithmetic are crucially needed in both formalised mathematics and the verification of real-world cyber-physical systems. Consequently, many researchers have proposed fast, sound but incomplete **RCF** proof procedures which are useful in various practical applications. We show how such practically useful, sound but incomplete **RCF** proof methods may be systematically utilised in the context of a complete **RCF** proof method without sacrificing its completeness. In particular, we present an extension of the **RCF** quantifier elimination method Partial CAD (P-CAD) which uses incomplete $\exists$ **RCF** proof procedures to "short-circuit" expensive computations during the lifting phase of P-CAD. We present the theoretical framework and preliminary experiments arising from an implementation in our **RCF** proof tool **RAHD**.

Keywords: decision procedures, nonlinear arithmetic, real closed fields

1 Introduction

Tarski's theorem that the elementary theory of real closed fields (**RCF**) admits effective elimination of quantifiers is one of the longstanding hallmarks of mathematical logic [13]. From this result, the decidability of elementary algebra and geometry readily follow, and a most tantalising situation arises: In principle, every elementary arithmetical conjecture over finite-dimensional real and complex spaces may be decided simply by formalising the conjecture and asking a computer of its truth. So why then do we still not know how many unit hyperspheres may kiss[3] in five dimensions? Is it 41? 42?

The issue is one of complexity. Though decidable, **RCF** is fundamentally infeasible. Due to Davenport-Heintz [5], it is known that there exist families of

* This paper reports on work presented in Chapters 7 and 8 of the first author's 2011 University of Edinburgh Ph.D. thesis [9], supervised by the second author. This research was supported by the EPSRC [grant numbers EP/I011005/1 and EP/I010335/1]. We thank the referees for their helpful comments and suggestions.

[3] See, e.g., [11] for background on the kissing problem for n-dimensional hyperspheres.

n-dimensional **RCF** formulas of length $O(n)$ whose only quantifier-free equivalences must contain polynomials of degree $2^{2^{\Omega(n)}}$ and of length $2^{2^{\Omega(n)}}$. Nevertheless, there are countless examples of difficult, high-dimensional **RCF** problems solved in mathematical and engineering practice. What is the disconnect? (1) **RCF** problems solved in practice are most often solved using an ad hoc combination of methods, *not* by a general decision method. (2) **RCF** problems arising in practice commonly have structural properties dictated by the application domain from which they originated. Such structural properties can often be exploited making such problems more amenable to analysis and pushing them within the reaches of restricted, more efficient variants of known decision methods.

With this in mind, many researchers have proposed fast, sound but *incomplete* **RCF** proof procedures, many of them being of substantial practical use [1,7,14,10,12,6,4]. This is especially true for formal methods, where improved automated **RCF** proof methods are needed in the formal verification of cyber-physical systems. In these cases, as the **RCF** problems to be analysed are usually machine-generated (and incomprehensibly large), incomplete proof procedures can go a long way. For example, there is no denying the fact that applying a full quantifier elimination algorithm to decide the falsity of a formula such as $\exists x_1, \ldots, x_{100} \in \mathbb{R}\ (x_1 * x_1 + \ldots + x_{100} * x_{100} < 0)$ is an obvious misappropriation of resources. While such an example may seem contrived, consider the fact that when an **RCF** proof method is used in formal verification efforts, it is often fed huge collections of machine-generated formulas which may be (un)satisfiable for extremely simple reasons. Ideally, one would like to be able to use fast, sound but incomplete proof procedures as much as possible, falling back on the far more computationally expensive complete methods only when necessary. It would be desirable to have a principled manner in which incomplete proof methods could be used to improve the performance of a complete method without sacrificing its completeness.

We present *Abstract Partial Cylindrical Algebraic Decomposition* (AP-CAD), an extension of the **RCF** quantifier elimination procedure partial CAD. In AP-CAD, arbitrary sound but possibly incomplete $\exists$ **RCF** proof procedures may be used to "short-circuit" certain expensive computations during CAD construction. This is done in such a way that the completeness of the combined proof method is guaranteed. We restrict our AP-CAD presentation to the practically useful case of $\exists$ **RCF**. We have implemented AP-CAD within our **RCF** proof tool **RAHD** [9] for the case of full-dimensional cell decompositions and present experiments. **RAHD** contains many **RCF** proof methods and allows users to combine them into their own heuristic **RCF** proof procedures through a *proof strategy language*. This is ideal for AP-CAD, as the proof procedure parameters used by AP-CAD can be formally realised as **RAHD** proof strategies.

2 CAD Preliminaries

For a detailed account of CAD, we refer the reader to [2]. We present only the background on (P-)CAD required to understand AP-CAD for $\exists$ **RCF**. P-

CAD is currently the most efficient known general quantifier elimination method for **RCF**[4]. An important fact is that the complexity of the (P-)CAD decision algorithm is doubly exponential in the dimension (number of variables) of its input formula. Generally, the most expensive phase of the (P-)CAD algorithm is the so-called "lifting phase." Let us fix some notation.

A *semialgebraic set* is a subset of $\mathbb{R}^n$ definable by a quantifier-free formula in the language of ordered rings. A *region* of $\mathbb{R}^n$ is a connected component of $\mathbb{R}^n$. An *algebraic decomposition* of $\mathbb{R}^n$ is a decomposition of $\mathbb{R}^n$ into finitely many semialgebraic regions. A *cylindrical algebraic decomposition* is a special type of algebraic decomposition whose regions are in a sense "well-behaved" with respect to projections onto lower dimensions. A *cell* is a region of a CAD.

Before delving into technical details, let us discuss how we can use a CAD to make $\exists$ **RCF** decisions. By "the polynomials of (an $\exists$ **RCF** formula) φ," we shall mean the collection of polynomials obtained by zeroing the RHS of every atom in φ through subtracting the RHS from both sides. We assume each such $\exists$ **RCF** formula is in prenex normal form, so that it is an $\exists$-closed boolean combination of *sign conditions*, i.e., of atoms of the form $(p \odot 0)$ with $p \in \mathbb{Z}[x_1, \ldots, x_n]$, $\odot \in \{<, \leq, =, \geq, >\}$. We use $QF(\varphi)$ to mean the quantifier-free matrix of φ.

The key point is that if we have in hand a suitable CAD $C = \{c_1, \ldots, c_m\} \subset 2^{\mathbb{R}^n}$ derived from an $\exists$ **RCF** formula φ, we can decide the truth of φ from the CAD directly. The reason is simple: C will have the property that every polynomial of φ has *constant sign* on each c_i, i.e., given p a polynomial of φ and a c_i a cell, it shall hold that $\forall \boldsymbol{r} \in c_i(p(\boldsymbol{r}) = 0) \vee \forall \boldsymbol{r} \in c_i(p(\boldsymbol{r}) > 0) \vee \forall \boldsymbol{r} \in c_i(p(\boldsymbol{r}) < 0)$. Consequently, $QF(\varphi)$ has constant truth value at every point in a given cell. Thus, to decide φ, we simply substitute a single sample point from each c_i into $QF(\varphi)$ and see if it ever evaluates to **true**. It will evaluate to **true** on at least one sample point if and only if φ is **true** over $\mathbb{R}^n$.

We shall define CAD by induction on dimension[5]. A CAD of $\mathbb{R}$ is a decomposition of $\mathbb{R}$ into finitely many cells $c_i \subseteq \mathbb{R}$ s.t. each c_i is of the form (i) $\{\alpha_1\}$, or (ii) $]\alpha_1, \alpha_2[$, or (iii) $]\text{-}\infty, \alpha_1[$ or $]\alpha_1, +\infty[$ for algebraic real numbers α_i. Let $\mathcal{A}$ be a region of $\mathbb{R}^i$. We call $\mathcal{A} \times \mathbb{R}$ the *cylinder over* $\mathcal{A}$ and denote it by $Z(\mathcal{A})$.

Definition 1 (Stack). *Let $f_1, \ldots, f_k \in \mathcal{C}(\mathcal{A}, \mathbb{R})$. That is, f_j is a continuous function from $\mathcal{A}$ to $\mathbb{R}$. Furthermore, suppose that the images of the f_j are ordered over $\mathcal{A}$ s.t. $\forall \alpha \in \mathcal{A}(f_j(\alpha) < f_{j+1}(\alpha))$. Then, $f_1, \ldots, f_k$ induce a* stack $\mathfrak{S}$ *over $\mathcal{A}$, where $\mathfrak{S}$ is a decomposition of $Z(\mathcal{A})$ into $2k+1$ regions of the following form:*

- $r_1 = \{\langle \alpha, x \rangle \mid \alpha \in \mathcal{A}, x < f_1(\alpha)\}$,
  $r_3 = \{\langle \alpha, x \rangle \mid \alpha \in \mathcal{A}, f_1(\alpha) < x < f_2(\alpha)\}$,
  $\vdots$

[4] See [8] for an explanation as to why P-CAD is also currently the best known general decision method for practical $\exists$ **RCF** problems, despite the fact that $\exists$ **RCF** has a theoretical exponential speed-up over **RCF**.

[5] We shall speak freely of the symbolic manipulation and arithmetic of (irrational) real algebraic numbers. See, e.g., [2] for an algorithmic account.

$r_{2k-1} = \{\langle \alpha, x\rangle \mid \alpha \in \mathcal{A}, f_{k-1}(\alpha) < x < f_k(\alpha)\}$,
$r_{2k+1} = \{\langle \alpha, x\rangle \mid \alpha \in \mathcal{A}, f_k(\alpha) < x\}$,

- $r_2 = \{\langle \alpha, x\rangle \mid \alpha \in \mathcal{A}, x = f_1(\alpha)\}$,

$\vdots$

$r_{2k} = \{\langle \alpha, x\rangle \mid \alpha \in \mathcal{A}, x = f_k(\alpha)\}$.

A CAD of $\mathbb{R}^{i+1}$ will be obtained from a CAD C of $\mathbb{R}^i$ by constructing a stack over every cell in C.

Definition 2 (CAD in $\mathbb{R}^{i+1}$). *An algebraic decomposition C_{i+1} of $\mathbb{R}^{i+1}$ is a CAD iff C_{i+1} is a* union of stacks $C_{i+1} = \bigcup_{j=1}^{k} w_j$, *s.t. the stack w_j is constructed over cell c_j in a CAD $C_i = \{c_1, \ldots, c_k\}$ of $\mathbb{R}^i$.*

The P-invariance property will allow us to use CADs to make $\exists$ **RCF** decisions.

Definition 3 (P-invariance). *Let $P = \{p_1, \ldots, p_k\} \subset \mathbb{Z}[x_1, \ldots, x_n]$ and $\mathcal{A}$ be a region of $\mathbb{R}^n$. Then, we say $\mathcal{A}$ is* P-invariant *iff every member of P has constant sign on $\mathcal{A}$. That is given any $p_i \in P$,*

$$\forall \boldsymbol{r} \in \mathcal{A}(p_i(\boldsymbol{r}) = 0) \quad \vee \quad \forall \boldsymbol{r} \in \mathcal{A}(p_i(\boldsymbol{r}) > 0) \quad \vee \quad \forall \boldsymbol{r} \in \mathcal{A}(p_i(\boldsymbol{r}) < 0).$$

Given a CAD C, we say C is P-invariant iff every cell of C is P-invariant.

2.1 CAD Construction and Evaluation for $\exists$ RCF

The use of CADs for deciding $\exists$ **RCF** sentences will take place in four steps. In what follows, φ is an $\exists$ **RCF** sentence and $P = \{p_1, \ldots, p_k\} \subset \mathbb{Z}[x_1, \ldots, x_n]$ is the collection of polynomials of φ.

Projection The *projection phase* will begin with P and iteratively apply a *projection operator* $Proj_i$ of the form $Proj_i : 2^{\mathbb{Z}[x_1,\ldots,x_{i+1}]} \to 2^{\mathbb{Z}[x_1,\ldots,x_i]}$ until a set of polynomials is obtained over $\mathbb{Z}[x_1]$. This process will consist of levels, one for each dimension, s.t. at each level i we will have what is called a *level-i projection set*, $P_i \subset \mathbb{Z}[x_1, \ldots, x_i]$. These level-$i$ projection sets will have a special property: If we have a P_i-invariant CAD of $\mathbb{R}^i$, then we can use this CAD to construct a P_{i+1}-invariant CAD of $\mathbb{R}^{i+1}$.

Base The *base phase* consists of computing a P_1-invariant CAD of $\mathbb{R}^1$, implicitly described as a sequence of sample points, one for each cell in the CAD. This can be done by *univariate real root isolation* and basic machinery for arithmetic with real algebraic numbers. Let us suppose we have done this and our sequence of sample points is $\boldsymbol{s_1} < \boldsymbol{s_2} < \ldots < \boldsymbol{s_{2m+1}}$.

Lifting The *lifting phase* will take an implicit description of a P_1-invariant CAD of $\mathbb{R}^1$ and progressively transform it into an implicit description of P_n-invariant CAD of $\mathbb{R}^n$. Let $C = \{c_1, \ldots, c_m\}$ be the P_i-invariant CAD for $\mathbb{R}^i$ which we will lift to a P_{i+1}-invariant CAD of $\mathbb{R}^{i+1}$. Let $S = \{\boldsymbol{s_1}, \ldots, \boldsymbol{s_m}\}$ be our set of sample points, one from each cell in C. Then, for each cell c_j, we will use the sample point $\boldsymbol{s_j} \in c_j$ to construct a set of sample points in $\mathbb{R}^{i+1}$ corresponding to a *stack over c_j*:

1. As $\boldsymbol{s_j} \in \mathbb{R}^i$, we have that $\boldsymbol{s_j} = \langle r_1, \ldots, r_i \rangle$ for some $r_1, \ldots, r_i \in \mathbb{R}$.
2. Let $P_{i+1}[\boldsymbol{s_j}]$ denote $P_{i+1}[x_1 \mapsto r_1, x_2 \mapsto r_2, \ldots, x_i \mapsto r_i]$. Then $P_{i+1}[\boldsymbol{s_j}] \subset \mathbb{Z}[x_{i+1}]$ is a univariate family of polynomials.
3. Using the same process as we did in the base phase, compute a $P_{i+1}[\boldsymbol{s_j}]$-invariant CAD of $\mathbb{R}^1$. Let this CAD be represented by a sequence of sample points $\boldsymbol{t_1} < \boldsymbol{t_2} < \ldots < \boldsymbol{t_{2v+1}} \in \mathbb{R}$.
4. Then, the *stack over* c_j will be represented by the set of $2v+1$ sample points obtained by appending each $\boldsymbol{t_j}$ to the lower-dimensional sample point $\boldsymbol{s_j}$. That is, our stack over c_j will be represented by the following sequence of sample points $\boldsymbol{z_1}, \ldots, \boldsymbol{z_{2v+1}}$ in $\mathbb{R}^{i+1}$: $\boldsymbol{z_1} = \langle r_1, \ldots, r_i, \boldsymbol{t_1} \rangle$, $\boldsymbol{z_2} = \langle r_1, \ldots, r_i, \boldsymbol{t_2} \rangle$, …, $\boldsymbol{z_{2v+1}} = \langle r_1, \ldots, r_i, \boldsymbol{t_{2v+1}} \rangle$.

In the above construction, we call the cell c_j (or the sample point representing it, $\boldsymbol{s_j}$) the *parent* of the stack $\{\boldsymbol{z_1}, \ldots, \boldsymbol{z_{2v+1}}\}$.

Evaluation Let $S = \{\boldsymbol{s_1}, \ldots, \boldsymbol{s_m}\} \subset \mathbb{R}^n$ be our final set of sample points. Return the boolean value $\bigvee_{\boldsymbol{r} \in S} QF(\varphi)[\boldsymbol{r}]$.

2.2 Partial CAD

Let us now sketch the idea of partial CAD, due to Collins and Hong [3]. As it stands, the CAD construction algorithm will build a P-invariant CAD induced by the polynomials P of an $\exists$ **RCF** formula φ without paying any attention to the logical content of the formula itself. But, when performing lifting, i.e., constructing a stack of regions of $\mathbb{R}^{i+1}$ over a lower-dimensional cell $c_j \subset \mathbb{R}^i$, we may be easily able to see — simply by substitution and evaluation — that the formula $QF(\varphi)$ could *never* be satisfied over c_j. For instance, let $QF(\varphi) = \left((x_4^4 + x_3x_2^3 + 3x_1 > 2x_1^4) \wedge (x_1^2 > x_2 + x_3)\right)$. If c_j is a cell in a P_3-invariant CAD of $\mathbb{R}^3$ represented by the sample point $\boldsymbol{s_j} = \langle 0, 1, 5 \rangle$, then we can see $QF(\varphi)$ will *never* be satisfied over a cell in a stack which is a child of c_j. Thus, we need not lift over c_j and can eliminate it.

This is the idea behind *partial CAD* when applied to $\exists$ **RCF** formulas: Before lifting over a cell in a CAD of $\mathbb{R}^i$, check if there are any atoms in your formula only involving the variables $x_1, \ldots, x_i$. If so, then perform *partial* evaluation of your formula by evaluating those atoms upon your sample point in $\mathbb{R}^i$, and then use simple propositional reasoning to try to deduce the truth of your formula. This can also allow us to find a *satisfying* assignment for the variables in $QF(\varphi)$ without constructing a whole CAD. For instance, let $QF(\varphi) = \left((x_4^4 + x_3x_2^3 + 3x_1 > 2x_1^4) \vee (x_1^2 < x_2)\right)$. If c_j is a cell in a P_2-invariant CAD of $\mathbb{R}^2$ represented by the sample point $\boldsymbol{s_j} = \langle -1, 2 \rangle$, then we can see immediately by substitution that $QF(\varphi)$ is satisfiable over $\mathbb{R}^4$. As a witness to this satisfiability, we may return $\langle -1, 2, r_3, r_4 \rangle$ where $r_3, r_4 \in \mathbb{R}$ are arbitrary.

3 Abstract Partial CAD

From a high level of abstraction, we can see partial CAD to be normal CAD augmented with three pieces of algorithmic data:

1. A strategy for selecting lower-dimensional cells to use for evaluating lower-dimensional atoms in our input formula,
2. An algorithm which when given a cell c_j will construct a formula $F(c_j)$ which, if it both has a truth value and is decided, can be used to tell (i) if the cell c_j can be thrown away (i.e., $F(c_j)$ is decided to be **false**), or (ii) if a satisfying assignment for our formula can be extracted already from a lower-dimensional cell (i.e., $F(c_j)$ is decided to be **true**),
3. A proof procedure which will be used to decide the formulas $F(c_j)$ generated by the algorithm above.

In fact, in their original paper on partial CAD, Collins and Hong make the point that different cell selection strategies could be used and even implement and experiment with a number of them[6]. For partial CAD restricted to $\exists$ **RCF**, these three pieces of algorithmic data described above would be:

1. Select cells $c_i \in C$ in some specified enumeration order (specified by s): $c_{s(1)}, c_{s(2)}, c_{s(3)}, \ldots$.
2. Given a cell c_j represented by a sample point $\boldsymbol{s_j} = \langle r_1, \ldots, r_i \rangle \in \mathbb{R}^i$, the formula $F(c_j)$ will be constructed from our original $\exists$ **RCF** formula φ by the following process:
   (a) Let φ' be $QF(\varphi)$ augmented by instantiating x_1 with r_1, x_2 with r_2, ...
   (b) Evaluate all variable-free atoms in φ' to obtain a new formula φ''.
   (c) Replace all (unique) variable-containing atoms in φ'' with fresh propositional variables to obtain a new formula $F(c_j)$.
3. Use a propositional logic proof procedure to attempt to decide $F(c_j)$.

If $F(c_j)$ is **false** (i.e., unsatisfiable), cell c_j can be abandoned and we need not lift over it. If $F(c_j)$ is **true** (i.e., tautologous), then we can extract a witness to the truth of φ from the sample point $\boldsymbol{s_j}$. Otherwise, we lift over c_j. These three pieces of data give us the widely-used partial CAD of Collins and Hong. But, from this point of view, we see that there are *many other choices* we could make.

3.1 Stages, Theatres and Lifting

The fundamental notion of AP-CAD will be that of an *stage*[7]. Let $\mathcal{L}_{\exists OR}$ be the fragment of the language of ordered rings consisting of purely $\exists$ prenex sentences.

Definition 4 (Stage). *A* stage $\mathfrak{A} = \langle \langle \mathbb{S}, w \rangle, \mathbb{F}, \mathbb{P} \rangle$ *will be given by three pieces of algorithmic data. We describe a stage by how it acts in the context of a fixed (but arbitrary) i-dimensional space $\mathbb{R}^i$. These data are as follows:*

[6] For Collins and Hong, a cell selection strategy selects *single* cells in some specified order. In Abstract Partial CAD, cell selection strategies will select *sets* of cells in some specified order and $\exists$ **RCF** proof procedures will be applied to see if every cell in a selected set of cells may be eliminated.

[7] The intended connotation is of a *stage* in a *theatre*.

1. *A* cell selection strategy *for selecting* subsets *of* C_i *for analysis (we call such a subset a "selection of cells"),*
2. *A* formula construction strategy *for constructing an* $\exists$ ***RCF*** *formula whose truth value will correspond to the relevance of a selection of cells (we call such a formula a "cell selection relevance formula"),*
3. *An* $\exists$ ***RCF*** *proof procedure used to (attempt to) decide the truth or falsity of a cell selection relevance formula.*

In the context of CAD construction, sample points will be eliminated when their corresponding cells are deemed to be irrelevant to the $\exists$ ***RCF*** *formula inducing the CAD. This removal might then result in a set of sample points for which the cell selection function behaves differently than it did initially. This motivates the* containment axiom *for covering width functions, so that these dynamics do not result in a non-terminating CAD-based decision algorithm employing the stage machinery. In what follows, let* $R_i = \{s \subset \mathbb{R}^i \mid |s| < \omega\}$.

1. *A* cell selection strategy *consists of two components:*
   (a) *A* covering width function $w : R_i \to \mathbb{N}$,
   (b) *A* cell selection function $\mathbb{S} : R_i \times \mathbb{N} \to R_i$ *obeying for all* $S_i \in R_i$ *and all* $j \in \{1, \ldots, w(S_i)\}$ *the* containment axiom*:* $\mathbb{S}(S_i, j) \subset S_i$.
2. *A* formula construction strategy *is a function* $\mathbb{F} : \mathcal{L}_{\exists OR} \times R_i \to \mathcal{L}_{\exists OR}$ *obeying certain* relevance judgment axioms*. To describe these axioms, we need the context of a fixed (but arbitrary)* $\exists$ ***RCF*** *formula and an associated* P_i*-invariant CAD of* $\mathbb{R}^i$*. Let* φ *be an* $\exists$ ***RCF*** *formula with polynomials* $P \subset \mathbb{Z}[x_1, \ldots, x_n]$ *and let* $P_n, \ldots, P_1$ *be a sequence of level-*$(n, \ldots, 1)$ *projection sets rooted in* P *(recall* $P_n = P$*). Let* $C_i = \{c_1, \ldots, c_m\}$ *be a* P_i*-invariant CAD of* $\mathbb{R}^i$ *with* S_i *a set of sample points drawn from a subset of the cells in* C_i*. If we are given a set of sample points* $\{\boldsymbol{s_{a_1}}, \ldots \boldsymbol{s_{a_v}}\} \subseteq S_i$*, then* $\triangle(\{\boldsymbol{s_{a_1}}, \ldots \boldsymbol{s_{a_v}}\})$ *will denote the set of cells from which the sample points* $\boldsymbol{s_{a_j}}$ *are drawn. Then, for each set of sample points* S_i *and each* $j \in \{1, \ldots, w(S_i)\}$ *the following* relevance judgment axioms *must hold:* $[(\boldsymbol{RCF} \models \neg\mathbb{F}(\varphi, \mathbb{S}(S_i, j))) \implies \mathcal{N}(\varphi, \mathbb{S}(S_i, j))]$*, and* $[(\boldsymbol{RCF} \models \mathbb{F}(\varphi, \mathbb{S}(S_i, j))) \implies \boldsymbol{RCF} \models \varphi]$*, where* $\mathcal{N}(\varphi, \{\boldsymbol{s_{a_1}}, \ldots \boldsymbol{s_{a_v}}\})$ *means that no child (at any ancestral depth, i.e., in a* P_{i+1}*-invariant CAD of* $\mathbb{R}^{i+1}$*, in a* P_{i+2}*-invariant CAD of* $\mathbb{R}^{i+2}, \ldots$*, in a* P_n*-invariant CAD of* $\mathbb{R}^n$*) of any cell in the set* $\Delta(\{\boldsymbol{s_{a_1}}, \ldots, \boldsymbol{s_{a_v}}\})$ *will satisfy* $QF(\varphi)$*.*
3. *An* $\exists$ ***RCF*** proof procedure *is a function*

$$\mathbb{P} : \mathcal{L}_{\exists OR} \to \left(\{\boldsymbol{true}, \boldsymbol{false}, \boldsymbol{unknown}\} \cup \bigcup_{j \in \mathbb{N}^+} \mathbb{R}^j \right)$$

obeying the soundness axioms*:* $((\mathbb{P}(\psi) = \boldsymbol{true}) \implies \boldsymbol{RCF} \models \psi)$, $((\mathbb{P}(\psi) = \boldsymbol{false}) \implies \boldsymbol{RCF} \models \neg\psi)$, $((\mathbb{P}(\psi) \in \mathbb{R}^j) \implies \boldsymbol{RCF} \models QF(\psi)[\mathbb{P}(\psi)])$ *for arbitrary* $\psi \in \mathcal{L}_{\exists OR}$ *and with* $QF(\psi)[\mathbb{P}(\psi)]$ *in the final axiom being the substitution of the* j*-vector* $\mathbb{P}(\psi)$ *(or an arbitrary extension of it to the dimension of the polynomials appearing in* ψ*) into* ψ*, resulting in a variable-free formula. In this case, we call* $\mathbb{P}(\psi)$ *a* witness *to the truth of* ψ*.*

We will want to have the freedom to give our AP-CAD algorithm a *sequence* of stages, one for each dimension $1, \ldots, n$. The intuition is as follows: Stages are introduced so that one can present a *strategy* to an underlying CAD decision algorithm which will prescribe a method for the algorithm to recognise when it can short-circuit certain expensive computations. If we can abandon a cell at a low dimension, for instance at the base phase or when beginning to lift over cells of $\mathbb{R}^2$, then this can potentially give us *hyper-exponential* savings down the line. Thus, it makes sense to arrange stages $\mathfrak{A}_1, \mathfrak{A}_2, \ldots \mathfrak{A}_n$ so that stage $\mathfrak{A}_1$ works *hardest* to make relevance judgments about cells. For if $\mathfrak{A}_1$ causes us to throw away cell $c_j \subset \mathbb{R}^1$, then this could lead to huge savings later. Then, $\mathfrak{A}_2$ might still work hard but a bit *less* hard, and so on. This collection of stages gives rise to the notion of an *n-theatre*. In what follows, let Θ be the set of all stages.

Definition 5 (Theatre). *An* n-theatre $\mathbb{T}$ *is a function* $\mathbb{T} : \{1, \ldots, n\} \to \Theta$.

Stage i in a theatre will be used to make judgments about cells in a P_i-invariant (partial) CAD of $\mathbb{R}^i$ (i.e., at level i). Let us describe a decision method we will use for deciding $\exists$ **RCF** sentences in the framework of AP-CAD.

Algorithm 1 (AP-CAD with Theatrical Lifting) *Suppose we are given an* $\exists$ ***RCF*** *sentence* φ *with polynomials* $P \subset \mathbb{Z}[x_1, \ldots, x_n]$*, and an* n*-theatre* $\mathbb{T}$*.*

1. ***Projection*** *As with standard P-CAD, compute a sequence of projection sets* $P_1, \ldots, P_n$*.*
2. ***Base*** *As with standard P-CAD, compute a* P_1*-invariant CAD of* $\mathbb{R}^1$*,* $C_1 = \{c_1, \ldots, c_{2m+1}\}$ *represented by sample points* $S_1 = \{\boldsymbol{s_1}, \ldots, \boldsymbol{s_{2m+1}}\}$*. Set the current dimension* $i := 1$*.*
3. ***Lifting*** *Let* $\mathbb{T}(i) = \mathfrak{A}_i = \langle\langle \mathbb{S}_i, w_i\rangle, \mathbb{F}_i, \mathbb{P}_i\rangle$ *be the stage for dimension* i*, and* S_i *the set of sample points for the* P_i*-invariant (partial) CAD of* $\mathbb{R}^i$ *over which we need to lift.*
   (a) *Let* $U := w_i(S_i)$ *and let* $j := 1$*.*
   (b) ***While*** $j \leq U$ ***do***
      i. *Let* $\{\boldsymbol{s_{a_1}}, \ldots, \boldsymbol{s_{a_v}}\} := \mathbb{S}_i(S_i, j)$*.*
      ii. *Let* $\chi := \mathbb{P}_i(\mathbb{F}_i(\{\boldsymbol{s_{a_1}}, \ldots, \boldsymbol{s_{a_v}}\}))$*.*
      iii. *If* $\chi =$ ***true****, then* ***return true****.*
      iv. *If* $\chi = \langle x_1, \ldots, x_w\rangle \in \mathbb{R}^w$ *for some* $w \leq n$*, then*
         A. *Fix an* n*-dim. extension of* χ*, e.g.,* $\boldsymbol{r} = \langle x_1, \ldots, x_w, \boldsymbol{0}\rangle \in \mathbb{R}^n$*.*
         B. *Evaluate* $QF(\psi)[\boldsymbol{r}]$ *and set* $R \in \{\boldsymbol{true}, \boldsymbol{false}\}$ *to this result.*
         C. *If* $R =$ ***true****, then* ***return r*** *as a witness to the truth of* φ*.*
         D. *If* $R =$ ***false****, then* ***return true***[8]*.*

[8] This is perhaps the one counter-intuitive part of the algorithm. Note, however, that this is actually correct: By the combination of the second *relevance judgment axiom* for $\mathbb{F}_i$ and the *soundness axioms* for $\mathbb{P}_i$, the fact that $\mathbf{RCF} \models \mathbb{F}_i(\mathbb{S}_i(S_i, j))$ means that φ is **true**. It's just that the witness $\mathbb{P}_i$ computed for the truth of $\mathbb{F}_i(\mathbb{S}_i(S_i, j))$ might fail to be a witness for φ. In this case, we simply know φ is true without knowing a witness for it.

v. *If* $\chi =$ ***false****, then set* $S_i' := S_i \setminus \{\boldsymbol{s_{a_1}}, \ldots, \boldsymbol{s_{a_v}}\}$*, else set* $S_i' := S_i$*.*
vi. *If* $S_i' = \emptyset$ *then* ***return false****.*
vii. *If* $S_i' = S_i$ *then set* $j := j + 1$*.*
viii. *If* $S_i' \subset S_i$ *then*
A. *Set* $S_i := S_i'$*.*
B. *Set* $U := w_i(S_i)$*.*
C. *Set* $j := 1$*.*

(c) *Now,* $S_i = \{\boldsymbol{t_1}, \ldots, \boldsymbol{t_u}\}$ *contains sample points corresponding to the cells we have not deemed to be irrelevant. We need to lift over them.*
i. *Let* $S_{i+1} := \emptyset$*.*
ii. ***For*** j ***from*** 1 ***to*** u ***do***
A. *Substitute the components of* $\boldsymbol{t_j}$ *in for the variables* $x_1, \ldots, x_i$ *in* P_{i+1} *to obtain a univariate family* $P_{i+1}[\boldsymbol{t_j}] \subset \mathbb{Z}[x_{i+1}]$*.*
B. *Compute a* $P_{i+1}[\boldsymbol{t_j}]$*-invariant CAD of* $\mathbb{R}^1$*, represented by sample points* K_j*.*
C. *Set* $S_{i+1} := S_{i+1} \cup K_j$*.*

(d) *Increase the current dimension by setting* $i := i + 1$*.*
(e) *If* $i = n$ *then lifting is done and we may proceed to the* evaluation phase*.*
(f) *If* $i < n$ *then we loop and begin the lifting process again, but now with the set of sample points* S_{i+1}*.*

4. ***Evaluation*** *Return the boolean value* $\bigvee_{\boldsymbol{r} \in S_n} QF(\varphi)[\boldsymbol{r}]$*.*

Theorem 2 *Algorithm 1 is a sound and complete* $\exists$ ***RCF*** *proof procedure.*

Proof. By induction on dimension. (See the extended version of this paper.)

4 Experimental Results

As an experiment (explicated in the extended version of this paper), we built a concrete AP-CAD theatre combining interval constraint reasoning with standard partial CAD [9]. As CAD performance is strongly dependent on the number of cells retained at each dimension, we compared this for three CAD variants: (i) CAD, (ii) standard P-CAD, and (iii) AP-CAD, w.r.t. an $\exists$ **RCF** sentence φ s.t.

$$QF(\varphi) = \left[\begin{array}{r} (x_1x_4 + x_2x_4 + x_3x_2 < 0) \wedge \ (x_2 > 0) \ \wedge \ (x_3 > 0) \ \wedge \ (x_4 > 0) \\ \wedge \ (x_3x_4 - x_4^2 + x_3^2 + 1 < 0) \end{array} \right].$$

As $QF(\varphi)$ involves only strict inequalities, we appeal to a theorem of McCallum allowing us to only consider full-dimensional cells (selecting rational sample points), and compare the methods w.r.t. this CAD variant [9]. We observe that the AP-CAD method retains less cells than the other methods. This is supported by experiments we have done with other $\exists$ **RCF** formulas. In all cases, the cost of AP-CAD theatre execution measured $< 0.01\%$ of the total CPU time, indicating that there is much positive impact to be made by using incomplete **RCF** proof procedures to enhance the performance of CAD-based decision methods. The cell retainment counts are as follows:

	CAD	P-CAD	AP-CAD
$\mathbb{Q}^1$	2	2	1
$\mathbb{Q}^2$	14	7	5
$\mathbb{Q}^3$	40	10	7
$\mathbb{Q}^4$	200	50	35

5 Conclusion

AP-CAD allows strategic algorithmic data to be used to "short-circuit" expensive computations during the *lifting phase* of a CAD-based decision algorithm. This provides a principled approach for utilising fast, sound but possibly incomplete $\exists$ **RCF** proof procedures to enhance a *complete* decision method without threatening its completeness. We see many ways this work might be extended. It would be very interesting to work out similar machinery to be used during the *projection phase* of P-CAD. For this work to bear serious practical fruit, many more AP-CAD stages should be constructed and experimented with heavily.

References

1. J. Avigad and H. Friedman. Combining Decision Procedures for the Reals. In *Logical Methods in Computer Science*, 2006.
2. S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry.* Springer-Verlag, Secaucus, NJ, USA, 2006.
3. G. E. Collins and H. Hong. Partial Cylindrical Algebraic Decomposition for Quantifier Elimination. *J.Sym.Comp.*, 12(3):299–328, 1991.
4. M. Daumas, D. Lester, and C. Muñoz. Verified Real Number Calculations: A Library for Interval Arithmetic. *IEEE Trans. Comp.*, 58(2):226–237, 2009.
5. J. H. Davenport and J. Heintz. Real Quantifier Elimination is Doubly Exponential. *J. Symb. Comput.*, 5:29–35, 1988.
6. S. Gao, M. Ganai, F. Ivancic, A. Gupta, S. Sankaranarayanan, and E. Clarke. Integrating ICP and LRA solvers for deciding nonlinear real arithmetic problems. In *FMCAD, 2010*, pages 81–89, 2010.
7. J. Harrison. Verifying Nonlinear Real Formulas via Sums of Squares. In *TPHOLs'07*, pages 102–118, Berlin, Heidelberg, 2007. Springer-Verlag.
8. H. Hong. Comparison of Several Decision Algorithms for the Existential Theory of the Reals. Technical report, RISC, 1991.
9. G. O. Passmore. *Combined Decision Procedures for Nonlinear Arithmetics, Real and Complex.* PhD thesis, University of Edinburgh, 2011.
10. G. O. Passmore and P. B. Jackson. Combined Decision Techniques for the Existential Theory of the Reals. In *Calculemus'09*, 2009.
11. F. Pfender and G. M. Ziegler. Kissing Numbers, Sphere Packings, and Some Unexpected Proofs. *Notices of the A.M.S.*, 51:873–883, 2004.
12. A. Platzer, J.-D. Quesel, and P. Rümmer. Real World Verification. In *CADE-22*, pages 485–501, Berlin, Heidelberg, 2009. Springer-Verlag.
13. A. Tarski. *A Decision Method for Elementary Algebra and Geometry.* RAND Corporation, 1948.
14. A. Tiwari. An Algebraic Approach for the Unsatisfiability of Nonlinear Constraints. In *CSL 2005*, volume 3634 of *LNCS*, pages 248–262. Springer, 2005.