Comput. complex. **7** (1998), 205 – 263
1016-3328/98/030205–59 $ 1.50+0.20/0

computational complexity

A QUANTIFIER ELIMINATION FOR THE THEORY OF p-ADIC NUMBERS

Lavinia Egidi

Abstract. This paper presents a detailed analysis of a quantifier elimination algorithm for the first order theory of p-adic numbers based on a p-adic analogue of the cylindrical algebraic decomposition. It is believed that this method should lead to an elementary upper bound for the theory. The present paper gives strong arguments against this conjecture and offers a basis for further speculation.
Key words. Complexity; theory of p-adic numbers; cylindrical algebraic decomposition.
Subject classifications. 68Q25

1. Introduction

The first order theory of p-adic numbers is, for each fixed prime p, the first order theory of henselian fields of characteristic 0 that have finite residue class field, a $\mathbf{Z}$-group as valuation group, and the valuation of p is one. A lower bound for the decision problem in this theory has been known for a long time; it is the same one that holds for Presburger arithmetic (since the latter is interpreted in the theory of p-adics numbers), i.e., doubly exponential alternating time with a linear number of alternations. As for the upper bound, the best algorithm known so far is due to Cohen [6], and it is primitive recursive but not elementary. It is a quantifier elimination process in which the formula analyzed is expanded in a dramatic (double exponential?) way each time a quantifier is eliminated.

The problem of a similar explosion was faced before, in the different context of the first order theory of the reals $Th(\mathbf{R})$. Consider, for instance, Cohen's real counterpart ([6]) of his own algorithm for $Th(\mathbf{Q}_p)$. Like the latter, it treats the quantified variables one at a time and suffers from exponential growth at each step of the recursion. The solution proposed in the real case by Collins [7] is a way of eliminating all the quantifiers at the same time through exploitation of geometrical properties of polynomials and their roots.

Since there are interesting analogies between the real field and the p-adic field $\mathbf{Q}_p$ (see [12]), it is reasonable to expect that an elementary algorithm could be obtained by using Collins' method on the p-adics. Macintyre [12] conjectured that the complexity of $Th(\mathbf{Q}_p)$ is the same as that of Presburger arithmetic (since the former interprets the latter), and he suggested [13] that a fast algorithm should be based on an analogue of Collins' quantifier elimination. Indeed, is by exploiting similar ideas, Brown [5] proved interesting complexity bounds about transfer principles involving $\mathbf{Q}_p$. Scowcroft and Van den Dries [14] point out that the essential algebra needed for tailoring the method to the p-adic case is provided by Denef's study [8] of a cylindrical algebraic decomposition for $\mathbf{Q}_p$, which points out the complexity aspect of the problem.

Nevertheless, note that the mentioned analogies between p-adics and reals cannot be pushed too far. A deep difference between the topological properties of the two fields complicates the p-adic setting significantly. The topology of $\mathbf{Q}_p$ is totally disconnected. Because of this no analogue of Sturm's theorem is known thus far. This has serious implications for the quantifier eliminations since they essentially come down to root isolations and related polynomial manipulations.

A first consequence of this fact is that at present one can't think of adapting to the theory of $\mathbf{Q}_p$ the algorithm due to Ben-Or, Kozen and Reif [4], which is an alternative, more efficient decision procedure for the theory of the reals. Collins' algorithm requires double exponential time. Ben-Or, Kozen and Reif have proven an exponential space upper bound, but their work makes use in an essential way of Sturm sequences and Sturm's theorem.

A second consequence is that the only technique available for treating roots of polynomials is quite cumbersome from a complexity point of view, as will be clarified later.

The present paper studies Collins' method, and uses Denef's ideas in addition. The whole analysis will be useful for a better understanding of the complexity of the theory of p-adic numbers. It leads to interesting conclusions that are meant to provoke some discussion on the subject.

It turns out that there is one critical point where the complexity of the algorithm explodes and, as mentioned, the uncontrolled growth of the complexity seems to depend on the way that the roots of polynomials are treated. Indeed this calls for a comparison with the real case for understanding what makes a good method for $\mathbf{R}$ ineffective on $\mathbf{Q}_p$. The point is discussed in the final section, but it is worth anticipating that a natural interpretation of the phenomenon leads us back to the lack of some analogue of Sturm's theorem on the p-adics.

This suggests an investigation of the depth of the differences between p-adic and real fields. It stirs curiosity on the inherent difficulty of the treatment of polynomials and their roots. However the question of whether the method could still lead to an elementary algorithm by some shortcut or *ad hoc* algebraic manipulation remains open.

A full understanding of the nature and complexity of the p-adic version of Collins' method is a good starting point for further pursuing of answers to the above questions.

This paper is an attempt to present the methodology clearly. The theoretical background on which the algorithm is based is described in detail, at the same time providing a bridge between the existential character of many proofs and a more constructive point of view.

The algorithm is organized in a modular way for understanding and ease of complexity analysis. Its structure is as close as possible to that of Collins' algorithm, which makes it easier to compare the two scenarios. To obtain an algorithm of this form, it was necessary to unwind the multiple recursions hidden behind Denef's clean exposition, taking care not to remain entangled in the many tiny threads.

The complexity analysis is exposed in two distinct, neatly separated parts, in order to make evident how far the method is efficient and what makes it non elementary in the end. This feature enhances the intended character of this paper as a basis for further research in the area.

In writing about my work I have been struggling for readability without compromising precision. The subject is itself so intricate at times that it hasn't been possible to avoid clumsiness. Yet I have tried to identify basic steps and essential concepts, and to build the paper around them. There are complex definitions that are intended to provide clean and intuitive tools for exposing the arguments. Fine detail is hidden in these structures, yet available to the interested reader.

The paper is essentially self-contained. Some proofs that are available elsewhere in the literature are here simply sketched in order to provide the basis for further analysis without overloading the paper. Only a few have been omitted altogether, but, in such cases, precise references have been provided. I believe that the absence of these proofs will not impair the comprehension of the material presented.

Section 2 gives the main notions about p-adic fields and the theory studied in the paper. In Sections 3 and 4, the state of the art is discussed in a progressive fashion leading to the description of the method used in this paper. All the theoretical background is given.

Subsection 4.1 is essentially an anticipation of the proof of correctness. It is necessary in order to introduce the reader in a smooth fashion to the algorithm. Subsections 4.2 and 4.3 are mainly concerned with remarks on notation but also stress some non obvious points.

Section 5 defines the support structures mentioned above. Then the algorithm is presented, in all its details (Section 6). As described here, it works for a prenex formula only, but it could be slightly modified to work also for a general formula using the same amount of space.

The proof of correctness (Section 7) should help put together all the tiles of the puzzle.

Section 8 gives a detailed analysis of the complexity. It follows the modular organization of the presentation of the algorithm and benefits from the support tools defined in the previous chapters. Where possible, the proofs follow the structure of the definitions of the objects being analyzed.

A concluding section attempts a brief comparison between the cylindric algebraic decomposition for the reals and its p-adic counterpart.

2. The p-adic numbers and the theory

Let p denote any fixed prime number. A map that satisfies the properties

1. $|x|_p \geq 0 \wedge (|x|_p = 0 \leftrightarrow x = 0)$
2. $|x \cdot y|_p = |x|_p \cdot |y|_p$
3. $|x + y|_p \leq \max(|x|_p, |y|_p)$ (ultrametric inequality)
4. $|p|_p = \frac{1}{p}$

is called *p-adic norm* (it is a non-Archimedean norm). The completion of the field $\mathbf{Q}$ of rational numbers with respect to $|\cdot|_p : \mathbf{Q} \to \{p^n\}_{n \in \mathbf{Z}} \bigcup \{0\}$ is the field of p-adic numbers $\mathbf{Q}_p$. Notice that this construction mimicks the construction of the reals from $\mathbf{Q}$. The p-adic norm over $\mathbf{Q}_p$ (also denoted $|\cdot|_p$) is defined in the obvious way from the p-adic norm over $\mathbf{Q}$ and takes values in the same range.

One can define the *valuation*, a map $v_p : \mathbf{Q}_p \to \mathbf{Z} \bigcup \{\infty\}$ logarithmically related to the p-adic norm on $\mathbf{Q}_p$. It satisfies the following properties:

1. $v_p x \leq \infty \wedge (v_p x = \infty \leftrightarrow x = 0)$
2. $v_p(x \cdot y) = v_p x + v_p y$
3. $v_p(x + y) \geq \min(v_p x, v_p y)$
4. $v_p p = 1$.

$\mathbf{Q}_p$ is said to be a *field with valuation*. $\mathbf{Z}$ is the *valuation group* of $\mathbf{Q}_p$.

The subdomain of $\mathbf{Q}_p$ of *p-adic integers*, i.e., p-adic numbers with nonnegative valuation, is denoted by $\mathbf{Z}_p$. The elements of $\mathbf{Z}$ are sometimes called *rational integers* to distinguish them explicitly from the p-adic integers.

The unique maximal ideal of $\mathbf{Z}_p$ is $\mathbf{M}_p = \{x \in \mathbf{Z}_p | v_p x > 0\}$. The quotient $\mathbf{Z}_p/\mathbf{M}_p$, the *residue class field* of $\mathbf{Q}_p$, is the field of p elements $\mathbf{F}_p$.

Each p-adic number is uniquely determined by its valuation and its *angular component*. For each $b \in \mathbf{Q}_p$

$$b = p^{v_p b} \cdot \mathrm{ac} b,$$

where $\mathrm{ac} : \mathbf{Q}_p \to \mathbf{Z}_p$ takes as argument a p-adic number and yields its angular component. Intuitively $ac(x) = x \cdot p^{-v_p x}$.

Any p-adic number b can be written in a unique way as

$$b = \sum_{i=s}^{\infty} b_i p^i \tag{2.1}$$

where s is any rational integer, and $b_i \in \{1, \ldots, p-1\}$. The expression on the right hand side of (2.1) is called the *p-adic expansion* of b. In terms of the p-adic expansion, the valuation of b is the least i, such that b_i is nonzero (in case $b \in \mathbf{Z}$, it amounts to saying that p^i is the largest power of p that divides b).

Given any pair of p-adic integers x and y, write $x \equiv y \bmod p^s$ for $v_p(x-y) \geq s$. This notation induces obvious concepts of *residue class* and *residue, modulo a power of p*.

An essential tool in p-adic arithmetic is Hensel's lemma. It gives a sufficient condition for the existence of a root.

Lemma 2.1 (Hensel's lemma). *Let $f(x)$ be a polynomial*

$$f(x) = a_0 + a_1 x + \ldots + a_d x^d$$

where $a_i \in \mathbf{Z}_p$. Let $f'(x)$ be

$$f'(x) = a_1 + 2a_2 x + \ldots + d a_d x^{d-1}.$$

Let $\beta \in \mathbf{Z}_p$. If $f(\beta) \equiv 0 \bmod p^{2e+1}$, and $f'(\beta) \not\equiv 0 \bmod p^{e+1}$ for some nonnegative integer e, then there exists $\xi \in \mathbf{Z}_p$ such that $f(\xi) = 0$ and $\xi \equiv \beta \bmod p^{e+1}$.

The proof can be found in [10] for the case $e = 1$, or, e.g., in [6].

A field with valuation in which the above lemma holds is called *henselian*.

By looking at a few more residue classes, Hensel's lemma can be used as a necessary and sufficient condition for the existence of roots:

LEMMA 2.2 (EXISTENCE OF ROOTS). *Let $f(x)$ and $f'(x)$ be as in Lemma 2.1. Assume that, for $x \in \mathbf{Z}_p$, $v_p f'(x) \leq e$. Then f has a root $\xi \in \mathbf{Z}_p$ if and only if there exists a residue modulo p^{2e+1}, β, that satisfies Hensel's lemma. The root will be such that $\xi \equiv \beta \bmod p^{2e+1}$.*

PROOF. The "if" part is a weak form of Hensel's Lemma. For the "only if" part, consider that since the coefficients of $f'(x)$ are clearly p-adic integers by the assumptions on $f(x)$, and the same holds for x by hypothesis, e must be nonnegative. Let $\xi \in \mathbf{Z}_p$ be a root of $f(x)$. Now $v_p f'(x) \leq e$, for $x \in \mathbf{Z}_p$ implies $v_p f'(\xi) < e + 1$; on the other hand, $v_p f(\xi) \geq 2e + 1$, since ξ is a zero of f. Therefore, any β such that $\beta \equiv \xi \bmod p^{2e+1}$ satisfies the hypotheses of Lemma 2.1. □

This brief introduction to $\mathbf{Q}_p$ shows that, for each prime p, the field of p-adic numbers is a model of the first order theory of henselian fields of characteristic 0, with finite residue class field, valuation group a $\mathbf{Z}$-group, and $v_p p = 1$. The theory is complete and recursively axiomatizable, therefore decidable [1]–[3]; its completeness enables us to reason about one of its models in order to derive results about the whole theory. In this light we refer to it as $Th(\mathbf{Q}_p)$. It can be expressed in a variety of languages, ranging from one with infinitely many sorts, in which the valuation is expressed by a symbol of the language (see [6]), to a language with only one sort.

The latter choice is possible because the valuation is definable in the pure field language:

$$\mathbf{Q}_p \models (\forall x)[v_p x \geq 0 \leftrightarrow (\exists y)(y^2 = 1 + px^2)] \quad if \quad p \neq 2$$
$$\mathbf{Q}_p \models (\forall x)[v_p x \geq 0 \leftrightarrow (\exists y)(y^3 = 1 + px^3)] \quad if \quad p \neq 3.$$

The axioms of $Th(\mathbf{Q}_p)$ are those for a field with valuation (field axioms and axioms describing properties of the valuation) together with those that state that the characteristic is 0, the valuation group is a $\mathbf{Z}$-group, the residue class field is finite, Hensel's lemma holds and $v_p p = 1$ (see [2]).

The theory admits quantifier elimination in a language that makes use of the *cross-section* [1]–[3]. The cross section is a map $\pi : \mathbf{Z} \bigcup \{\infty\} \to \mathbf{Q}_p$ acting as a right inverse of the valuation: $\pi(x) = p^x$. It is awkward since it introduces rather complicated definable sets.

On the other hand $Th(\mathbf{Q}_p)$ admits quantifier elimination in the pure field language as well, augmented with predicates P_n, such that $P_n(x)$ if and only if

x is an n-th power [11]. The axioms that give the proper meaning to the new predicates are therefore

$$(\forall x)[P_n(x) \leftrightarrow (\exists y)y^n = x].$$

I work in the language with one sortand predicates P_n. My choice is motivated by the observation that the study of complexity properties over a simpler language is more informative. Moreover, the length of a formula in the pure field language is polynomially related to the length of the equivalent formula in a two sorted language with a symbol for the valuation.

The logical language used is the minimal one, containing $\wedge$, $\neg$ and the quantifiers. (However, for expository reasons, I shall freely make use in the following of a broader language than the one chosen.)

It should be noted that a formula of the above language can be seen as a (quantified) boolean combination of atomic formulas, each stating that some polynomial with coefficients in $\mathbf{Z}$ is an n-th power, for some n. For a detailed introduction to p-adic numbers, see e.g., [10]; an interesting, broad survey is in [12]; more specifically, about quantifier elimination in valued fields, see [15].)

3. Collins' method for the reals

First, note that any atomic formula in the language of $Th(\mathbf{R})$ is a polynomial inequality, or, equivalently, it states that some polynomial is a square (in $\mathbf{R}$, $x \geq 0$ if and only if x is a square).

Collins' procedure partitions the space into a finite number of sets (*cells*), in each of which every polynomial that appears in the sentence to be decided has constant sign; then it chooses a *sample point* from each cell C of the partition todetermine the sign that each of the polynomials involved has in C. At this point each existential (resp. universal) quantifier can be replaced by a finite disjunction (resp. conjunction) of the polynomial inequalities evaluated at the sample points. In this way a block of quantifiers is eliminated in a single step.

A cell is a set defined recursively on the dimension of the space, bounded at each stage by continuous real-valued algebraic functions. Its crucial property is that the poynomials that appear in the sentence to be decided have constant sign. The decomposition of the space in cells can be done efficiently, as Collins shows, and, due to a nice geometry of the cells, the sample points can be chosen in a uniform and quick way. The geometrical properties of the cells are succintly described as a *cylindrical algebraic decomposition* (for short *c.a.d.*) given to the cell decomposition.

4. The method for $Th(\mathbf{Q}_p)$

The novelty of Collins' procedure is the idea of removing all the quantifiers in a single step. Otherwise Cohen's algorithms are not too different in spirit.

Cohen defines a cell in $\mathbf{Q}_p$ as a set of the form

$$\{x | x = x_0 + up^a, \text{ where } u \in \mathbf{Z}_p\}.$$

Then he shows "how to cover the p-adic numbers by cells in each of which a given polynomial behaves in a simple fashion" [6], p. 139. The "simple fashion" that Cohen refers to means essentially that the valuation of the polynomial can be related in each cell to the valuation of one of its monomials, if necessary after a change of variable has been performed. The following steps of the quantifier elimination are very intricate and depend on the language used (essentially a language with infinitely many sorts, although most of them finite; it also includes the cross-section).

Denef extends the notion of cell to more dimensions (here and in the following $\bar{x}$ stands for $(x_1, \ldots, x_r)$, and $\tilde{y}$ for $(y_1, \ldots, y_m)$).

Definition 4.1 (Cell). *A* cell *in $\mathbf{Q}_p^r \times \mathbf{Q}_p$ is a set of the form*

$$\{(\bar{x}, y) \in \mathbf{Q}_p^r \times \mathbf{Q}_p | \bar{x} \in C \wedge v_p a_1(\bar{x}) \square_1 v_p(y - \chi(\bar{x})) \square_2 v_p a_2(\bar{x})\}$$

where the $\square_i$ are either $\leq$, or $<$, or there is no condition at all, $\chi(\bar{x})$ and $a_i(\bar{x})$ are semialgebraic functions of $\bar{x}$, and C is a semialgebraic subset of $\mathbf{Q}_p^r$. The function $\chi(\bar{x})$ is called the center *of the cell.*

(Note that Cohen's definition of a cell could be rephrased as $\{x | v_p(x - x_0) \geq a\}$.)

A *semialgebraic* set is a boolean combination of sets of the form

$$\{\tilde{y} \in \mathbf{Q}_p^m | P_n(h(\tilde{y}))\},$$

where h is a polynomial (consistent with the notion of semialgebraic sets over the reals which are the sets definable via polynomial inequalities). Observe that sets of this form are the only sets that can be defined in the language chosen.

Semialgebraic (partial) functions preserve, in some sense, semialgebraic sets:

Definition 4.2 (Semialgebraic functions). *A function $f : \mathbf{Q}_p^r \to \mathbf{Q}_p$ is said to be* semialgebraic *if, for all semialgebraic sets, $S \subset \mathbf{Q}_p^{m+1}$, the set*

$$\{(\bar{x}, \tilde{y}) \in \mathbf{Q}_p^{r+m} | (f(\bar{x}), \tilde{y}) \in S\}$$

is also semialgebraic.

The same notion can also be expressed in a more intuitive way:

PROPOSITION 4.3 (CHARACTERIZATION). *A function $f : \mathbf{Q}_p^r \to \mathbf{Q}_p$ is semi algebraic if and only if for each polynomial $h : \mathbf{Q}_p^{m+1} \to \mathbf{Q}_p$ and rational integer n, there exist finitely many polynomials $h_i^* : \mathbf{Q}_p^{r+m} \to \mathbf{Q}_p$ and rational integers m_i, such that the set*

$$\{(\bar{x}, \tilde{y}) \in \mathbf{Q}_p^{r+m} | P_n(h(f(\bar{x}), \tilde{y}))\}$$

can be expressed as a boolean combination of the sets

$$\{(\bar{x}, \tilde{y}) \in \mathbf{Q}_p^{r+m} | P_{m_i}(h_i^*(\bar{x}, \tilde{y}))\}.$$

This result follows by combining explicitly the definition of semialgebraic functions with the notion of semialgebraic set. The characterization will be useful later; it amounts to saying that, although in general a semialgebraic function f is not a polynomial, yet any semialgebraic condition on it (i.e., a condition stating that a polynomial form in f is an n-th power for some n) can be translated into another involving only polynomials.

Denef shows that a decomposition of the $r+1$-dimensional space in the spirit of Collins' c.a.d. is definable.

THEOREM 4.4 (DENEF [8]). *Let $f_i(\bar{x}, y)$, for $i = 1, \ldots, m$, be polynomials in y with coefficients that are semialgebraic functions of $\bar{x}$. Let $n \in \mathbf{N}$, $n > 0$, be fixed. Then there exists a finite partition of $\mathbf{Q}_p^r \times \mathbf{Q}_p$ into cells A, such that each such cell A has a center $\chi(\bar{x})$ such that for all $(\bar{x}, y) \in A$, we have*

$$f_i(\bar{x}, t) = u_i(\bar{x}, t)^n h_i(\bar{x})(t - \chi(\bar{x}))^{\nu_i}, \text{ for } i = 1, \ldots, r,$$

with $v_p u_i(\bar{x}, y) = 0$, $h_i(\bar{x})$ a semi algebraic function of $\bar{x}$, and $\nu_i \in \mathbf{N}$.

The decomposition is built in a recursive fashion as in the real case. The "boundaries" of the cells are semialgebraic functions; the difference between this definition that of Collins' is nontrivial. The difficulty that arises is better seen by a reference to the characterization of semialgebraic functions. That characterization states that any semialgebraic condition on a semialgebraic function can be expressed using only polynomials and predicates P_n; it doesn't say anything about *how* the simpler expression is to be obtained. Since our target is a procedure, we need something constructive rather than existential.

An analysis of Denef's decomposition reveals that, starting out with a polynomial, only specific functions must be considered. I have called these functions *constructive* semialgebraic to stress the fact that a witness of their semialgebraic character can be computed.

Definition 4.5 (Constructive semialgebraic functions). *The set of constructive semialgebraic functions is the smallest set $\mathcal{C}$ of functions such that:*

1. *$+$, $-$, $\times$ and $\div$ belong to $\mathcal{C}$.*
2. *Let $g(\bar{x},t)$ be a polynomial in t whose coefficients are functions of $\bar{x}$ belonging to $\mathcal{C}$ and taking values in $\mathbf{Z}_p$; let D, a semi algebraic subset of $\mathbf{Q}_p^r$, and $e \in \mathbf{N}$ be such that, for all $\bar{x} \in D$ and $t \in \mathbf{Z}_p$,*

$$v_p g'(\bar{x},t) \leq e$$

(here and in the following I use the notation $g'(\bar{x},t)$ to denote $\frac{\partial}{\partial t}g(\bar{x},t)$). Let h be such that

$$v_p g(\bar{x},h) \geq 2e+1$$

for all $\bar{x} \in D$. Then the "root-function" $\xi : D \to \mathbf{Z}_p$ such that

$$g(\bar{x},\xi(\bar{x})) = 0 \quad \text{and} \quad v_p(\xi(\bar{x}) - h) \geq e+1$$

is in $\mathcal{C}$.

3. *Let $\psi(\bar{x}) : D \to \mathbf{Q}_p$ be in $\mathcal{C}$, where D is a semialgebraic subset of of $\mathbf{Q}_p^r$. Let $k \in \mathbf{N}$, $k \geq 2$; assume that $\psi(\bar{x}) \neq 0$ for $\bar{x} \in D$, that for some s $v_p\psi(\bar{x}) \equiv s \mod k$, and that for some ρ, $\psi(\bar{x}) = \rho\cdot$(nonzero N-th power), with $N = p^{2v_p k}(p-1)$. Let $\psi_1(\bar{x}) = \frac{\psi(\bar{x})}{\rho}p^{v_p\rho - s}$.*
Then the "θ-function" $\theta : D \to \mathbf{Q}_p$ such that, for all $\bar{x} \in D$,

$$\theta(\bar{x})^k = \psi_1(\bar{x}) \ \wedge \ P_{N_1}(\theta(\bar{x})),$$

with $N_1 = p^{v_p k}(p-1)$, belongs to $\mathcal{C}$.

4. *$\mathcal{C}$ is closed under composition.*

The class $\mathcal{C}$ is well-defined: the existence and uniqueness of root functions is guaranteed by Hensel's lemma; for θ functions, see [8], Lemma 2.4. It is moreover the required class:

Lemma 4.6 (Constructivity). *The functions in $\mathcal{C}$ are semi algebraic. Moreover, for each $f \in \mathcal{C}$, $f : \mathbf{Q}_p^r \to \mathbf{Q}_p$, there exists a procedure to compute, given any polynomial $h : \mathbf{Q}_p^{m+1} \to \mathbf{Q}_p$ and rational integer n, the polynomials h_i^* and the integers m_i of Proposition 4.3.*

The lemma is prove by actually showing the mentioned procedure, a lengthy and very complex one. It is the same as described in [8], Lemmas 2.3 and 2.4. Although the proof is not explicitly presented here, the definition of csa trees in Section 5 can be viewed as a quite extended sketch of it. For further details the reader is referred to the mentioned lemmas in [8].

The constructive semialgebraic functions are actually either polynomials or zeroes of polynomials.

With the new "boundary" functions, the p-adic cells have geometrical properties that are in some sense analogous to those of the cells defined by Collins [7], but still play a different role in the decision procedure. In each cell of the decomposition for the reals, each of the polynomials at issue has constant sign. In the p-adic setting, the corresponding condition is not automatically verified.

Saying that a polynomial has constant sign in a cell amounts (in the reals) to saying that it is a square at each point of the cell, or its opposite is. In other words, either the polynomial is in the same coset of the squares as 1, or it is in the same coset as -1. In the p-adic case, we must look in general at n-th powers, not only squares; it can be proved, however, that for each n, there is a finite number of different cosets of the n-th powers (Lemma 8.1). Requiring that a polynomial have constant sign is equivalent in the p-adic setting to the condition that the polynomial be in a fixed coset of the n-th powers. In the following, for any n, Λ_n denotes the (finite) set of representatives of n-th power cosets.

Denef's theorem ensures that the polynomials are in a fixed coset of the n-th powers in each cell of the p-adic c.a.d., *provided* that some auxiliary functions of the form $y - \chi(\bar{x})$ are also in a fixed coset of the n-th powers. Because of this, the quantifier elimination will use sets defined by (boolean combinations of) conditions of the form

$$y - \chi(\bar{x}) = \rho \cdot (\text{nonzero } n\text{-th power}) \tag{4.1}$$

$$v_p a_1(\bar{x}) \square_1 v_p(y - \chi(\bar{x})) \square_2 v_p a_2(\bar{x}), \tag{4.2}$$

where $\chi(\bar{x})$, $a_i(\bar{x})$ are constructive semialgebraic functions and $\square_i$ are as in Denef's definition of cells. Therefore, in the following the term *cells* will refer to sets of this form. Conditions of the form (4.1) will be called *power conditions*; conditions of the form (4.2) *valuation conditions*. The functions $a_1(\bar{x})$ and $a_2(\bar{x})$ in valuation conditions will be referred to as *boundary functions*.

The final issue is picking the sample points from each of the cells. For the purpose of choosing a sample point from each of these sets, it is first necessary to achieve a decomposition of the space in cells defined in a simpler way, namely

using just one power condition and one valuation condition. The procedure that performs the simplification is very straighforward but benefits from a quite tedious analysis of the possible cases that can arise, each to be treated separately. Then, the sample points can be chosen from each cell according to the following lemma:

LEMMA 4.7 (SAMPLING). *Let*

$$\begin{aligned} &y - \chi(\bar{x}) = \rho \cdot (\textit{nonzero } n\textit{-th power}),\\ &v_p a_1(\bar{x}) \leq v_p(y - \chi(\bar{x})) \leq v_p a_2(\bar{x}),\\ &\bar{x} \in C \end{aligned}$$

be the definition of the cell that one wants to sample from. Let $\bar{x}_0$ be the r coordinates of the sample point chosen in C. If $P_n(a_1(\bar{x})\rho^{-1}c)$ holds, then

$$y_0 = a_1(\bar{x}_0)c + \chi(\bar{x}_0)$$

is a good choice for the $(r+1)$ coordinate.

This lemma is based on the ideas in the quantifier elimination in the final section of [8]. The procedure is slightly more complex than this to take care also of cells in which the valuation is not bounded from below.

The geometry of the decomposition is analogous (modulo topological differences) to that of Collins' c.a.d., so that the choice of the sample points is just as straightforward. All the quantifiers can be removed at the same time.

THEOREM 4.8 (QUANTIFIER ELIMINATION). *Consider the sentence*

$$(Q_1 x_1) \dots (Q_r x_r)(Q_{r+1} x_{r+1})\phi(x_1, \dots, x_r, x_{r+1}), \tag{4.3}$$

where $Q_j \in \{\exists, \forall\}$, for $j = 1, \dots, r+1$, and ϕ is a quantifier free boolean combination of atomic formulae. There exists a finite partition of $\mathbf{Q}_p^{r+1}$ in cells C_i, such that, if $(\beta_{1,i}, \dots, \beta_{r+1,i})$ is the sample point chosen from C_i according to the Sampling Lemma, the sentence (4.3) is equivalent in the theory to

$$(B_1)_i \dots (B_r)_i (B_{r+1})_i \phi(\beta_{1,i}, \dots, \beta_{r,i}, \beta_{r+1,i}),$$

where i ranges over all the subscripts of the cells and, for $j = 1, \dots, r+1$, $(B_j) = \bigvee$ if $Q_j = \exists$, and $(B_j) = \bigwedge$ if $Q_j = \forall$.

4.1. The decomposition. At the $(r+1)$st stage, a decomposition of the $(r+1)$st dimension of the space with respect to a set of conditions has to be carried out. Let

$$f(\bar{x}, y) = \rho \cdot (\text{nonzero } n\text{-th power})$$

be one of these conditions. The polynomial appearing in it is viewed as an univariate polynomial in the $(r+1)$st variable with coefficients that are constructive semialgebraic functions of $\bar{x}$:

$$f(\bar{x}, y) = a_0(\bar{x}) + a_1(\bar{x})y + \ldots + a_d(\bar{x})y^d. \tag{4.4}$$

The coefficients are regarded as black boxes, except that in order for the decomposition to be properly carried out, it is assumed that at the preceding stage, conditions on the coefficients and certain other appropriate functions (that will be mentioned later as they appear) have been treated. Moreover, the space must have been already decomposed in cells in each of which the *partial derivative*

$$f'(\bar{x}, y) = a_1(\bar{x}) + 2a_2(\bar{x})y + \ldots + da_d(\bar{x})y^{d-1}$$

is in a fixed coset of the n-th powers. This implies an inner recursion on the degree of f, and the decomposition relative to f is built over a decomposition relative to f'. Since, while treating f', a change of variable might have been carried out (see later for the reason why this is so), the coefficients of f are in general not polynomials but more general constructive semialgebraic functions.

A second assumption on the decomposition of the r-th dimensional space is that it has been split in cells in each of which every coefficient of f either never vanishes or is always zero (notice that, for each $\gamma \in \mathbf{Q}_p$, $\gamma = 0$ if and only if $P_2(p\gamma^2)$). Afterwards the polynomial might not have monomials of each degree in y, but it is important that one takes here (4.4) as a general form for the polynomial. It must be stressed, though, that the monomials which do appear never vanish.

Recall that the goal is writing f as

$$f(\bar{x}, y) = u(\bar{x}, y)^n h(\bar{x})(y - \chi(\bar{x}))^\nu, \tag{4.5}$$

where $v_p u(\bar{x}, y) = 0$, $h(\bar{x})$ is a constructive semialgebraic function and ν is a positive integer. Therefore if $h(\bar{x})$ and $y - \chi(\bar{x})$ are n-th powers, so is f. We are assuming that the required functions of $\bar{x}$ have already been dealt with, and therefore the above condition on $h(\bar{x})$ is met. The one on $y - \chi(\bar{x})$ is one of the conditions defining the $(r+1)$st dimension of the cell.

A fact used times and again in the process of writing f in the form (4.5) is:

LEMMA 4.9 (n-TH POWER RESIDUE). *For any $n \in \mathbf{N}$, there exists a rational integer λ for which any p-adic integer u such that $u \equiv 1 \pmod{p^\lambda}$ is an n-th power. Namely, if $v(n) = m$, then $\lambda = 2m + 1$.*

The proof is obtained by direct inspection of the p-adic expansion of an n-th power. The proof holds only for p-adic integers because it is based on Hensel's lemma. The constant λ defined will appear very often in the following:

NOTATION 4.10 (λ). *Let $\lambda_\mu = 2v_p\mu + 1$ for each subscript μ. If no subscript appears, n is intended: $\lambda = \lambda_n$.*

If f has coefficients that take p-adic integer values, y has nonnegative valuation, and the valuation of f is bounded in a cell, then one can split the space so that f's coefficients have fixed residues modulo λ in each cell, as y in each cell. This implies that the same holds for f, and so by Lemma 4.9, f is in a fixed coset of the n-th powers.

Notice for instance that in the cells in which there is an i_0 such that

$$v_p(a_{i_0}(\bar{x})y^{i_0}) \leq v_p(a_i(\bar{x})y^i) - \lambda \quad \text{for all} \quad i \neq i_0, \tag{4.6}$$

f can be written as $f(\bar{x}, y) = u(\bar{x}, y)^n a_{i_0}(\bar{x})y^{i_0}$ since $\frac{f(\bar{x},y)}{a_{i_0}(\bar{x})y^{i_0}} \equiv 1 \mod p^\lambda$.

But f's coefficients and variable are not in general p-adic integers. The solution is to factor f into a polynomial that has the properties just mentioned times some function of $\bar{x}$.

Suppose that in the cell C, y is such that $v_p y = v_p\theta(\bar{x})$, and i_0 is such that $v_p(a_{i_0}(\bar{x})y^{i_0}) < v_p(a_i(\bar{x})y^i)$ for all $i \neq i_0$. Then

$$g(\bar{x}, u) = \frac{f(\bar{x}, y)}{a_{i_0}(\bar{x})\theta(\bar{x})^{i_0}} \quad \text{with} \quad u = \frac{y}{\theta(\bar{x})} \tag{4.7}$$

has coefficients and variable u in $\mathbf{Z}_p$ in the cell C. The function g will be therefore referred to as the *integral polynomial* relative to f.

Since

$$v_p g(\bar{x}, u) = v_p f(\bar{x}, y) - v_p(a_{i_0}(\bar{x})\theta(\bar{x})^{i_0}),$$

if the difference

$$v_p f(\bar{x}, y) - \min_i\{v_p(a_i(\bar{x})y^i)\} \tag{4.8}$$

is bounded, so is the valuation of g, and it is possible to split the space into a finite number of cells in each of which g is in a fixed coset of the n-th powers.

The first step is therefore splitting the space in cells into such a way that in each of them, a function like the $\theta(\bar{x})$ above can be defined, and the difference (4.8) is bounded.

The difference (4.8) is not bounded in cells in which f has a nonzero root. By a basic fact of p-adic arithmetic, f can have a root only in those cells in which $v_p y$ is such that two monomials have the same valuation. Applying the properties of the valuation, the resulting condition on y is

$$v_p y = \frac{1}{i-j} v_p \frac{a_j(\bar{x})}{a_i(\bar{x})} \tag{4.9}$$

for some $i \neq j$.

For each pair (i,j) $(i,j \in \{0,\ldots,d\}, |i-j| \geq 2)$, for each $s \in \{0,\ldots,i-j-1\}$ and $\rho \in \Lambda_{N_2}$ for $N_2 = (i-j)p^{v_p(i-j)}(p-1)$, define a function $\theta_{ij}(\bar{x})$ as in Item 3 of Definition 4.5, with $\psi(\bar{x}) = \frac{a_j(\bar{x})}{a_i(\bar{x})}$ and $k = i - j$[1].

For the pairs (i,j) such that $|i-j| = 1$, let $\theta_{ij}(\bar{x}) = \frac{a_j(\bar{x})}{a_i(\bar{x})}$. By Item 1 of Definition 4.5, this is a semialgebraic function if $a_i(\bar{x})$ and $a_j(\bar{x})$ are. For each pair (i,j), $v_p\theta_{ij}(\bar{x})$ equals the right hand side of (4.9)—see [8], Lemma 2.4 for the proof. Notice that for each pair (i,j), $\theta_{ij} = \theta_{ji}$.

Now for each (i,j), such that $j > i$, and for each θ_{ij} defined with $s = 0$, the cells

$$V^{(ij)} = \left\{ (\bar{x},y) | (\bar{x},y) \in D \wedge v_p y = v_p\theta_{ij}(\bar{x}) \wedge \right.$$
$$\left. \bigwedge_{h<i} v_p y \leq v_p\theta_{ih}(\bar{x}) \ \wedge \ \bigwedge_{h>i} v_p y \geq v_p\theta_{ih}(\bar{x}) \right\}$$

might contain a root.

If the $(r+1)$th dimension of the space is split according to the valuation conditions described in each item of the following enumeration, f can be written in the form (4.5) in each of the resulting sets, as discussed in each case below.

1. For all $i = 1,\ldots,d$, in the cells

$$A_\lambda^{(i_0)} = \left\{ (\bar{x},y) | (\bar{x},y) \in D \wedge \right.$$
$$\left. \bigwedge_{j<i_0} v_p\theta_{i_0 j}(\bar{x}) - v_p y \geq \left\lceil \frac{\lambda}{i_0 - j} \right\rceil \ \wedge \ \bigwedge_{j>i_0} v_p\theta_{i_0 j}(\bar{x}) - v_p y \leq \left\lceil \frac{\lambda}{i_0 - j} \right\rceil \right\}$$

[1]The notation $\theta_{ij}(\bar{x})$ hides the dependence of the definition from s and ρ. But a more precise notation would have been too heavy.

i_0 is as in Example (4.6); the form (4.5) for f is soon attained, as shown in that example. Here $P_n(a_{i_0}(\bar{x})) \ \wedge \ P_n(y)$ implies $P_n(f(\bar{x}))$.

2. For $i_0 = 0, \ldots, d$, $j = 0, \ldots, d$, $j \neq i$, $s = 1, \ldots, \lambda - 1$, in the cells

$$A_s^{(i_0 j)} = \left\{ (\bar{x}, y) | (\bar{x}, y) \in D \wedge v_p y = v_p \theta_{i_0 j}(\bar{x}) - \left\lceil \frac{s}{i_0 - j} \right\rceil \wedge \right.$$
$$\left. \bigwedge_{h<i_0} v_p y \leq v_p \theta_{i_0 h}(\bar{x}) - \left\lceil \frac{s}{i_0 - h} \right\rceil \ \wedge \bigwedge_{h>i_0} v_p y \geq v_p \theta_{i_0 h}(\bar{x}) - \left\lceil \frac{s}{i_0 - h} \right\rceil \right\}$$

$a_{i_0}(\bar{x}) y^{i_0}$ is the (unique) monomial of minimum order.

In each of these cells, define the integral polynomial g relative to f as in (4.7) above (replacing $\theta(\bar{x})$ by $\frac{\theta_{i_0 j}(\bar{x})}{p^s}$). Its coefficients $b_i(\bar{x})$ and its variable u have nonnegative valuation in the cells. The valuation of g itself is null. Thus in the cells

$$C_{\{h_1, \ldots, h_{d+1}\}} = \{(\bar{x}, y) | \bigwedge_{0 \leq i \leq d} b_i(\bar{x}) \equiv h_i \bmod p^\lambda \wedge u \equiv h_{d+1} \bmod p^\lambda\} \quad (4.10)$$

where $h_i \in \{0, \ldots, p^\lambda - 1\}$, $g(\bar{x}, u)$ has a constant power residue, and can therefore be written as $bv(\bar{x}, u)^n$ for some integer b and function v having valuation 0. Therefore

$$f(\bar{x}, y) = bv(\bar{x}, u)^n a_{i_0}(\bar{x}) \left(\frac{\theta_{i_0 j}(\bar{x})}{p^s} \right)^{i_0}.$$

Notice that the condition defining the space decomposition in cells of the form (4.10) is

$$\bigwedge_{0 \leq i \leq d} b_i(\bar{x}) \quad \text{has constant residue } \bmod p^\lambda \wedge$$
$$u \quad \text{has constant residue } \bmod p^\lambda. \quad (4.11)$$

3. As mentioned, in the cells $V^{(ij)}$ in which $v_p y = v_p \theta_{ij}(\bar{x})$, there might be a root.

Define here the integral polynomial relative to f, $g_{f,d}$. Here $\theta_{ij}(\bar{x})$ must replace $\theta(\bar{x})$ in the definition of the form (4.7). (The subscripts indicate that $g_{f,d}$ refers to f of degree d; they will soon be useful because it will

be necessary to distinguish this function from the integral polynomial relative to f'.)

There exists a constant e_d (which depends on the degree of f) such that

$$v_p g'_{f,d}(\bar{x}, u) \leq e_d. \tag{4.12}$$

The existence of this constant is due to the fact that

$$g'_{f,d}(\bar{x}, u) = \frac{f'(\bar{x}, y)}{a_{i_0}(\bar{x})\theta_{ij}(\bar{x})^{i_0-1}},$$

and it is assumed that a c.a.d. with respect to f' has been carried out.

Indeed, writing

$$g'_{f,d}(\bar{x}, u) = b_1(\bar{x}) + 2b_2(\bar{x})u + \ldots + db_d(\bar{x})u^{d-1},$$

the difference $v_p g'_{f,d}(\bar{x}, u) - \min_i v_p(ib_i(\bar{x})u^{i-1})$ equals

$$v_p f'(\bar{x}, y) - \min_i v_p(ia_i(\bar{x})y^{i-1})$$

which, by the induction hypothesis is bounded by a constant, say CONST. Since

$$\min_i v_p(ib_i(\bar{x})u^{i-1}) \leq v_p(i_0 b_{i_0}(\bar{x})u^{i_0-1}) = v_p i_0.$$

and $v_p i_0 < \infty$, $v_p g'_{f,d}(\bar{x}, u) \leq \max_{0<i\leq d} v_p i + \text{CONST}$. So it remains to determine what CONST is.

The algorithm is further decomposing a cell of the c.a.d. for f'. If no two monomials of f' have the same valuation in the cell, then CONST $= 0$. This is the case also in a cell where f' had a root since in this case a change of variable has been performed. Otherwise, consider $g_{f,d-1}(\bar{x}, u)$ as the integral polynomial relative to f' (which has degree $d-1$): $v_p g_{f,d-1}(\bar{x}, u) \leq 2e_{d-1}$ in this cell. Since

$$v_p g_{f,d-1}(\bar{x}, u) = v_p f'(\bar{x}, y) - \min_i v_p(ia_i(\bar{x})y^{i-1})$$

in this case CONST $= 2e_{d-1}$.

Consider the cells

$$V_h^{(ij)} = \{(\bar{x}, y))|(\bar{x}, y) \in V^{(ij)} \wedge u \equiv h \bmod p^{2e_d+1}\}.$$

(Cells with these property are defined via the request that u have constant residue mod p^{2e_d+1}.) By Hensel's Lemma, $g_{f,d}$ (and therefore f) has a unique root only in the cell (if any) $V_{h_0}^{(ij)}$, such that $v_p g_{f,d}(\bar{x}, h_0) \geq 2e_d+1$.

In the cells $V_h^{(ij)}$ where there is no root (and therefore $v_p g_{f,d}(\bar{x}, h) \leq 2e_d$), the situation is exactly the same as in the cells $A_s^{(i_0 j)}$ in Item 2 above. Assuming that in the cells of the r-dimensional decompositions each coefficient $b_i(\bar{x})$ of g has constant residue mod $p^{2e_d+\lambda}$, and adding

$$u \text{ has constant residue } \mod p^{2e_d+\lambda}$$

to the conditions defining the $(r+1)$st dimension, f can be written in the form (4.5). Here $P_n(a_i(\bar{x})) \ \wedge \ P_n(\theta_{ij}(\bar{x}))$ implies $P_n(f(\bar{x}))$.

4. The root detected in the cell $V_{h_0}^{(ij)}$ is a function $\xi(\bar{x})$ such that

$$(\bar{x}, \xi(\bar{x})\theta_{ij}(\bar{x})) \in V_{h_0}^{(ij)} \wedge g_{f,d}(\bar{x}, \xi(\bar{x})) = 0.$$

The polynomial in $u - \xi(\bar{x})$ with semialgebraic coefficients $c_i(\bar{x})$ obtained from $g_{f,d}$ via a change of variable

$$u \to u - \xi(\bar{x}),$$

has no root in the cell. The subcells of $V_{h_0}^{(ij)}$

$$W_\lambda^{(ij)} = \{(\bar{x}, y) | (\bar{x}, y) \in V_{h_0}^{(ij)} \wedge v_p(u - \xi(\bar{x})) \geq e_d + \lambda\}$$

and, for $s = 0, \ldots, \lambda - 1$:

$$W_s^{(ij)} = \{(\bar{x}, y) | (\bar{x}, y) \in V_{h_0}^{(ij)} \wedge v_p(u - \xi(\bar{x})) = e_d + s\}$$

play the analogous role as the cells $A_\lambda^{(i_0)}$ and $A_s^{(i_0 j)}$ discussed in Items 1 and 2 above, except that here the monomial of minimal valuation is always $c_1(\bar{x})(u - \xi(\bar{x}))$. The range of s is motivated by the fact that $v_p(u - \xi(\bar{x})) \geq e_d + 1$ in the cell; the bound in the cells $W_\lambda^{(ij)}$ is due to the fact that $v_p c_1(\bar{x}) - v_p c_j(\bar{x}) \leq e$. (See [8] for more details.) In the cells $W_\lambda^{(ij)}$, $P_n(a_i(\bar{x})) \ \wedge \ P_n(\theta_{ij}(\bar{x})) \ \wedge P_n(c_1(\bar{x})) \ \wedge P_n(u - \xi(\bar{x}))$ implies $P_n(f(\bar{x}))$.

The condition analogous to the (4.11) above is

$$\bigwedge_{0 \leq i \leq d} \left(\frac{c_i(\bar{x}) p^{(e_d+s)(i-1)}}{c_1(\bar{x})}\right) \quad \text{has constant residue } \mod p^\lambda \wedge$$

$$\left(\frac{u - \xi(\bar{x})}{p^{e_d+s}}\right) \quad \text{has constant residue } \mod p^\lambda.$$

This condition, along with $P_n(a_i(\bar{x})) \wedge P_n(\theta_{ij}(\bar{x})) \wedge P_n(c_1(\bar{x}))$ implies $P_n(f(\bar{x}))$ in the cells $W_s^{(ij)}$.

4.2. A word on the notation. A brief aside on the notation used is necessary here. In order to keep a certain degree of readability, the notation cannot be very precise. For instance a θ-function should be identified by mentioning at least four parameters (i, j, s and ρ). In fact when it becomes necessary to refer at the same time to θ-functions relative to different polynomials, or to partial derivatives of different order of the same polynomial, yet more subscripts would be necessary. This is obviously impractical.

The policy adopted here follows two main lines: to start with some parameters are never mentioned in subscripts because their presence would not add any useful information (cf. s and ρ in the θ-functions).

The second choice is less orthodox. Two different sets of notations are used both for θ-functions and for the integral polynomials g (that are strictly related to θ-functions). When it is necessary to distinguish which coefficients of the same polynomial are involved in the definitions, then the subscript ij appears (as has been the case so far for θ-functions θ_{ij}). When otherwise different levels of the inner recursion are discussed, then a polynomial f is specified to identify which is the polynomial at issue, and an integer J denotes the level of the inner recursion that is focused (cf. $g_{f,d}$ and $g_{f,d-1}$ in Item 3 above). A comma stands between the two subscripts in the second case to lessen the syntactical chaos.

Hopefully the use that is made of the notation is more intuitive than this explanation. The second kind of notation mentioned is discussed further in the next subsection.

4.3. Cell definitions. The decomposition process just described leads to defining the roots of f via subsequent θ-functions and root functions. A generic root of f will thus have the form

$$\Xi_{f,\ell}(\bar{x}) = \sum_{i\in\{1,\dots,\ell\}} \xi_{f,i}(\bar{x})\theta_{f,i}(\bar{x}).$$

It will be called in the following the *generalized root function*.

The notation introduced is meant to stress that $\Xi_{f,\ell}$ is a root of the s-th partial derivative with respect to y of $f(\bar{x}, y)$, with $s \geq d-\ell$ (and therefore the degree of $f^{(s)}$ is at most ℓ). Notice that $\theta_{f,i}(\bar{x})$ is meant to be a θ-function of $f^{(d-i)}$ and $\xi_{f,i}(\bar{x})$ the corresponding root function.

For uniformity it will be convenient to write $\Xi_{f,0}$ to denote the identically zero function (i.e. $y - \Xi_{f,0}$ is y itself).

Although this notation is ambiguous, because there are many θ-functions and root functions at each stage, it is still useful in order to give a more precise description of the general form that the cells definitions can have (cf. Section 4, Equations (4.1) and (4.2)).

Namely, the $(r+1)$st dimension of a cell of the decomposition is defined via boolean combinations of conditions of the form

$$v_p\theta_{f,\ell_1}(\bar{x}) + \textsc{Const}_1 \Box_1 v_p(y - \Xi_{f,\ell}(\bar{x})) \Box_2 v_p\theta_{f,\ell_2}(\bar{x}) + \textsc{Const}_2$$
$$y - \Xi_{f,\ell}(\bar{x}) = \rho \cdot (\text{nonzero } n\text{-th power})$$

where $\ell \leq \ell_1, \ell_2 \leq d$, the boxes stand for either $<$ or $\leq$ or no condition at all, and $\textsc{Const}_1$ and $\textsc{Const}_2$ are constants.

5. Support structures

In the description of the decomposition process, in Section 4.1, times and again conditions are referred to which have to be satisfied in each cell of the decomposition of $\mathbf{Q}_p^r$ in order to make the decomposition of the new dimension possible. The purpose of the following definition is to group all those conditions in a set. This will be clearer through the proof of correctness (Section 7).

To make many of the formulae to follow more readable, a symbol for the order of the group of units of $\mathbf{Z}_p/p^\alpha$ ($\alpha \in \mathbf{Z}$) is introduced here. Because of the fact that this will be proved in Lemma 7.1, this quantity will be often useful.

Notation 5.1 (m_α). *Let* $m_\alpha = p^{\alpha-1}(p-1)$.

The coefficients of a polynomial $f(\bar{x}, y)$ after a change of variable $y \to y - \Xi_{f,\ell}(\bar{x})$ are $f^{(i)}(\bar{x}, \Xi_{f,\ell}(\bar{x}))$. Notice that, if $f(\bar{x}, y) = a_0(\bar{x}) + \ldots + a_d(\bar{x})y^d$, $f^{(i)}(\bar{x}, \Xi_{f,0}(\bar{x})) = a_i(\bar{x})$, by the conventional meaning given to $\Xi_{f,0}$ in Subsection 4.3.

Definition 5.2 (Support Set). *For any condition*

$$f(\bar{x}, y) = \rho \cdot (\textit{nonzero } n\textit{-th power})$$

define its support set *as the set containing the following conditions:*

- *the conditions* $f^{(i)}(\bar{x}, \Xi_{f,\ell}(\bar{x}))^2 = \rho \cdot (\textit{nonzero square})$ *for each* $i, \ell = 0, \ldots, d$ *and for each* $\Xi_{f,\ell}$*;*

- *the conditions* $f^{(i)}(\bar{x}, \Xi_{f,\ell}(\bar{x})) = \rho \cdot (\textit{nonzero } N\textit{-th power})$ *for each* $i, \ell = 0, \ldots, d$, *for each* $\Xi_{f,\ell}$, *and with* $N = lcm_{h=1,\ldots,d}(hm_{2v_ph+1}, hm_{2e_d+\lambda}, hn)$;

- *the conditions*

$$(h-k)v_p\frac{f^{(j)}(\bar{x}, \Xi_{f,\ell}(\bar{x}))}{f^{(i)}(\bar{x}, \Xi_{f,\ell}(\bar{x})))} \leq (i-j)v_p\frac{f^{(k)}(\bar{x}, \Xi_{f,\ell}(\bar{x}))}{f^{(h)}(\bar{x}, \Xi_{f,\ell}(\bar{x})))}$$

  for each $i, j, h, k, \ell = 0, \ldots, d$, *with* $i > j$ *and* $h > k$, *and for each* $\Xi_{f,\ell}$;

- *the conditions*

$$(h-j)v_p\frac{f^{(i)}(\bar{x}, \Xi_{f,\ell}(\bar{x}))}{f^{(h)}(\bar{x}, \Xi_{f,\ell}(\bar{x}))} \leq (h-i)v_p\frac{f^{(j)}(\bar{x}, \Xi_{f,\ell}(\bar{x}))}{f^{(h)}(\bar{x}, \Xi_{f,\ell}(\bar{x})))} + s$$

  for each $i, j, h, \ell = 0, \ldots, d$, *with* $h > j$, *for each* $\Xi_{f,\ell}$ *and for* $s \in \{\lambda, 2e_d + \lambda\}$;

- *the conditions*

$$(j-h)\frac{v_pf^{(i)}(\bar{x}, \Xi_{f,\ell}(\bar{x}))}{v_pf'(\bar{x}, \Xi_{f,\ell}(\bar{x}))} + (j-h)(i-1)(e_d+s) \leq$$
$$\leq (i-1)v_p\frac{f^{(h)}(\bar{x}, \Xi_{f,\ell}(\bar{x}))}{f^{(j)}(\bar{x}, \Xi_{f,\ell}(\bar{x}))} + (j-h)\lambda$$

  for each $i, j, h, \ell = 0, \ldots, d$, *with* $j > h$, *and for each* $\Xi_{f,\ell}$.

Note that the last three conditions are about root functions, since the numerators and denominators of the fractions are coefficients of some polynomials obtained from f or its derivatives by a change of variable.

The support set contains the mentioned conditions for *all definable* root functions: thus functions defined from a $\psi_1(\bar{x})$ of the form

$$\psi_1(\bar{x}) = \frac{\psi(\bar{x})}{\rho}p^{v_p\rho-s}$$

for each $s \in \{0, \ldots, i-j-1\}$ and $\rho \in \Lambda_{N_2}$ for $N_2 = (i-j)p^{v_p(i-j)}(p-1)$ will be taken into account.

The definition of a complex structure (referred to as *csa tree*) is needed in order to give a clean exposition of the algorithm. Its definition is motivated by the fact that the witnesses of semi algebraicity of a condition labelling the root

of a csa tree are built following down the branches of the tree. The witnesses are the purely polynomial semi algebraic conditions to which any semi algebraic condition on a constructive semi algebraic function can be reduced, and they label the leaves of the csa tree (cf. Lemma 7.7). Therefore whenever a semi algebraic condition which is not purely polynomial is at issue, a csa tree provides the constructive means to define the purely polynomial conditions that are to replace it.

The definition is modelled on the proofs of Lemmas 2.3 and 2.4 of [8]. Whereas in those proofs the centers and boundary functions are mentioned in a generic way as semi algebraic functions, here θ-functions and generalized root functions are used instead (by Subsections 4.1 and 4.3). Those proofs outline a recursive procedure to build out of a generic semi algebraic condition a set of purely equivalent polynomial conditions. A csa tree describes one execution of the procedure on the condition that labels the root; each internal node represents one recursive call of the procedure and is labelled by the input condition. The leaves are labelled by the output. It is in this procedure that the complexity grows beyond control, as will be made clear later.

A few comments on the approach that the procedure takes should improve the understanding of the definition of csa trees. The most intricated part is the proof of semi algebraicity of root and θ-functions. In both cases the idea is to replace the function at issue with a place holder variable in the polynomial condition in which it appears and to carry out the c.a.d. for the resulting polynomial condition to obtain cell definitions. At this point, after having substituted back the function for the place holder variable, different cases must be considered as distinguished by the valuation and residue (modulo some power of p) of the functions involved. In each case, the conditions reduce to problems which are analogous to the initial one, but involve polynomials of a smaller degree (after Euclidean division).

It might be noticed that the procedure can follow one of several strategies. Since it is irrelevant for our purposes which strategy is chosen, one is tacitly picked without further comment.

In the following, the functions Ξ are the generalized root functions defined in Section 4.3. The function f_ℓ is such that $f_\ell(\bar{x}, \xi_{f,\ell}(\bar{x})\theta_{f,\ell}(\bar{x})) = 0$; it is obtained from f via the change of variable $y \to \Xi_{f,\ell-1}$ (this implies that $f_1 = f$). The functions θ and ψ, and the constant k are as in the definition of θ-functions (Def. 4.5, Item 3). The variable $\tilde{z}$ is meant to stand for an m-tuple of semi algebraic functions.

Definition 5.3 (csa tree). *A* csa tree *is a labelled tree recursively defined as follows:*

1. *Each node labelled by*

$$h(\bar{x}, \tilde{z}, \gamma(\bar{x})) = \rho \cdot (\textit{nonzero } n\textit{-th power})$$

for some polynomial h of degree d_h, and semi algebraic function γ which is the root of a polynomial f of degree $d_f < d_h$, has a child labelled by

$$r(\bar{x}, \tilde{z}, \gamma(\bar{x})) = \rho \cdot (\textit{nonzero } n\textit{-th power})$$

where $r(\bar{x}, \tilde{z}, t)$ is the remainder of Euclidean division of h by f.

2. *Each node labelled by*

$$h\left(\bar{x}, \tilde{z}, \frac{f_1(\bar{x})}{f_2(\bar{x})}\right) = \rho \cdot (\textit{nonzero } n\textit{-th power})$$

for some $f_1, f_2 \in \mathcal{C}$ and polynomial h of degree d_h, has a child labelled by

$$f_2(\bar{x})^{\nu} h\left(\bar{x}, \tilde{z}, \frac{f_1(\bar{x})}{f_2(\bar{x})}\right) = \rho \cdot (\textit{nonzero } n\textit{-th power})$$

where $\nu = \min\{m \geq d_h \textit{ s.t. } n|m\}$.

3. *Each node labelled by*

$$h(\bar{x}, \tilde{z}, \Xi_{f,\ell}(\bar{x})) = \rho \cdot (\textit{nonzero } n\textit{-th power}) \tag{5.1}$$

for some polynomial h, of degree d_h, smaller than the degree of the polynomial f_ℓ null at $\xi_{f,\ell}(\bar{x})\theta_{f,\ell}(\bar{x})$, has children labelled by the conditions in the support set of condition (5.1), plus children labelled by

$$v_p(\Xi_{f,\ell}(\bar{x}) - \Xi_{h,s}(\bar{x}, \tilde{z})) \quad \leq \quad v_p\theta_{h,t}(\bar{x}, \tilde{z}) + i$$

and children labelled by

$$\Xi_{f,\ell}(\bar{x}) - \Xi_{h,s}(\bar{x}, \tilde{z}) \quad = \quad \rho \cdot (\textit{nonzero } \mu\textit{-th power}),$$

with $\mu = lcm(n, m_{\lambda+1}, m_{2e_{d_h}+\lambda})$, for each $t \leq s \leq d_h$, $i = -\lambda, \ldots, 0$ and $i = e_{d_h} + 1, \ldots, e_{d_h} + \lambda$, and for each $\Xi_{h,s}$ and $\theta_{h,t}$ (recall the notational ambiguity).

4. *Each node labelled by*

$$v_p(\Xi_{f,\ell}(\bar{x}) - \Xi_{h,s}(\bar{x},\tilde{z})) \;\leq\; v_p\Pi_{h,t}(\bar{x},\tilde{z}) + i$$

has children labelled by each one of the following conditions

$v_p\frac{\Xi_{f,\ell-1}(\bar{x})-\Xi_{h,s}(\bar{x},\tilde{z})}{\theta_{f,\ell}(\bar{x})} \leq -\lambda$
$v_p\frac{\Xi_{f,\ell-1}(\bar{x})-\Xi_{h,s}(\bar{x},\tilde{z})}{\theta_{f,\ell}(\bar{x})} < 0$
$v_p\frac{\Xi_{f,\ell-1}(\bar{x})-\Xi_{h,s}(\bar{x},\tilde{z})}{\theta_{f,\ell}(\bar{x})} > 0$
$v_p(\Xi_{f,\ell-1}(\bar{x}) - \Xi_{h,s}(\bar{x},\tilde{z})) \leq v_p\theta_{h,t}(\bar{x},\tilde{z}) + i$

$\frac{\Xi_{f,\ell-1}(\bar{x})-\Xi_{h,s}(\bar{x},\tilde{z})}{\theta_{f,\ell}(\bar{x})} = \rho\cdot(\text{nonzero } N\text{-th power}) \qquad \text{with } N = m_{e_\ell+2\lambda}$
$f^{(j)}(\bar{x},\Xi_{f,\ell}(\bar{x})) = \rho\cdot(\text{nonzero } M\text{-th power}) \qquad \text{for } j = d-\ell,\ldots,d$
where $M = k\cdot m_{2e_\ell+\lambda}$, *and* k *is as in Def. 4.5, Item 3*

$$v_p\left(-\frac{f_\ell\big(\bar{x},\Xi_{f,\ell-1}(\bar{x})-\Xi_{h,s}(\bar{x},\tilde{z})\big)}{f'_\ell\big(\bar{x},\Xi_{f,\ell-1}(\bar{x})-\Xi_{h,s}(\bar{x},\tilde{z})\big)}\right) \leq v_p\theta_{h,t}(\bar{x},\tilde{z}) + i.$$

5. *Each node labelled by*

$$\Xi_{f,\ell}(\bar{x}) - \Xi_{h,\ell_h}(\bar{x},\tilde{z}) \;=\; \rho\cdot(\text{nonzero } n\text{-th power})$$

has children labelled by each of the following conditions

$v_p\frac{\Xi_{f,\ell-1}(\bar{x})-\Xi_{h,s}(\bar{x},\tilde{z})}{\theta_{f,\ell}(\bar{x})} \leq -\lambda$
$v_p\frac{\Xi_{f,\ell-1}(\bar{x})-\Xi_{h,s}(\bar{x},\tilde{z})}{\theta_{f,\ell}(\bar{x})} < 0$
$v_p\frac{\Xi_{f,\ell-1}(\bar{x})-\Xi_{h,s}(\bar{x},\tilde{z})}{\theta_{f,\ell}(\bar{x})} > 0$
$\Xi_{f,\ell-1}(\bar{x}) - \Xi_{h,s}(\bar{x},\tilde{z}) = \rho\cdot(\text{nonzero } N\text{-th power})$
with $N = \mathrm{lcm}(n, m_{e_\ell+3\lambda})$
$f^{(j)}(\bar{x},\Xi_{f,\ell}(\bar{x})) = \rho\cdot(\text{nonzero } M\text{-th power}) \quad \text{for } j = d-\ell,\ldots,d$
where $M = k\cdot m_{2e_\ell+3\lambda}$, *and* k *is as in Def. 4.5, Item 3*

$$-\frac{f_\ell\big(\bar{x},\Xi_{f,\ell-1}(\bar{x})-\Xi_{h,s}(\bar{x},\tilde{z})\big)}{f'_\ell\big(\bar{x},\Xi_{f,\ell-1}(\bar{x})-\Xi_{h,s}(\bar{x},\tilde{z})\big)} = \rho\cdot(\text{nonzero } n\text{-th power}).$$

6. *Each node labelled by*

$$h(\bar{x},\tilde{z},\theta_{f,\ell}(\bar{x})) = \rho\cdot(\text{nonzero } n\text{-th power}) \tag{5.2}$$

for some polynomial h *of degree* d_h *smaller than* k *(cf. Def. 4.5, Item 3), has children labelled by the conditions in the support set of condition (5.2), children labelled by*

$$v_p(\theta_{f,\ell}(\bar{x}) - \Xi_{h,s}(\bar{x},\tilde{z})) \;\leq\; v_p\theta_{h,t}(\bar{x},\tilde{z}) + i$$

and children labelled by

$$\theta_{f,\ell}(\bar{x}) - \Xi_{h,s}(\bar{x},\tilde{z}) \;=\; \rho\cdot(\textit{nonzero } \mu\textit{-th power})$$

with $\mu = lcm(n, m_{\lambda+1}, m_{2e_s+\lambda})$, *for each* $s,t \le d_h$, $i = \lambda,\dots,0$ *and* $i = e_d+1,\dots,e_d+\lambda$, *and for each* $\Xi_{h,s}$ *and* $\theta_{h,t}$.

7. *Each node labelled by*

$$v_p(\theta_{f,\ell}(\bar{x}) - \Xi_{h,s}(\bar{x},\tilde{z})) \;\le\; v_p\theta_{h,t}(\bar{x},\tilde{z}) + i$$

has children labelled by each one of the following conditions

$v_p\psi_{f,\ell}(\bar{x}) \ge k(j + v_p\Xi_{h,s}(\bar{x},\tilde{z}))$ *for all* $j = -\lambda+1,\dots,0,\dots,\lambda$
$v_p\psi_{f,\ell}(\bar{x}) \le k(j + v_p\Xi_{h,s}(\bar{x},\tilde{z}))$ *for all* $j = -\lambda,\dots,0,\dots,\lambda-1$
$v_p\psi_{f,\ell}(\bar{x}) \le k(v_p\theta_{h,t}(\bar{x},\tilde{z}) + i)$
$v_p\Xi_{h,s}(\bar{x},\tilde{z}) \le v_p\theta_{h,t}(\bar{x},\tilde{z}) + i$
$\psi_{f,\ell}(\bar{x}) = \rho\cdot(\textit{nonzero } N\textit{-th power})$ *with* $N = k\cdot m_{v_pk+\lambda}$
$\Xi_{h,s}(\bar{x},\tilde{z}) = \rho\cdot(\textit{nonzero } M\textit{-th power})$ *with* $M = m_{v_pk+\lambda}$

$v_p\frac{\psi_1(\bar{x})-\Xi^k_{h,s}(\bar{x},\tilde{z})}{k\Xi^{k-1}_{h,s}(\bar{x},\tilde{z})} \le v_p\theta_{h,t}(\bar{x},\tilde{z}) + i.$

8. *Each node labelled by*

$$\theta_{f,\ell}(\bar{x}) - \Xi_{h,s}(\bar{x},\tilde{z}) \;=\; \rho\cdot(\textit{nonzero } n\textit{-th power})$$

has children labelled by each one of the following conditions

$v_p\psi_{f,\ell}(\bar{x}) \ge k(j + v_p\Xi_{h,s}(\bar{x},\tilde{z}))$ *for all* $j = -\lambda+1,\dots,0,\dots,\lambda$
$v_p\psi_{f,\ell}(\bar{x}) \le k(j + v_p\Xi_{h,s}(\bar{x},\tilde{z}))$ *for all* $j = -\lambda,\dots,0,\dots,\lambda-1$
$\psi_{f,\ell}(\bar{x}) = \rho\cdot(\textit{nonzero } N\textit{-th power})$ *with* $N = k\cdot lcm(n, m_{v_pk+2\lambda})$
$\Xi_{h,s}(\bar{x},\tilde{z}) = \rho\cdot(\textit{nonzero } M\textit{-th power})$ *with* $M = lcm(n, m_{v_pk+2\lambda})$
$\frac{\psi_1(\bar{x})-\Xi^k_{h,s}(\bar{x},\tilde{z})}{k\Xi^{k-1}_{h,s}(\bar{x},\tilde{z})} = \rho\cdot(\textit{nonzero } n\textit{-th power}).$

The last structure that will be needed is the set $\mathcal{F}_r$ of conditions with respect to which a decomposition has to be carried out at the r-th level. It is defined in terms of the set $\mathcal{F}_{r+1}$. Each condition is represented by a pair (f,n) for the power condition $f(\bar{x},y) = \rho\cdot(\text{nonzero } n\text{-th power})$. Remember that the valuation conditions are just specific power conditions, by the definability of the valuation in the language with only one sort.

DEFINITION 5.4 ($\mathcal{F}_r$). *For each pair $(f,n) \in \mathcal{F}_{r+1}$ the set $\mathcal{F}_r$ contains:*

(For DECOMPOSITION[2]:)

- *the conditions labelling the leaves of the csa trees whose roots are labelled by the conditions in the support sets of each condition in $\mathcal{F}_{r+1}$ of the form $f(\bar{x},y) = \rho \cdot (\text{nonzero } n\text{-th power})$;*

(For SIMPLIFY and SAMPLE:)

- *for each quadruple $(\Xi_1,\Xi_2,\Xi_3,\Xi_4)$ of roots of functions appearing in $\mathcal{F}_{r+1}$, the conditions labelling the leaves of the csa trees whose roots are labelled respectively by*

$$v_p(\Xi_1-\Xi_2) \le v_p(\Xi_3-\Xi_4),$$
$$v_p(\Xi_1-\Xi_2) \le v_p(\Xi_3-\Xi_4)+\lambda, \text{ and}$$
$$v_p(\Xi_1-\Xi_2) < v_p(\Xi_3-\Xi_4)+2\lambda;$$

- *for each pair (Ξ_1,Ξ_2) of roots of functions appearing in $\mathcal{F}_{r+1}$, the conditions labelling the leaves of the csa trees whose roots are labelled by*

$$\Xi_1-\Xi_2 = \rho \cdot (\text{nonzero } \nu\text{-th power})$$
$$\text{with } \nu = m_{3\lambda_\mu} \text{ and } \mu = lcm(n, m_{\lambda+1}, m_{2e_d+\lambda});$$

- *for each pair (Ξ_1,Ξ_2) of roots of functions, and for each of θ-functions appearing in $\mathcal{F}_{r+1}$ and for all $i = -\lambda,\dots,0$ and $i = e_d+1,\dots,e_d+\lambda$, and for all $j = -\lambda,\dots,2\lambda$, the conditions labelling the leaves of the csa trees whose roots are labelled by*

$$v_p(\Xi_1-\Xi_2)+j \le v_p\theta(\bar{x})+i;$$

- *for each pair (θ_1,θ_2) of θ-functions appearing in $\mathcal{F}_{r+1}$, and for each $i,j = -\lambda,\dots,0$ and $i,j = e_d+1,\dots,e_d+\lambda$ the conditions labelling the leaves of the csa trees whose roots are labelled by*

$$\theta_1+i \le \theta_2+j.$$

[2]This comment and the following will become meaningful later.

6. The algorithm

The algorithm takes in input a sentence

$$(Q_1 x_1)\dots(Q_{\text{DimSp}} x_{\text{DimSp}})\phi(x_1,\dots,x_{\text{DimSp}}),$$

where ϕ is a boolean combination of atomic formulae and each Q_i is either $\forall$ or $\exists$. It outputs a quantifier free formula, equivalent to the input sentence in the theory $Th(\mathbf{Q}_p)$.

The dimension of the space is DimSp.

For each r, the set SamplePoints$_r$ will contain the first r coordinates of the sample points computed. For uniformity, the set SamplePoints$_0$ is also defined as the set containing the single element 0, which (again for uniformity) will be considered to be a tuple of length 0.

Since so far when discussing a generic stage of the algorithm, the focus was on the $(r+1)$ variable, the outer loop in the algorithm has r ranging from 0 to DimSp -1, and in the r-th loop, the $(r+1)$ variable is under consideration.

The inner recursion is implemented using the sets Call^J_{r+1}. These sets help manage the recursive calls to DECOMPOSITION. Each 5-tuple

$$(f, n, \text{Var}, \text{Cell}, e_{J-1})$$

in Call^J_{r+1} specifies the condition (f,n) with respect to which the space must be decomposed, keeps track of the changes of variable that have been performed thus far on the $(r+1)$ variable (through Var) of the cell definition attained so far which has to be further decomposed while working on the $(d-J)$st derivative of f (through Cell), and keeps track of the value of the parameter e_{J-1} (recall its recursive definition).

For each $J = 1,\dots,\max\{d | d = \text{degree of} f \in \mathcal{F}_{r+1}\}$, the variables Cad_J are used to memorize the definition of the $r+1$-st dimension of the cells. The variable Cad will eventually hold the global definition of the cells in which $\mathbf{Q}_p^{\text{DimSp}}$ is partitioned.

Algorithm: QuantifElim
Input: a sentence $\Phi = ((Q_1 x_1)\dots(Q_{\text{DimSp}} x_{\text{DimSp}})\phi(x_1,\dots,x_{\text{DimSp}}))$ where ϕ is a boolean combination of atomic formulae and each Q_i is either $\forall$ or $\exists$.
Output: a quantifier free formula QFFormula equivalent to the input sentence in the theory $Th(\mathbf{Q}_p)$.

Description:

- Let $n = \text{lcm}\{m|$ the predicate P_m appears in some atomic formula in $\phi\}$.
- $\mathcal{F}$ =INIT(ϕ, n).
- Let $\lambda = 2v_p n + 1$.
- Let $\text{SAMPLEPOINTS}_0 = \{0\}$.
- For $r = 0$ to $\text{DIMSP} - 1$ do:

  For each r-tuple $\bar{x}_0 \in \text{SAMPLEPOINTS}_r$ do:
  - Let $\text{Call}^1_{r+1} = \{(f, n, x_{r+1}, 1, 0)|(f, n) \in \mathcal{F}_{r+1}\}$.
  - Let $M = \max\{d|d =$ degree of$f \in \mathcal{F}_{r+1}\}$.
  - For each $J = 2, \ldots, M$,
    let $\text{Call}^J_{r+1} = \emptyset$.
  - Let $\text{CNT} = 0$.
  - For $J = 1$ to M do:
    For each 5-tuple $T \in \text{Call}^J_{r+1}$
    call DECOMPOSITION(J, T).
  - For $J = 0$ to CNT
    let $\text{CAD}_J = \text{SIMPLIFY}(\text{CAD}_J)$.
  - Let $\text{CAD} = \bigcup_{J=1}^{\text{CNT}} \text{CAD}_J$.
  - Let CAD =SIMPLIFY(CAD).
  - Let $\text{SAMPLEPOINTS}_{r+1}$ =SAMPLE$(\text{CAD}, \bar{x}_0)$

- Let QFFORMULA = EVALUATE$(\Phi, \text{SAMPLEPOINTS}_{\text{DIMSP}})$
- Output QFFORMULA

Procedure: INIT
Input: a formula ϕ and a positive integer n which is the least common multiple of the integers m such that a predicate P_m appears in ϕ.
Output: sets $\mathcal{F}_r$ $(r = 1, \ldots, \text{DIMSP})$ such that $\mathcal{F}_{\text{DIMSP}}$ contains, for each atomic formula in ϕ, a power condition that implies it, and for all other r, $\mathcal{F}_r$ is obtained from $\mathcal{F}_{\text{DIMSP}}$ according to Definition 5.4.
Description:

- Let $\mathcal{F}_{\text{DIMSP}} = \{(f_i, n) | f_i$ is a polynomial appearing in some atomic formula in $\phi\}$.
- For $r = \text{DIMSP} - 1$ downto 1 define $\mathcal{F}_r$, according to Definition 5.4.

Procedure: DECOMPOSITION
Input: the pair (J, T) where J defines the step of the inner recursion and T is a 5-tuple from Call_{r+1}^{J} (see the initial part of this section).
Output: if J is smaller than f's degree, the output is Call_{r+1}^{J+1}, otherwise the output is CAD_{CNT} (the definition of the $(r+1)$ dimension of the cell).
Description:

1. Let $y = \text{VAR}$. Perform a change of variable: write $f(\bar{x}, x_{r+1})$ as a function of $y = \text{VAR}$.
   Let d be the degree of f.
2. For each pair (i, j) $(i, j \in \{0, \ldots, d\})$ and for each $s \in \{0, \ldots, i-j-1\}$ and $\rho \in \Lambda_{N_2}$ for $N_2 = (i-j)p^{v_p(i-j)}(p-1)$, of indices of monomials of $f^{(d-J)}$, define $\theta_{ij}(\bar{x})$.
3. Let $e_J = v_p i_0 + e_{J-1}$.
4. Define the $(r+1)$ dimension of the space decomposition as follows:
   (a) For each i_0, and $\rho \in \Lambda_n$, define the following cells:

$$\begin{aligned}\text{CELL}_{\lambda}^{(i_0)} = \text{CELL} \wedge & \bigwedge_{j<i_0} v_p\theta_{i_0 j}(\bar{x}) - v_p y \geq \left\lceil \frac{\lambda}{i_0 - j} \right\rceil \ \wedge \\ & \wedge \bigwedge_{j>i_0} v_p\theta_{i_0 j}(\bar{x}) - v_p y \leq \left\lceil \frac{\lambda}{i_0 - j} \right\rceil \ \wedge \\ & \wedge \ y = \rho \cdot (\text{nonzero } n\text{-th power})\end{aligned}$$

(b) For each $s = 1, \dots, \lambda - 1$, pair (i_0, j) and $\sigma \in \Lambda_N$ with $N = m_\lambda$, define the following cells:

$$\begin{aligned}\textsc{Cell}_s^{(i_0 j)} = \textsc{Cell} \wedge v_p y = v_p \theta_{i_0 j}(\bar{x}) - \left\lceil \frac{s}{i_0 - j} \right\rceil \wedge \\ \wedge \bigwedge_{h<i_0} v_p \theta_{i_0 j}(\bar{x}) \leq v_p \theta_{i_0 h}(\bar{x}) \ \wedge \\ \wedge \bigwedge_{h>i_0} v_p \theta_{i_0 h}(\bar{x}) \leq v_p \theta_{i_0 j}(\bar{x}) \wedge \\ y = \sigma \cdot (\text{nonzero } N\text{-th power})\end{aligned}$$

(c) For each pair (i, j) and $\sigma \in \Lambda_N$ with $N = m_{2e_J+\lambda}$, such that

$$f(\bar{x}, \theta(\bar{x})\sigma) < 2e_J + v_p(a_{i_0}(\bar{x})\theta_{ij}(\bar{x})),$$

define the following cells:

$$\begin{aligned}\textsc{Cell}^{(ij)} = \textsc{Cell} \wedge v_p y = v_p \theta_{ij}(\bar{x}) \wedge \\ \wedge y = \sigma \cdot (\text{nonzero } N\text{-th power})\end{aligned}$$

(d) For each pair (i, j), $\rho \in \Lambda_n$ and $\sigma \in \Lambda_N$ with $N = m_{2e_J+1}$, such that

$$f(\bar{x}, \theta(\bar{x})\sigma) \geq 2e_J + v_p(a_{i_0}(\bar{x})\theta_{ij}(\bar{x})),$$

define the following cells:

$$\begin{aligned}\textsc{Cell}_{0\lambda}^{(ij)} = \textsc{Cell} \wedge v_p y = v_p \theta_{ij}(\bar{x}) \wedge \\ \wedge v_p(y - \xi_{ij}(\bar{x})\theta_{ij}(\bar{x})) \geq v_p \theta_{ij}(\bar{x}) + e_J + \lambda \wedge \\ \wedge y - \xi_{ij}(\bar{x})\theta_{ij}(\bar{x}) = \rho \cdot (\text{nonzero } n\text{-th power}) \wedge \\ \wedge y = \sigma \cdot (\text{nonzero } N\text{-th power})\end{aligned}$$

where $\xi_{ij}(\bar{x})$ is defined by

$$f(\bar{x}, \xi_{ij}(\bar{x})\theta_{ij}(\bar{x})) = 0 \ \wedge \xi_{ij}(\bar{x}) \equiv \sigma \bmod p^{2e_J+1}$$

(by Hensel's lemma it is well-defined).

(e) For each pair (i, j), for each $\rho \in \Lambda_M$, with $M = \operatorname{lcm}(m_{\lambda+1}, n)$ for each $\sigma \in \Lambda_N$ with $N = m_{2e_J+1}$, such that

$$f(\bar{x}, \theta(\bar{x})\sigma) \geq 2e_J + v_p(a_{i_0}(\bar{x})\theta_{ij}(\bar{x})),$$

and for $s = 0, \ldots, \lambda - 2$, define the following cells:

$$\begin{aligned}\textsc{Cell}_{0s}^{(ij)} = \textsc{Cell} \wedge v_p y = v_p\theta_{ij}(\bar{x}) \ \wedge \\ \wedge\, y = \sigma \cdot (\text{nonzero } N\text{-th power}) \ \wedge \\ \wedge\, v_p(y - \xi_{ij}(\bar{x})\theta_{ij}(\bar{x})) \geq v_p\theta_{ij}(\bar{x}) + e_J + s + 1 \wedge \\ \wedge\, y - \xi_{ij}(\bar{x})\theta_{ij}(\bar{x}) = \rho \cdot (\text{nonzero } M\text{-th power})\end{aligned}$$

where $\xi_{ij}(\bar{x})$ is defined as in the previous Item.

5. Expand y as VAR in the cell-defining conditions.
6. If $J < d$ do:
   (a) For each cell defined in 4a. set
   $\text{Call}_{r+1}^{J+1} = \text{Call}_{r+1}^{J+1} \bigcup \{(f, n, \textsc{Var}, \textsc{Cell}_{\lambda}^{(i_0)}, 0)\}$.
   (b) For each cell defined in 4b. set
   $\text{Call}_{r+1}^{J+1} = \text{Call}_{r+1}^{J+1} \bigcup \{(f, n, \textsc{Var}, \textsc{Cell}_{s}^{(i_0 j)}, 0)\}$.
   (c) For each cell defined in 4c. set
   $\text{Call}_{r+1}^{J+1} = \text{Call}_{r+1}^{J+1} \bigcup \{(f, n, \textsc{Var}, \textsc{Cell}^{(ij)}, 2e_J)\}$.
   (d) For each cell defined in 4d. set
   $\textsc{Var} = \textsc{Var} - \xi_{ij}(\bar{x})\theta_{ij}(\bar{x})$
   $\text{Call}_{r+1}^{J+1} = \text{Call}_{r+1}^{J+1} \bigcup \{(f, n, \textsc{Var}, \textsc{Cell}_{0\lambda}^{(ij)}, 0)\}$.
   (e) For each cell defined in 4e. set
   $\textsc{Var} = \textsc{Var} - \xi_{ij}(\bar{x})\theta_{ij}(\bar{x})$
   $\text{Call}_{r+1}^{J+1} = \text{Call}_{r+1}^{J+1} \bigcup \{(f, n, \textsc{Var}, \textsc{Cell}_{0s}^{(ij)}, 0)\}$.
7. else, if $J = d$, do:
   - $\textsc{Cnt} = \textsc{Cnt} + 1$.
   - $\textsc{Cad}_{\textsc{Cnt}} = \bigcup \textsc{Cell}_{\lambda}^{(i_0)} \ \cup \ \bigcup \textsc{Cell}_{s}^{(i_0 j)} \cup$
     $\cup \bigcup \textsc{Cell}^{(ij)} \ \cup \bigcup \textsc{Cell}_{0\lambda}^{(ij)} \ \cup \ \bigcup \textsc{Cell}_{0s}^{(ij)}$.

Procedure: SIMPLIFY
Input: the $(r+1)$ dimension SETOFCELLS of a decomposition in cells of the space.
Output: an equivalent decomposition SETOFCELLS in which each cell is defined by a single power condition and a single valuation condition.
Description:

- Let $I = 0$.
  For each center $\xi(\bar{x})$ of some cell in SetOfCells
  - let Center$[I] = \xi(\bar{x})$;
  - $I = I + 1$.
- Let Boundaries $= \{\alpha(\bar{x})|$ such that $v_p\alpha(\bar{x})$ is in a condition for some cell in SetOfCells$\} \cup \{$Center$[i] -$ Center$[j]|i < j\}$.
- Let $N = \text{lcm}\{m|\ y -$ Center$[I] = \rho \cdot$ (nonzero m-th power) appears in some cell of SetOfCells$\}$.
- Let SetOfCells $= \emptyset$.
- For each I, for each $\rho \in \Lambda_N$ and for each pair (i, j) such that $\alpha_i(\bar{x})$ and $\alpha_j(\bar{x})$ are in Boundaries, let

$$\begin{aligned}
\text{SetOfCells} \;=\; & \text{SetOfCells} \cup \\
& \{v_p\alpha_i(\bar{x}) + \lambda \le v_p(x_{r+1} - \text{Center}[I]) < \lambda + \alpha_j(\bar{x}) \wedge \\
& \quad x_{r+1} - \text{Center}[I] = \rho \cdot (\text{nonzero } N\text{-th power})\} \cup \\
& \{v_p\alpha_i(\bar{x}) + \lambda \le v_p(x_{r+1} - \text{Center}[I]) < 2\lambda + \alpha_j(\bar{x}) \wedge \\
& \quad x_{r+1} - \text{Center}[I] = \rho \cdot (\text{nonzero } N\text{-th power})\} \cup \\
& \{v_p\alpha_i(\bar{x}) + \lambda \le v_p(x_{r+1} - \text{Center}[I]) \wedge \\
& \quad x_{r+1} - \text{Center}[I] = \rho \cdot (\text{nonzero } N\text{-th power})\} \cup \\
& \{v_p(x_{r+1} - \text{Center}[I]) < \lambda + \alpha_j(\bar{x}) \wedge \\
& \quad x_{r+1} - \text{Center}[I] = \rho \cdot (\text{nonzero } N\text{-th power})\} \cup \\
& \{v_p(x_{r+1} - \text{Center}[I]) < 2\lambda + \alpha_j(\bar{x}) \wedge \\
& \quad x_{r+1} - \text{Center}[I] = \rho \cdot (\text{nonzero } N\text{-th power})\} \quad .
\end{aligned}$$

- Return SetOfCells.

Procedure: SAMPLE
Input: the $(r + 1)$ dimension SetOfCells of a decomposition in cells of the space in which each cell is defined by a single power condition and a single valuation condition.
Output: a set NewPoints of sample points in $\mathbf{Q}_p^{r+1}$ such that there is at least one point in NewPoints$\bigcap C$ for each cell C in the c.a.d. of $\mathbf{Q}_p^{r+1}$.

Description:

- Let NEWPOINTS $= \emptyset$.
- For each cell in SETOFCELLS, of the form

$$v_p\alpha_1(\bar{x}) \leq v_p(x_{r+1} - \text{CENTER}[I]) < v_p\alpha_2(\bar{x}) \wedge$$
$$x_{r+1} - \xi(\bar{x}) = \rho \cdot (\text{nonzero } N\text{-th power}),$$

  and for all $c \in \Lambda_N$, let

$$\text{NEWPOINTS} = \text{NEWPOINTS} \cup (x_{10}, \ldots, x_{r0}, \alpha_1(\bar{x}_0)c + \xi(\bar{x}_0)),$$

  where $\bar{x}_0 = (x_{10}, \ldots, x_{r0})$.

- For each cell in SETOFCELLS, of the form

$$v_p(x_{r+1} - \text{CENTER}[I]) < v_p\alpha_1(\bar{x}) \wedge$$
$$x_{r+1} - \xi(\bar{x}) = \rho \cdot (\text{nonzero } N\text{-th power}),$$

  and for all $c \in \Lambda_N$, let

$$\text{NEWPOINTS} = \text{NEWPOINTS} \cup (x_{10}, \ldots, x_{r0}, \alpha_1(\bar{x}_0)cp^{-N} + \xi(\bar{x}_0)),$$

  where $\bar{x}_0 = (x_{10}, \ldots, x_{r0})$.

- Return NEWPOINTS.

The actual elimination of quantifiers is exactly like Collins' for the reals.

Procedure: EVALUATE
Input: the pair (Φ,$\text{SAMPLEPOINTS}_{\text{DIMSP}}$) where Φ is a prenex sentence

$$(Q_1x_1)\ldots(Q_{\text{DIMSP}}x_{\text{DIMSP}})\phi(x_1, \ldots, x_{\text{DIMSP}}),$$

with each Q_i being either $\exists$ or $\forall$, and $\text{SAMPLEPOINTS}_{\text{DIMSP}}$ is the set of sample points of the c.a.d. of $\mathbf{Q}_p^{r+1}$ relative to Φ.
Output: a quantifier free formula QFFORMULA equivalent in the theory to the sentence in input.

Description:

- Set

$$\text{QFFormula} = (B_1)_{i_1} \dots (B_{\text{DimSp}})_{i_{\text{DimSp}}} \phi(\beta_{1,i}, \dots, \beta_{i_{\text{DimSp}}})$$

  where for each $r = 1, \dots, \text{DimSp}$

  - if $Q_r = \exists$, then $(B_r) = \bigvee$;
  - if $Q_r = \forall$, then $(B_r) = \bigwedge$;

  and i_r ranges over the r-th coordinates of the sample points.

- Return QFFormula.

7. Correctness

The correctness of the algorithm depends on the correctness of the single procedures that it consists of and therefore the latter will be analyzed one by one.

A couple of technical results are in order first. It will be often useful to split the space in cells in each one of which some function has a constant residue modulo p^α for some α. The lemma below and its corollary provide a useful tool to this end.

Lemma 7.1. *For any function $f(\bar{x}) : \mathbf{Q}_p^r \to \mathbf{Q}_p$, $acf(\bar{x})$ has constant residue modulo p^α if $f(\bar{x})$ is in a fixed coset of the m_α-th powers, with $m_\alpha = p^{\alpha-1}\,(p-1)$.*

Proof. Since $p^{\alpha-1}(p-1)$ is the order of the group of units of $\mathbf{Z}_p/p^\alpha$, $P_{m_\alpha}(\text{ac}f(\bar{x}))$ implies $\text{ac}f(\bar{x}) \equiv 1 \bmod p^\alpha$. Now consider any representative h of the residue class modulo p^α: $P_{m_\alpha}(\frac{\text{ac}f(\bar{x})}{h})$ implies $\text{ac}f(\bar{x}) \equiv h \bmod p^\alpha$. If $f(\bar{x}) = h \cdot u(\bar{x})^{m_\alpha}$ for some constant h, then $\text{ac}f(\bar{x}) = \text{ac}h \cdot \text{ac}(u(\bar{x})^{m_\alpha})$. Because of the choice of m_α, $u(\bar{x})^{m_\alpha} \equiv 1 \bmod p^\alpha$ and the claim follows. □

COROLLARY 7.2 (CONSTANT RESIDUE MOD p^α). *For any $f(\bar{x}) : \mathbf{Q}_p^r \to \mathbf{Z}_p$, if $f(\bar{x})$ is in a fixed coset of the m_α-th powers, and $v_p f(\bar{x})$ is either always smaller than α, or always greater than or equal to it, then $f(\bar{x})$ has constant residue modulo p^α.*

Notice that if $0 \leq v_p f(\bar{x}) \leq k$ for some positive rational integer k, if $f(\bar{x})$ has a constant residue mod $p^{\lambda+k}$, then it is in a fixed coset of the n-th powers.

Let us now consider a root-function $\xi(\bar{x})$ of some $g(\bar{x}, t)$ as in Def. 4.5.

LEMMA 7.3 (CONGRUENCES AND ROOT-FUNCTIONS). *Let the functions $\xi(\bar{x})$ and $g(\bar{x}, t)$ be as in Item 2 of Def. 4.5 above. The residue modulo p^α of $\xi(\bar{x})$ depends on the residues modulo $e + \alpha$ of the coefficients of $g(\bar{x}, t)$ viewed as a polynomial in t.*

PROOF. Let $\bar{x}, \bar{x}'$ be such that the coefficients of $g(\bar{x}, t)$ and those of $g(\bar{x}', t)$ are equivalent modulo $p^{e+\alpha}$. The Taylor expansion of $g(\bar{x}, t)$ in a neighborhood of $\xi(\bar{x})$ computed at $\xi(\bar{x}')$ is

$$g(\bar{x}, \xi(\bar{x}')) = g'(\bar{x}, \xi(\bar{x}))(\xi(\bar{x}') - \xi(\bar{x})) + f_1(\bar{x})(\xi(\bar{x}') - \xi(\bar{x}))^2 + \dots.$$

The first term of the expansion has valuation smaller than the rest, since: $v_p(\xi(\bar{x}') - \xi(\bar{x})) \geq e+1$ because both $\xi(\bar{x})$ and $\xi(\bar{x}')$ are by hypothesis congruent to k modulo p^{e+1}; $v_p g'(\bar{x}, \xi(\bar{x})) \leq e$ again by hypothesis; and $f_1(\bar{x})$ and all the successive coefficients of the expansion take values in $\mathbf{Z}_p$. Therefore

$$v_p g(\bar{x}, \xi(\bar{x}')) = v_p g'(\bar{x}, \xi(\bar{x})) + v_p(\xi(x') - \xi(\bar{x})) \leq e + v_p(\xi(\bar{x}') - \xi(\bar{x})).$$

Note that, because of the hypothesis on the coefficients of $g(\bar{x}, t)$ and $g(\bar{x}', t)$, the valuation of $g(\bar{x}, \xi(\bar{x}'))$ is not smaller than $e+\alpha$, and therefore $v_p(\xi(\bar{x}') - \xi(\bar{x})) \geq \alpha$, as required. □

Therefore, by splitting the r-dimensional space into cells, in each of which the coefficients of $g(\bar{x}, t)$ as a polynomial in t have a constant residue modulo $p^{e+\alpha}$, it is obtained that in each cell $\xi(\bar{x})$ has a fixed residue modulo p^α.

The conditions on ξ's residue modulo some power of p are conditions on the coefficients of g, i.e. on the coefficients of f and on some θ-function (up to multiplicative constants).

Let us now analyze each of the procedures that have been presented.

LEMMA 7.4 (DECOMPOSITION). *For any $r \in \{1, \dots, \text{DIMSPACE} - 1\}$, consider at the r-th stage of the outer recursion a call to DECOMPOSITION with arguments J and T, where T is the 5-tuple $(f, n, \text{VAR}, \text{CELL}, e_{J-1})$, J*

defines the step of the inner recursion and (f, n) *the power condition with respect to which the space has to be decomposed,* VAR *specifies the change of variable that has been performed at the previous stage,* CELL *describes the cell of the decomposition that has to be further split,* e_{J-1} *is the constant* CONST *from Item 3 in Subsection 4.1.*

The procedure defines the $r+1$ *dimension of cells in each of which*

$$f^{(d-J)}(\bar{x}, x_{r+1}) = \rho \cdot (\textit{nonzero } n\textit{-th power})$$

provided that the r*-dimensional space has been split into cells into each of which the conditions in the relevant support set are satisfied.*

PROOF. The present lemma simply summarizes the results already discussed in Section 4.1. □

LEMMA 7.5 (SIMPLIFY). *Consider a call to the procedure SIMPLIFY with argument the* $(r+1)$ *dimension* SETOFCELLS *of a decomposition into cells of the space. It yields in output an equivalent decomposition in which each cell is defined only by two conditions of the form*

$$y - \chi(\bar{x}) = \rho \cdot (\textit{nonzero } n\textit{-th power}), \tag{7.1}$$

$$v_p a_1(\bar{x}) \square_1 v_p(y - \chi(\bar{x})) \square_2 v_p a_2(\bar{x}) \tag{7.2}$$

for some constructive semialgebraic functions $\chi(\bar{x})$, $a_1(\bar{x})$ *and* $a_2(\bar{x})$, *and where* $\square_1$ *stands for* $\leq$ *or no condition at all, and* $\square_2$ *for* $<$ *or no condition at all, provided that the* r*-dimensional space has been split into cells in each of which:*

$$\begin{aligned}
&v_p(\chi_1(\bar{x}) - \chi_2(\bar{x})) < v_p(\chi_3(\bar{x}) - \chi_4(\bar{x})) + i \qquad \textit{with } i = 0, \lambda, 2\lambda \\
&\chi_1(\bar{x}) - \chi_2(\bar{x}) = \rho \cdot (\textit{nonzero } \nu\textit{-th power}) \quad \textit{with} \quad \nu = m_{3\lambda_\mu} \\
&\qquad\qquad \textit{and} \quad \mu = lcm(n, m_{\lambda+1}, m_{2e_d+\lambda}) \\
&v_p(\chi_1(\bar{x}) - \chi_2(\bar{x})) + j \leq v_p\theta(\bar{x}) + i \qquad \textit{with } j = -\lambda, \ldots, 2\lambda \\
&\qquad \textit{and for each } i = -\lambda, \ldots, 0 \textit{ and } i = e_d + 1, \ldots, e_d + \lambda; \\
&\theta_1 + i \leq \theta_2 + j \\
&\qquad \textit{for each } i, j = -\lambda, \ldots, 0 \textit{ and } i, j = e_d + 1, \ldots, e_d + \lambda,
\end{aligned}$$

for each quadruple of centers $(\chi_1(\bar{x}), \chi_2(\bar{x}), \chi_3(\bar{x}), \chi_4(\bar{x}))$ *and for each triple* $(\theta, \theta_1, \theta_2)$ *of* θ*-functions appearing in the cells of* SETOFCELLS.

PROOF. The first step of the proof is to examine the problem from a theoretical point of view, by quickly recalling the proof from [8], p. 163–164.

Let C be a cell with two centers, $\chi_1(\bar{x})$ and $\chi_2(\bar{x})$.
First the cell must be split into two subcells:

$$\begin{aligned} &\{(\bar{x}, y) \in C | \chi_1(\bar{x}) = \chi_2(\bar{x})\} \\ \text{and} \quad &\{(\bar{x}, y) \in C | \chi_1(\bar{x}) \neq \chi_2(\bar{x})\}. \end{aligned}$$

The first of the two subcells needs no further attention. The other is to be split according to the following cases:

1. In the subcells in which
$$v_p \frac{y - \chi_1(\bar{x})}{\chi_2(\bar{x}) - \chi_1(\bar{x})} \geq \lambda,$$
all conditions on $y - \chi_2(\bar{x})$ can be replaced by the analogous conditions on $\chi_2(\bar{x}) - \chi_1(\bar{x})$.

2. In the subcells in which
$$v_p \frac{y - \chi_1(\bar{x})}{\chi_2(\bar{x}) - \chi_1(\bar{x})} < -\lambda,$$
$y - \chi_2(\bar{x})$ can be replaced in all conditions by $y - \chi_1(\bar{x})$.

3. In the subcells in which
$$\frac{p^\lambda (y - \chi_2(\bar{x}))}{\chi_2(\bar{x}) - \chi_1(\bar{x})} \equiv 0 \bmod p^{2\lambda},$$
$\chi_2(\bar{x}) - \chi_1(\bar{x})$ can replace $y - \chi_1(\bar{x})$ in all conditions.

4. In the subcells in which
$$\frac{p^\lambda (y - \chi_2(\bar{x}))}{\chi_2(\bar{x}) - \chi_1(\bar{x})} \equiv a \bmod p^{3\lambda},$$
where $a \in \mathbf{Z}_p$, $v_p a < 2\lambda$, $a \not\equiv -p^\lambda \bmod p^{2\lambda}$ (the case $a \equiv -p^\lambda \bmod p^{2\lambda}$ reduces to Case 1), all conditions involving $y - \chi_1(\bar{x})$ can be replaced by conditions on $p^{-\lambda}(\chi_2(\bar{x}) - \chi_1(\bar{x}))(a + p^\lambda)$.

The four items above cover all the possible cases, as can be checked considering that
$$\frac{p^\lambda (y - \chi_1(\bar{x}))}{\chi_2(\bar{x}) - \chi_1(\bar{x})} = \frac{p^\lambda (y - \chi_2(\bar{x}))}{\chi_2(\bar{x}) - \chi_1(\bar{x})} + p^\lambda.$$

Let CENTERS be the set of all centers appearing in a cell. Fix some arbitrary order in the set appearing in a cell. Repeating the above steps subsequently for the pair of centers (χ_i, χ_{i+1}), for each i up to the cardinality of the set (minus one), only one center eventually remains. (The reference to the pair is not totally accurate, but is meant to be intuitive.)

The procedure SIMPLIFY takes a short cut to achieve the same goal. Following the process closely, it can be seen that the cells obtained are defined by conditions of the form

$$\begin{aligned}
&v_p(y-\chi(\bar{x})) \geq v_p(\chi_{j_2}(\bar{x})-\chi_{j_1}(\bar{x}))+\lambda\\
&v_p(y-\chi(\bar{x})) < v_p(\chi_{j_4}(\bar{x})-\chi_{j_3}(\bar{x}))+\lambda\\
&v_p(y-\chi(\bar{x})) < v_p(\chi_{j_6}(\bar{x})-\chi_{j_5}(\bar{x}))+2\lambda\\
&y-\chi(\bar{x}) = \rho\cdot(\text{nonzero } m_{3\lambda}\text{-th power})
\end{aligned}$$

with $\chi_{j_i} \in$ CENTERS, and a subset of all the conditions defining the cell, all written by substituting the old center with $\chi(\bar{x})$.

If the r-dimensional space has been decomposed so that in each cell the valuations of all the boundary functions (including the newly-introduced $(\chi_j(\bar{x})-\chi_i(\bar{x}))+\lambda$ and $(\chi_j(\bar{x})-\chi_i(\bar{x}))+2\lambda$ for all pairs of centers) are always distinct and in a constant relation to each other (one is greater in the whole cell), all of the above conditions involving the valuation can be substituted by a single one of the form (7.2).

All of the other conditions can be substituted by a single one of the form (7.1), where n is the least common multiple of all the powers appearing, since all power conditions are on $y-\chi(\bar{x})$ for a single center $\chi(\bar{x})$. □

LEMMA 7.6 (SAMPLE). *Let*

$$\begin{aligned}
&y-\chi(\bar{x}) = \rho\cdot(\textit{nonzero } n\textit{-th power}),\\
&v_p a_1(\bar{x}) \leq v_p(y-\chi(\bar{x}))\square v_p a_2(\bar{x}),
\end{aligned}$$

(where $\square$ *is either* $<$ *or no condition at all) and*

$$\begin{aligned}
&y-\chi(\bar{x}) = \rho\cdot(\textit{nonzero } n\textit{-th power}),\\
&v_p(y-\chi(\bar{x})) \leq v_p a_1(\bar{x}),
\end{aligned}$$

be the definitions of the $(r+1)$ *dimension of cells in the set* SETOFCELLS. *The procedure SAMPLE outputs a set* NEWPOINTS *of sample points in* $\mathbf{Q}_p^{r+1}$ *such that there is at least one point in* NEWPOINTS $\bigcap C$ *for each cell* C *in the*

decomposition of $\mathbf{Q}_p^{r+1}$, *provided that the* r*-dimensional space has been split into cells such that for each of them, there exists a* $\gamma \in \Lambda_n$ *for which*

$$P_n\left(a_1(\bar{x})\rho^{-1}\gamma\right)$$

holds in the cell.

PROOF. Let $\bar{x}_0$ be the first r coordinates of the sample points. Consider the first case, i.e. $v_p(y - \chi(\bar{x}))$ is bounded from below. In this case the values for the $(r+1)$ coordinate of the sample points chosen are $y_0 = a_1(\bar{x}_0)c + \chi(\bar{x}_0))$, where $c \in \Lambda_N$. Let γ_0 be such that $P_n\left(a_1(\bar{x})\rho^{-1}\gamma_0\right)$ holds in the cell. It will be proved later (see Lemma 8.1) that the coset representatives can be chosen in $\mathbf{N}$ and of valuation smaller than n. Then

$$\begin{aligned} &v_p(y_0 - \chi(\bar{x}_0)) = v_p(a_1(\bar{x}_0)\gamma_0) \geq v_p a_1(\bar{x}_0) \text{ and} \\ &y_0 - \chi(\bar{x}_0) = a_1(\bar{x}_0)\gamma_0 = \rho \cdot (\text{nonzero } n\text{-th power}). \end{aligned}$$

If the $\square$ stands for a $<$, the point belongs to the cell only if

$$v_p a_1(\bar{x}_0) + v_p\gamma_0 < v_p(a_2\bar{x}_0),$$

but this is a necessary and sufficient condition for the cell to be nonempty (cf. [8], p.165).

If the cell is empty and, in any case, for all coset representatives $c \neq \gamma_0$ the points picked are not in the cell. But insofar as each nonempty cell is represented by a sample point and the complexity does not increase greatly, the policy is acceptable.

If, on the other hand, $v_p(y - \chi(\bar{x}))$ is not bounded below in the cell definition, it can still be bounded artificially, say by $v_p a_1(\bar{x}) - n$ (the resulting cell is for sure non empty in this case). Then the same argument given above proves the correctness of the sampling procedure in this case too. $\square$

In each of the above lemmas a proviso appears involving conditions of one less variable. The algorithm works properly if all those conditions appear in the set $\mathcal{F}_r$. Indeed this is not possible in general because many of those conditions are not purely polynomial. But for each semi algebraic condition in the provisos, the conditions labelling the leaves of the csa tree whose root is labelled by that condition can (and do) appear in $\mathcal{F}_r$. The following lemma explains why it should be so.

LEMMA 7.7 (CSA TREE). *The leaves of a csa tree whose root is labelled by a semi algebraic condition*

$$h(\bar{x}, \tilde{z}, \gamma(\bar{x})) = \rho \cdot (\textit{nonzero } n\textit{-th power})$$

for some constructive semi algebraic function $\gamma(\bar{x})$ are labelled by purely polynomial semi algebraic conditions. If the space is split in cells in each one of which all the conditions labelling the leaves are verified, then so is the condition labelling the root in each cell.

This statement is essentially the proof of Lemma 4.6. Its proof is obtained by checking that for each item in the definition of a csa tree, splitting the space with respect to all the conditions labelling the children, yields a space decomposition for the parent node. The finiteness of the tree follows from the fact that the degree in each (place-holder) variable of the polynomials appearing is decreasing down the tree (keep in mind Item 1 of the definition of csa trees).

Thanks to the knowledge just acquired about csa trees, it can be directly checked that, the sets $\mathcal{F}_r$ are defined correctly. Notice that in the definition of $\mathcal{F}_r$, a hint is given as to which conditions are needed by each procedure.

The proof of correctness can now be completed:

PROPOSITION 7.8 (CORRECTNESS). *The algorithm QuantifElim computes a quantifier free formula equivalent in* $\mathrm{Th}(\mathbf{Q}_p)$ *to the formula it takes in input.*

PROOF. The lemmas above prove that each of the procedures that the algorithm relies on performs its duty at the $(r+1)$ stage, provided that some sets of conditions on r variables are satisfied by the decomposition of the r-dimensional space. The procedure INIT defines the sets $\mathcal{F}_r$ of conditions to be considered at each stage in such a way that the above is guaranteed, as has just been pointed out.

The procedure EVALUATE performs the actual elimination of quantifiers by making use of the sample points. Its correctness is based on the "cylindrical" structure of the cells that allows a separate treatment of the different coordinates of each sample point.

The correctness of the whole algorithm follows. □

8. The complexity

The complexity is analyzed by first estimating essentially the number of conditions and the maximum powers occurring in the tools defined in Section 5.

Then the coefficients' growth and the lengths of conditions and sample points are evaluated. All the bounds thus derived are eventually used in the complexity analysis of the single procedures.

Throughout, the maximum degree d of the polynomials involved appears as a parameter. Its growth is analyzed separately in order to isolate more clearly the complexity explosion. Indeed, the degree of the polynomials involved grow in a nonelementary fashion, thus blowing up the whole bound.

8.1. Analysis of the support structures. Let d be the maximum of the degrees of f in each variable.

As was mentioned earlier, it is possible to establish a bound on the number of cosets of the n-th powers for any n.

LEMMA 8.1 (NUMBER OF COSETS). *For each choice of n and p, there are at most p^λ cosets of the n-th powers in $\mathbf{Q}_p$, where $\lambda = 2v_p n + 1$. For each coset a representative c such that $v_p c < \lambda$ can be chosen.*

PROOF. The proof follows from Lemma 4.9, observing that any p-adic number w can be written as a product $m \cdot u$, where m is the residue of w modulo p^λ and, as a consequence, $u \equiv 1 \pmod{p^\lambda}$. There are p^λ residues modulo p^λ, and their canonical representatives have valuation smaller than λ. □

The following lemma gives an upper bound to the number of generalized root and θ-functions. It counts all the θ-functions that can be defined, since it is necessary for the correctness of the algorithm that they all be considered when building the sets $\mathcal{F}_r$.

LEMMA 8.2 (θ-FUNCTIONS AND ROOTS). *For a polynomial f of degree d, at most d^8 θ-functions $\theta_{f,d}$ and $d!^8$ (i.e., $O(d^{8d})$) generalized root functions Ξ can be defined.*

PROOF. A polynomial of degree d has at most $d+1$ coefficients which amounts to $(d+1)d$ pairs of distinct coefficients, and therefore functions $\psi_{ij}(\bar{x})$. One function $\theta_{f,d}$ must be defined for each

$$\psi_1(\bar{x}) = \psi_{ij}(\bar{x})\frac{p^{v_p\rho - s}}{\rho},$$

where s ranges over $\{0, \ldots, i-j-1\}$ and ρ over Λ_{N_2} for $N_2 = (i-j)p^{v_p(i-j)}(p-1)$. This means that for each ψ_{ij}, at most $(i-j)p^{4v_p(i-j)+1} \leq (i-j)^6$ functions $\theta_{f,d}$ can be defined (cf. Lemma 8.1 for the cardinality of Λ_{N_2}). Thus for a

polynomial of degree d at most $(d+1)\cdot d\cdot d^6$ functions $\theta_{f,d}$ can be defined. By the definition of generalized root functions and by the remark above on the number of root-functions that must be considered, the generalized root functions are $d!^8$ (for each $\ell = 1,\ldots,d$ one $\theta_{f,\ell}$ must be chosen). □

A bound on the value of the constant e_s that, at each stage of the inner recursion, bounds the valuation of the function $g'_{f,s}$ must be established.

LEMMA 8.3 (e_s). *The constant e_s that bounds the valuation of the function $g'_{f,s}$, where $g_{f,s}$ is built from $f^{(d-s)}$ (cf. Section 4.1), is smaller than $2^{s-2}s$.*

PROOF. The discussion in Section 4.1 amounts to the following recursive definition of e_s:

$$e_s = \begin{cases} (v_p i_0)_s & \text{if } f^{(s+1)} \text{ had a root or} \\ & v_p b_i(\bar{x})y^i \neq v_p b_j(\bar{x})y^j (\text{ for all } i,j) \\ (ordi_0)_s + 2e_{s-1} & \text{otherwise} \end{cases}$$

where $(v_p i_0)_s$ denotes here the valuation of the integer i which is the index of the monomial of minimum valuation in the cell at issue and at the s-th step of the inner recursion. Therefore $e_s \leq k_0 + 2e_{s-1}$ for $k_0 \leq \max_{i=1,\ldots,s}\{v_p i\}$ and $e_1 = 0$, i.e., $e_s < 2^{s-2}s$. □

On the basis of the above results rests the complexity analysis of the support structures defined in Section 5.

LEMMA 8.4 (SUPPORT SET). *The support set of a condition*

$$f(\bar{x},y) = \rho\cdot(\textit{nonzero } n\textit{-th power})$$

has size $O((d+1)^4 d!^8)$ and the maximum power appearing in the conditions is $O(p^{2^d}\cdot n^2)$.

PROOF. Let us count the number of conditions in each item in the definition of the support set. In the first item there are $(d+1)d!^8$ conditions (one for each choice of i and generalized root); similarly, there is the same number of conditions in the second item; in the third one there are $(d+1)^4 d!^8$, in the fourth $2(d+1)^3 d!^8$ and in the last one $(d+1)^3 d!^8$. The sum is $O((d+1)^4 d!^8)$. The maximum power involved is $N = \mathrm{lcm}_{h=1,\ldots,d}(hm_{v_p h+1}, hm_{2e_d+\lambda}, hn)$. By the bound just derived on e_d, $N \leq \mathrm{lcm}_{h=1,\ldots,d}(hm_{v_p h+1}, hm_{2^{d-1}d+\lambda}, hn)$; $m_{v_p h+1}$ is a factor of $m_{2^{d-1}d+\lambda}$ and can be therefore disregarded in the least common factor; $p^\lambda \leq n^2 p$; $p-1<p$; the least common factor is smaller than $d!\cdot p^{2^{d-1}d+1}n^2$ which is $O(p^{2^d}\cdot n^2)$. □

LEMMA 8.5 (LEAVES OF A CSA TREE). *A csa tree whose root is labelled by the polynomial form $h(\bar{x}, \xi(\bar{x}))$, where $h(\bar{x}, t)$ is a polynomial of degree d in t, has $O((v_p n)^{2^d})$ leaves with maximum power condition of order $O(n^{2^{2^d}} p^{2^{2^d}})$.*

PROOF. The tree is finite. Indeed, each node is labelled by a polynomial form of lesser degree than its parent in each variable (even though the number of variables might be increasing).

In order to give a bound on the number of leaves in the tree, it is sufficient to give a bound on the depth of the tree and on the number of children that each node can have.

Let us examine each item of the definition of a csa tree in an omologous enumeration.

1. Only one child.
2. Only one child.
3. $O(d!^8)$ children labelled by the conditions in the support set; $d!^8$ power conditions and $d!^8 d^8(2\lambda+1)$ valuation conditions (the latter resulting from $d!^8$ possible generalized root-functions, d^8 possible θ-functions, times all the possible choices for i). The total here is $O(d!^8(2 + d^8(2\lambda + 1))) = O(d!^8 2\lambda)$.
4. $d + 7$ children, $d + 1$ being conditions on f's derivatives.
5. $d + 6$ children, here too including conditions on f's derivatives.
6. Same as Item 3.
7. $4\lambda + 7$ children.
8. $4\lambda + 7$ children.

The maximum number of children that one node can have is $O(d!^8\lambda)$.

The children of each node are labelled by conditions on polynomial forms which might be on more semi algebraic functions. Up to $2d$, can be added to a polynomial form of degree d (d root functions and d θ-functions). But the degree of the resulting polynomial forms in each of the new functions is strictly smaller than d. Because of this, the process can be repeated at most d times. The total count of the new variables introduced is thus

$$2d \cdot 2(d-1) \cdot 2(d-2) \cdots 2 = 2^d d!.$$

Each one is dealt with in at most d successive levels. Therefore the depth of the tree is bounded by $d2^d d!$.

In order to put together the bounds on the fan-out and the depth of the csa tree, the growth of n, and therefore of λ, down the tree must be analyzed. Again the proof follows the pattern of the definition of csa trees.

1. n does not change.
2. n does not change.
3. In the support set n grows to $O(p^{2^d} \cdot n^2)$; valuation conditions are equivalent to power residue conditions with power 2 or 3; in the power residue condition, n grows to $O(n^2 p^{2^d})$ (this bound is derived in a way similar to the one for the support set). Globally, n grows to $O(n^2 p^{2^d})$ here.
4. The growth is in one case to $m_{e_\ell+2\lambda}$, in the other to $k \cdot m_{2e_\ell+\lambda}$; that is $O(p^{2^d} n)$.
5. n grows to $\mathrm{lcm}(n, m_{e_\ell+3\lambda})$ and to $k \cdot m_{2e_\ell+2\lambda}$; i.e., $O(p^{2^d} n^2)$.
6. Same as Item 3.
7. n grows to $k \cdot m_{v_p k+\lambda}$; this is $O(d^2 n)$.
8. n grows to $k \cdot \mathrm{lcm}(n, m_{v_p k+2\lambda})$; the order of growth is here $O(d^2 n)$.

The powers involved in the conditions labelling the children of a node whose condition is on the n-th powers is $O(n^2 p^{2^d})$. This implies that through a depth of $d2^d d!$, n increases to $O(n^{2^{2^d}} p^{2^d 2^{2^d}})$ which is also $O(n^{2^{2^d}} p^{2^{2^d}})$.

The number of leaves of a tree of fan-out w and depth h is w^{h-1}. The leaves of the csa tree are $O((2^{2^d} v_p n + 2^{2^d})^{2^d})$ which is $O((v_p n)^{2^d})$. □

Lemma 8.6 (Power conditions in sets $\mathcal{F}_r$). *If n is the maximum power appearing in $\mathcal{F}_{r+1}$, then the maximum power appearing in $\mathcal{F}_r$ is $O(n^{2^{2^d}} p^{2^{2^d}})$.*

Proof. The proof will follow the structure of the definition of sets $\mathcal{F}_r$:

- The maximum power in the conditions in the support set is $O(p^{2^d} \cdot n^2)$.
- The valuation conditions are conditions on the 2^{nd} or 3^{rd} power residue, depending on p.

- Here the power is $m_{3\lambda_\mu}$ and $\mu = \mathrm{lcm}(n, m_{\lambda+1}, m_{2e_s+\lambda})$ is the highest power involved in conditions labelling roots of the csa tree and it is $O(n^2 p^{2^d})$.
- The conditions are on 2^{nd} or 3^{rd} power residues.
- Again the conditions are on 2^{nd} or 3^{rd} power residues.

But in the sets $\mathcal{F}_r$, there are the leaves of the csa tree labelled by them. Therefore by the preceding lemma, the power grows to $O(n^{2^{2^d}} p^{2^{2^d}})$. □

At this point the maximum n involved in the power condition depending on the input can be computed.

COROLLARY 8.7 (n_{max}). *Let* N *be the power in the conditions in* $\mathcal{F}_{\text{DIMSP}}$. *Then the algorithm will have to deal at most with* n_{max}*-th power residues, where*

$$n_{max} = O((p\text{N})^{2^{2^{d}\text{DIMSP}}}).$$

PROOF. Considering that the growth of n proved in the lemma above is repeated DIMSP times, the result is obtained. □

The above allows to infer that

COROLLARY 8.8 (SUPPORT SET). *Let* N *be the power in the conditions in* $\mathcal{F}_{\text{DIMSP}}$. *Then the conditions in the support set are at most on the* n_{max}*-th power residues.*

COROLLARY 8.9 (LEAVES OF THE CSA TREE). *If* N *is the power in the conditions in* $\mathcal{F}_{\text{DIMSP}}$, *csa trees have at most* $O(2^{2^d \text{DIMSP}}(v_p\text{N})^{2^d})$ *leaves and the conditions labelling them are at most on the* μ*-th power residues, for* $\mu = O((p\text{N})^{2^{2^{d}\text{DIMSP}}})$.

The analysis of the sets $\mathcal{F}_r$ is not yet complete.

LEMMA 8.10 (NUMBER OF CSA TREES IN $\mathcal{F}_r$). *If* N *is the power appearing in the conditions in* $\mathcal{F}_{\text{DIMSP}}$, *then the leaves of* $O(2^{2^d \text{DIMSP}} v_p\text{N})$ *csa trees are in* $\mathcal{F}_r$.

PROOF. Let $\mu = O((p\text{N})^{2^{2^{d}\text{DIMSP}}})$, the greatest power that can appear. Once again the proof follows the pattern of the definition (of a set $\mathcal{F}_r$) for the sake of clarity. In each item the number of csa trees appearing in the corresponding item of the definition is estimated.

- $O(d!^8)$ from the support set.
- Three csa trees for each quadruple of generalized root functions. By the lemma on the root-functions, this amounts to $3 \cdot d!^{32}$.
- One csa tree for each pair of root-funtions: in total $d!^{16}$.
- At most $O(6\lambda_\mu^2)$ csa trees for each pair of root functions and for each product of θ-functions: in total $O(6\lambda_\mu^2 d!^{16})$.
- At most $4\lambda_\mu^2$ csa trees for each pair of θ-functions: in total $4\lambda_\mu^2 d^{18}$.

The number of csa trees whose leaves are in $\mathcal{F}_r$ is thus

$$O(d!^{32}\lambda_\mu^2) = O(2^{2^d \textsc{DimSp}}(v_p \mathrm{N})^{2^d}). \quad \square$$

Multiplying the number of csa trees times the number of leaves for each tree it can be concluded that for each condition in $\mathcal{F}_{r+1}$, the number of conditions in $\mathcal{F}_r$ is

$$O(2^{2^d \textsc{DimSp}}(v_p \mathrm{N})^{2^d}).$$

LEMMA 8.11 (SIZE OF $\mathcal{F}_r$). *If $\mathcal{F}_{\textsc{DimSp}}$ has cardinality k and* N *is the power in the conditions in $\mathcal{F}_{\textsc{DimSp}}$, then $\mathcal{F}_r$ has*

$$O(2^{2^d \textsc{DimSp}^2}(v_p \mathrm{N})^{2^d \textsc{DimSp}} \cdot k)$$

elements for each r.

PROOF. By the above lemma, $\mathcal{F}_{\textsc{DimSp}-1}$ has cardinality $O(2^{2^d \textsc{DimSp}}(v_p \mathrm{N})^{2^d} \cdot k)$, and repeating the argument DIMSP times, $\mathcal{F}_1$ has

$$O(2^{2^d \textsc{DimSp}^2}(v_p \mathrm{N})^{2^d \textsc{DimSp}} \cdot k)$$

elements and all $\mathcal{F}_r$ have less. $\square$

8.2. Lengths. The definitions of root and θ-functions require the use of auxiliary free variables. It is assumed that variables are represented in the language of the theory using binary subscripts for a single symbol, say x. If NOVAR is the number of auxiliary free variables needed, the length of each is bounded by $\log_2 \textsc{NoVar}$.

LEMMA 8.12 (INNER LOOP: COEFFICIENTS' GROWTH). *Let the length of the coefficients of the polynomial f of degree d be bounded by $|coeff|$. Through the inner loop of the algorithm, coefficients grow at most to*

$$O(d^d|coeff| + d^d \log_2 \text{NOVAR}).$$

PROOF. Let |coeff| be the maximum length of f's coefficients. The length of a θ-function of f is $O(2|\text{coeff}| + d \log_2 \text{NOVAR})$; this bound is derived by estimating the length of each element in the definition of a θ-function (Def. 4.5 Item 3). If a root function is defined as the function ξ such that

$$f(\bar{x}, \xi(\bar{x})\theta(\bar{x})) = 0 \ \wedge \ v_p(\xi - h) \geq e_d + 1,$$

the length of the definition is $O(d|\text{coeff}| + d^2 \log_2 \text{NOVAR} + 2\log_2(e_d+1) + |\theta|) = O(d|\text{coeff}| + d^2 \log_2 \text{NOVAR})$ (where $|\theta|$ is the length of the definition of a θ-function).
Once a root has been detected, a change of variable must be performed. The new coefficients are $f^{(i)}(\bar{x}, \xi(\bar{x})\theta(\bar{x}))$. Each coefficient of the i-th partial derivative is at most $\log_2(d!) = O(d)$ times longer than a coefficient of f. Computing the derivative at $\xi(\bar{x})\theta(\bar{x})$, the length of the resulting function is $O(d^2|\text{coeff}| + d^2 \log_2 \text{NOVAR} + |\xi| + |\theta|)$, where $|\xi|$ and $|\theta|$ are the lengths of the definitions of root and θ-functions. This yields a bound $O(d^2|\text{coeff}| + d^2 \log_2 \text{NOVAR})$ for the coefficients. Since in an inner loop the above can be repeated d times, the coefficients can grow to $O(d^d|\text{coeff}| + d^d \log_2 \text{NOVAR})$. □

From the proof above the maximum lengths of the definitions of θ-functions and root functions can be computed.

COROLLARY 8.13 ($|\theta|$ AND $|\xi|$). *Let $|coeff|$ be the maximum length of the coefficients of a polynomial f. The maximum length of a definition of a θ-function in an execution of an inner loop on f is $O(d^d|coeff| + d^d \log_2 \text{NOVAR})$. The same bound holds for definitions of root functions.*

The corollary is useful for computing a bound for the length of generalized root functions. A generalized root function $\Xi_{f,\ell}$ is expressed via at most ℓ root functions and ℓ θ-functions. Its length is at most $O(2d \cdot d^d|\text{coeff}| + 2d \cdot d^d \log_2 \text{NOVAR})$.

The bounds above are implicitly used in the proof of the following lemma.

LEMMA 8.14 (CSA TREE: GROWTH OF COEFFICIENTS). *Assume that $|coeff|$ bounds the length of the coefficients of the polynomials h, f and f_ℓ appearing in the definition of a csa tree, and let d bound their degree. Let n be the power in the label of some node. Then the longest coefficient of the polynomial forms in the conditions labelling the children of this node has length of $O(n^2 p^{2^d}|coeff|)$.*

PROOF. Once again the result follows by direct inspection of the definition of a csa tree. The coefficients in conditions labelling children are longer than those labelling their parent node only in a few cases: in Item 1 the growth is by a factor $O(2^d)$; in Item 2 by a factor n; in the last conditions of Items 4 and 5 they grow by a constant factor; in the conditions of Items 7 and 8, they grow again by a constant factor. Since the maximum power involved grows at most to $O(n^2 p^{2^d})$ (see the proof of Lemma 8.5), the bound on the length of the coefficients follows. □

In the rest of this section let k be the cardinality of $\mathcal{F}_{\text{DIMSP}}$, N the power in the conditions in $\mathcal{F}_{\text{DIMSP}}$, and d the maximum degree of the polynomials in $\mathcal{F}_{\text{DIMSP}}$.

LEMMA 8.15 (AUXILIARY VARIABLES). *The maximum length of strings representing auxiliary variables is*

$$\log_2 \text{NOVAR} = O(2^d \text{DIMSP}^2 + 2^d \text{DIMSP} \log_2(v_p \text{N}) + \log_2 k).$$

PROOF. One auxiliary free variable is needed for each definition of θ or root functions. For each f of degree at most d, the number of θ-functions and root functions is $O(d^8)$.

Since at each stage of the outer recursion $O(2^{2^d \text{DIMSP}^2}(v_p \text{N})^{2^d \text{DIMSP}} \cdot k)$ functions must be dealt with (the size of $\mathcal{F}_r$), for each r $O(2^{2^d \text{DIMSP}^2}(v_p \text{N})^{2^d \text{DIMSP}} \cdot k)$ auxiliary variables are needed. Over the DIMSP stages of the outer recursion, the total number is obtained by multiplying times DIMSP, still obtaining something that grows at most like $2^{2^d \text{DIMSP}^2}(v_p \text{N})^{2^d \text{DIMSP}} \cdot k$. The auxiliary variables add to the DIMSP variables appearing in the sentence in input. The total number of auxiliary variables needed is $O(2^{2^d \text{DIMSP}^2}(v_p \text{N})^{2^d \text{DIMSP}} \cdot k)$. □

The bound about the csa trees and the one derived for the inner loop of the algorithm, combined with the results proved about the maximum power, that the algorithm has to deal with and the maximum length of the strings that represent auxiliary variables, yield a final bound on the growth of the coefficients.

PROPOSITION 8.16 (MAX LENGTH OF COEFFICIENTS). *The length of the coefficients is* $O((p\text{N})^{2^{2^d \text{DIMSP}}}(|\text{COEFF}| + \log_2(k v_p \text{N})))$.

The bounds proved for generalized root functions and for θ-functions, together with the bound on the length of the representations of auxiliary variables, proves

LEMMA 8.17 (LENGTH OF $\Xi_{f,\ell}$ AND $\theta_{f,\ell}$). *Generalized root functions and θ-functions have length* $O((p\mathrm{N})^{2^{2^d \cdot \mathrm{DIMSP}}}(|\mathrm{COEFF}| + \log_2(kv_p\mathrm{N})))$.

After simplification, the $(r+1)$ dimension of each cell is defined by a pair of conditions like the following, with no boolean combinations occurring (cf. the general form of the cells after the decomposition, as described in Subsection 4.3).

$$\begin{aligned} &v_p\theta_{f_1,\ell_1}(\bar{x}) + \mathrm{CONST}_1 \square_1 v_p(x_{r+1} - \Xi_{f_2,\ell_2}(\bar{x})) \square_2 v_p\theta_{f_3,\ell_2}(\bar{x}) + \mathrm{CONST}_2 \\ &y - \Xi_{f,\ell}(\bar{x}) = \rho \cdot (\text{nonzero } n\text{-th power}) \end{aligned} \tag{8.1}$$

where the $\square_1$ stands for either $\leq$, or no condition at all, and $\square_2$ stands for either $<$, or no condition at all. Notice that this form is more general than the one in Subsection 4.3 in the sense that here root and θ-functions relative to different polynomials can appear.

LEMMA 8.18 (LENGTH OF CELLS). *The definition of each cell has length*

$$O((p\mathrm{N})^{2^{2^d \cdot \mathrm{DIMSP}}}(|\mathrm{COEFF}| + \log_2(kv_p\mathrm{N}))).$$

PROOF. Generalized root functions and θ-functions have length

$$O((p\mathrm{N})^{2^{2^d \cdot \mathrm{DIMSP}}}(|\mathrm{COEFF}| + \log_2(kv_p\mathrm{N})));$$

the constants appearing in the cell definitions are at most equal to $2\lambda_{n_{max}} + e_d$; the integer in the power condition is bounded by n_{max}; the length of the representations of the variables which are bound in the formula in input is at most $\log_2 \mathrm{DIMSP}$. Therefore the length of the r-th dimension of cell definitions is $O((p\mathrm{N})^{2^{2^d \cdot \mathrm{DIMSP}}}(|\mathrm{COEFF}| + \log_2(kv_p\mathrm{N})))$. There are DIMSP similar conditions, one for each dimension of the space. The claim follows. □

The final length that needs to be estimated relates to the definitions of sample points.

LEMMA 8.19 (LENGTH OF SAMPLE POINTS). *The length of a sample point is*

$$O((p\mathrm{N})^{2^{2^d \cdot \mathrm{DIMSP}}}(|\mathrm{COEFF}| + \log_2(kv_p\mathrm{N}))).$$

PROOF. Taking into account the general form of cells (8.1), the $(r+1)$ coordinate of a sample point is defined as a function of the previous r coordinates $\bar{x}_0$, in the worst case as

$$p^\gamma \theta_{f_1,\ell_1}(\bar{x}_0) p^{-n} c + \Xi_{f_2,\ell_2}(\bar{x}_0),$$

where $\gamma \leq 2\lambda_{n_{max}} + e_d$, and $n \leq n_{max}$. The variables $x_1, \ldots, x_r$ can be used as auxiliary variables where necessary. By the bounds on generalized root and θ-functions and on n_{max}, its length is $O((p\mathrm{N})^{2^{2^d \mathrm{DimSp}}}(|\mathrm{Coeff}| + \log_2(kv_p\mathrm{N})))$ and the length of the full definition of a sample point is of the same order. □

8.3. Analysis of the procedures. As in most of the previous subsections, let k be the cardinality of $\mathcal{F}_{\mathrm{DimSp}}$, N the power in the conditions in $\mathcal{F}_{\mathrm{DimSp}}$, and d the maximum degree of the polynomials in $\mathcal{F}_{\mathrm{DimSp}}$.

One last analysis of the definition of csa trees yields a bound on the space used by the procedure INIT.

Lemma 8.20 (INIT). *The procedure INIT needs at most space*

$$O((p\mathrm{N})^{2^{2^d \mathrm{DimSp}}}(|\mathrm{Coeff}| + \log_2 k)).$$

Proof. Again a direct inspection yields the following results about the growth of the length of the conditions labelling the nodes of a csa tree, when descending from parent to child. The growth can be of three types and is to be bounded accordingly:

- the length of the condition is expanded by a factor n; since the powers involved are no more than n_{max}, the growth factor here is $O((p\mathrm{N})^{2^{2^d \mathrm{DimSp}}})$;
- the length is increased by the length of the binary representation of some power n; since n_{max} is the largest possible, then in this case the condition length increases by $O(2^{2^d \mathrm{DimSp}} \log(\mathrm{N}p))$;
- the length is increased by addition of a number of auxiliary variables polynomial in d; by the bound on the length of strings representing auxiliary variables, the addition to the length of the conditions is here of $O(2^{2^d \mathrm{DimSp}^2} \log(k\mathrm{N}p))$;
- the length is increased by addition of a constant number of constants not greater than e_d; since $e_d = O(2^{d-2}d)$, the increase in length is $O(d)$.

By the bounds on the lengths of the coefficients and on n_{max}, the condition labelling the root of a generic csa tree has length

$$O((p\mathrm{N})^{2^{2^d \mathrm{DimSp}}}(|\mathrm{Coeff}| + \log_2(kv_p\mathrm{N}))).$$

Notice that the conditions labelling the internal nodes of a csa tree can be longer than those labelling the root and the leaves, because they might feature

some semi algebraic functions that are going to disappear down the tree. By the bounds for the growth at each level and on the depth of a csa tree ($d2^d d!$), the conditions labelling the nodes have length

$$O((p\mathrm{N})^{2^{2^d \mathrm{DimSp}}}(|\textsc{Coeff}| + \log_2(kv_p\mathrm{N}))).$$

Since a csa tree has $O(2^{2^d \mathrm{DimSp}}(v_p\mathrm{N})^{2^d})$ internal nodes and leaves, the procedure INIT needs at most space $O((p\mathrm{N})^{2^{2^d \mathrm{DimSp}}}(|\textsc{Coeff}| + \log_2 k))$. □

Next comes the count of the number of cells defined by each procedure.

Lemma 8.21 (Cells from DECOMPOSITION). *For each condition on a polynomial of degree d that appears in $\mathcal{F}_r$ ($r = 1, \ldots, \mathrm{DimSp}$), the procedure DECOMPOSITION defines $O((p\mathrm{N})^{2^{2^d \mathrm{DimSp}}})$ cells.*

Proof. Let n be the power at issue. DECOMPOSITION defines for the $(d - J)$-th derivative of the polynomial at issue:

- J^8 cells $\textsc{Cell}_\lambda^{(i_0)}$, one for each $\theta_{i_0 j}$;
- $O(J^8 p^{2\lambda+1}\lambda)$ cells $\textsc{Cell}_s^{(i_0 j)}$, one for each $\theta_{i_0 j}$, $s = 1, \ldots, \lambda-1$ and $\rho \in \Lambda_N$, where $N = m_\lambda$;
- $O(J^8(p^{2^J} + \lambda))$ cells $\textsc{Cell}^{(ij)}$, one for each θ_{ij} and coset of the N-th powers, for $N = m_{2e_J+\lambda}$;
- $O(J^8 p^{2^J})$ cells $\textsc{Cell}_{0\lambda}^{(ij)}$ one for each θ_{ij} and coset of the N-th powers, with $N = m_{2e_J+1}$;
- $O(J^8 p^\lambda p^{2^J})$ cells $\textsc{Cell}_{0s}^{(ij)}$, one for each θ_{ij}, $s = 0, \ldots, \lambda - 1$, coset of the N-th powers with $N = m_{2e_J+1}$ and coset of the M-th powers, with $M = \mathrm{lcm}(m_{\lambda+1}, n)$.

The above sums up to $O(J^8 p^\lambda p^{2^J})$. For each cell the procedure will be called again, and so on until $J = d$. The total number of calls to DECOMPOSITION on one condition in $\mathcal{F}_r$ is thus $O(d!^8 p^{d\lambda} p^{2^d d})$, the total number of cells is $O(d!^8 p^{d\lambda} p^{2^d d})$.

Finally, since n is $O((p\mathrm{N})^{2^{2^d \mathrm{DimSp}}})$, the number of cells for each condition in $\mathcal{F}_r$ is also $O((p\mathrm{N})^{2^{2^d \mathrm{DimSp}}})$. □

The total number of cells defined by DECOMPOSITION for all the conditions in $\mathcal{F}_r$ is thus $O((p\mathrm{N})^{2^{2^d \mathrm{DIMSP}}} \cdot k)$. SIMPLIFY is called in two stages: at the first stage, it is called on a family of $k \cdots (p\mathrm{N})^{2^{2^d \mathrm{DIMSP}}}$ sets, each one of $O((p\mathrm{N})^{2^{2^d \mathrm{DIMSP}}})$ cells and obtained by executing DECOMPOSITION on one condition in $\mathcal{F}_r$.

In the definition of each cell, there are at most $O((p\mathrm{N})^{2^{2^d \mathrm{DIMSP}}})$ valuation conditions and power conditions, and at most $O((p\mathrm{N})^{2^{2^d \mathrm{DIMSP}}})$ different centers. This is seen by going once again through the procedure DECOMPOSITION, and observing that each time a cell is defined, a number of conditions linear in d, and at most one new center are added to the definition. Since the procedure is called $O((p\mathrm{N})^{2^{2^d \mathrm{DIMSP}}})$ times on each cell, and at each call the cell definition is expanded as said, the bound follows.

This implies that the set BOUNDARIES and the array CENTER defined in the procedure SIMPLIFY have $O((p\mathrm{N})^{2^{2^d \mathrm{DIMSP}}})$ elements.

The set produced at the first stage by each call to SIMPLIFY contains $O((p\mathrm{N})^{2^{2^d \mathrm{DIMSP}}})$ cell definitions, five for each element of the vector of centers, pair of elements of the set BOUNDARIES and coset representative of the n_{max} powers, with $n_{max} = O((p\mathrm{N})^{2^{2^d \mathrm{DIMSP}}})$.

At the second stage, SIMPLIFY is called on the union of all the sets output by the first $O((p\mathrm{N})^{2^{2^d \mathrm{DIMSP}}} \cdot k)$ calls to SIMPLIFY. It is a set of $O((p\mathrm{N})^{2^{2^d \mathrm{DIMSP}}} \cdot k)$ cells, and the amount of centers and boundary functions that appear in the definitions is of the same order. Therefore, in the above setting:

LEMMA 8.22 (CELLS FROM SIMPLIFY). *The two stages of calls to the procedure SIMPLIFY yield $O((p\mathrm{N})^{2^{2^d \mathrm{DIMSP}}} \cdot k)$ cells.*

The procedure SAMPLE at the r-th stage, for any r, picks for each one of the cells defined, $O((p\mathrm{N})^{2^{2^d \mathrm{DIMSP}}})$ possible $(r+1)$ coordinates for the sample points—one choice for each representative c of n_{max} power cosets. Since $O((p\mathrm{N})^{2^{2^d \mathrm{DIMSP}}} \cdot k)$ cells have been defined, this amounts to $O((p\mathrm{N})^{2^{2^d \mathrm{DIMSP}}} \cdot k)$ total $(r+1)$ coordinates. Over the DIMSP stages of the outer recursion, a total of $O((p\mathrm{N})^{\mathrm{DIMSP}2^{2^d \mathrm{DIMSP}}} \cdot k)$ sample points are defined.

LEMMA 8.23 (SAMPLE POINTS). *The algorithm defines a total of*

$$O((p\mathrm{N})^{2^{2^d \mathrm{DIMSP}}} \cdot k)$$

sample points.

THEOREM 8.24. *The quantifier elimination for the theory of fields with valuation can be obtained using* $p^{2^{2^{|\phi|+d}}}$ *space, where* ϕ *is the sentence in input and* d *is the maximum degree of the polynomials that must be treated.*

PROOF. The procedure INIT needs at most space

$$O((p\mathrm{N})^{2^{2^{d_{\mathrm{DIMSP}}}}}(|\mathrm{COEFF}| + \log k)).$$

The procedure DECOMPOSITION produces a total of $O((p\mathrm{N})^{2^{2^{d_{\mathrm{DIMSP}}}}} \cdot k)$ cells. In the definition of each cell, there are at most $O((p\mathrm{N})^{2^{2^{d_{\mathrm{DIMSP}}}}})$ valuation conditions and power conditions. Each has length

$$O((p\mathrm{N})^{2^{2^{d_{\mathrm{DIMSP}}}}}(|\mathrm{COEFF}| + \log k)).$$

Thus, DECOMPOSITION needs $O((p\mathrm{N})^{2^{2^{d_{\mathrm{DIMSP}}}}} \cdot k \cdot |\mathrm{COEFF}|)$ space.

The two stages of calls to the procedure SIMPLIFY yield $O((p\mathrm{N})^{2^{2^{d_{\mathrm{DIMSP}}}}} \cdot k)$ cells, each again of size

$$O((p\mathrm{N})^{2^{2^{d_{\mathrm{DIMSP}}}}}(|\mathrm{COEFF}| + \log k)).$$

The space necessary is of the same order as the space needed by the procedure DECOMPOSITION.

The procedure SAMPLE outputs a total of $O((p\mathrm{N})^{2^{2^{d_{\mathrm{DIMSP}}}}} \cdot k)$ sample points. The formulae defining each of the sample points have length

$$O((p\mathrm{N})^{2^{2^{d_{\mathrm{DIMSP}}}}}(|\mathrm{COEFF}| + \log k)).$$

This amounts to $O((p\mathrm{N})^{2^{2^{d_{\mathrm{DIMSP}}}}} \cdot k \cdot |\mathrm{COEFF}|)$ space.

The space used by the procedure EVALUATION and the length of the output is again $O((p\mathrm{N})^{2^{2^{d_{\mathrm{DIMSP}}}}} \cdot k \cdot |\mathrm{COEFF}|)$ because the space used by the definitions of sample ponts is dominant.

The space required by the algorithm is thus

$$O((p\mathrm{N})^{2^{2^{d_{\mathrm{DIMSP}}}}} \cdot k \cdot |\mathrm{COEFF}|).$$

The parameters $|\mathrm{COEFF}|$ and k are smaller than $|\phi|$ since COEFF must occur in ϕ and k is the number of atomic formulae in ϕ.

The length of N is at most $|\phi|$ since N is the least common multiple of the integers appearing in the power conditions in ϕ and the length of a product is the sum of the lengths of the single factors. Therefore $\mathrm{N} \leq 2^{|\phi|}$. Similarly, $\mathrm{DIMSP} \leq 2^{|\phi|}$ (if the polynomials are very sparse no better bound holds). Some algebra yields the bound. □

8.4. The explosion. Let d be the maximum degree of the polynomials in the conditions in $\mathcal{F}_{r+1}$. The maximum degree of polynomials in the conditions in $\mathcal{F}_r$ is $\Omega(2^d)$. This is due to the way constructive semi algebraic functions are treated.

Let us show that $\mathcal{F}_r$ contains at least one polynomial for whose degree the lower bound claimed applies. Indeed, splitting the space into cells so that in each cell some polynomial f in $r+1$ variables and of degree d in the last one is in the fixed coset of the n-th powers, for some n, requires that the conditions in the relevant support set be treated first. Among these conditions there are some of the form

$$f^{(i)}(\bar{x}, \Xi_{f,\ell}(\bar{x})) = \rho \cdot (\text{nonzero } N\text{-th power})$$

for some i, ℓ and N.

LEMMA 8.25 (GROWTH OF THE DEGREE). *Let $\Xi_{f,d}(\bar{x})$ be a generalized root function of some polynomial f of degree d; $f'(\bar{x}, \Xi_{f,d}(\bar{x}))$ is a polynomial form of degree $d-1$ in $\Xi_{f,d}(\bar{x})$ (f' is f's first derivative). Let*

$$f'(\bar{x}, \Xi_{f,d}(\bar{x})) = \rho \cdot (\textit{nonzero } n\textit{-th power})$$

be the condition labelling the root of a csa tree. There exists some polynomial f such that the leaves of the tree just defined are labelled by conditions of the form $a(\bar{x}) = \rho \cdot (\textit{nonzero } N\textit{-th power})$ (for some suitable N) where $a(\bar{x})$ are polynomials of degree $\Omega(2^{d/4} \cdot d_{\bar{x}})$, and $d_{\bar{x}}$ is the maximum degree of the coefficients of f seen as a polynomial in the $(r+1)$ variable.

PROOF. (For simplicity this proof is written for $p \neq 2$; but with minor and obvious changes it can be rewritten for $p = 2$.) The aim is showing a path in a csa tree that leads to a leaf labelled by a polynomial condition that has the claimed relation to the condition labelling the root. First the structure of an eligible path is exposed, and then it is shown that a csa tree with such a path exists.

One of the children of the root will be labelled by a condition of the form

$$v_p(\Xi_{f,d}(\bar{x}) - \Xi^*_{f,d-1}(\bar{x})) \quad \leq \quad v_p\theta_{f,t}(\bar{x}) + i \tag{8.2}$$

for some $\theta_{f,t}$ and i (cf. Item 3 of Def. 5.3), where

$$\Xi^*_{f,d-1}(\bar{x}) \text{ is a simple root of } f' \tag{8.3}$$

such that

$$\Xi_{f,d}(\bar{x}) \text{ is not a root of } f(\bar{x}, t - \Xi^*_{f,d-1}(\bar{x})). \tag{8.4}$$

This node, in turn, has a child labelled by

$$v_p\left(-\frac{f\left(\bar{x},\Xi_{f,d-1}(\bar{x})-\Xi^*_{f,d-1}(\bar{x})\right)}{f'\left(\bar{x},\Xi_{f,d-1}(\bar{x})-\Xi^*_{f,d-1}(\bar{x})\right)}\right)\leq v_p\theta_{h,t}(\bar{x})+i \tag{8.5}$$

(cf. Item 4 of Def. 5.3). The latter label can be rewritten as

$$h_1(\bar{x},\Xi^*_{f,d-1}(\bar{x}),\Xi_{f,d-1}(\bar{x}))=\rho\cdot(\text{nonzero square}). \tag{8.6}$$

Some of the children of this node will be labelled by power conditions on the coefficients of $\Xi_{f,d-1}(\bar{x})$ in the polynomial form above. Let $h_2(\bar{x},\Xi^*_{f,d-1}(\bar{x}))$ be the coefficient of the degree 0 term. Notice that the maximum degree of the coefficents $\underline{\text{coeff}}_i(\bar{x})$ of $\Xi^*_{f,d-1}(\bar{x})$ in it is at least twice the maximum degree of the coefficients of y in $f(\bar{x},y)$. Consider that writing (8.5) in the form (8.6) involves squaring both the left and the right hand side of (8.5)—see the remarks about the definability of the valuation in the pure field language, in Section 2. Assume that $h_2(\bar{x},y)$ has degree $d-2$ in y (if necessary after Euclidean division).

Suppose that h_2 has a root $\Xi_{h_2,d-2}$. One of the children of this node is going to be labelled by a condition of the form

$$v_p(\Xi^*_{f,d-1}(\bar{x})-\Xi_{h_2,d-2}(\bar{x}))\quad\leq\quad v_p\theta_{h_2,t}(\bar{x})+i$$

for some $\theta_{h_2,t}$ and i.

Let us consider the whole path from the root to the leaves by obtaining repeating the above pattern: each node labelled by a power condition is followed by one with a label of the form (8.2), which in turn is followed by a node labelled by a condition of the form (8.5); this last node must again be followed in the path by a node labelled by a condition on the coefficients of the polynomial in its own parent node, which closes the cycle, being again a node labelled by a power condition. We also request that conditions analogous to (8.3) and (8.4) be always verified down the path.

Since in general before each node labelled by something like (8.2) a Euclidean division is necessary to reduce the degree of the polynomial at issue, the pattern described actually consists of four nodes.

The coefficients (analogous to the above) $\underline{\text{coeff}}_i(\bar{x})$ in the nodes labelled by power conditions are polynomial forms in the coefficients of f and, every second repetition of the pattern in the path, their degree is doubled.

The path has length $4d$, since at each repetition of the pattern, the degree of the polynomials involved decreases exactly by one. This path leads to a leaf

labelled by a polynomial in $\bar{x}$ whose degree is $\Omega(2^{d/2} \cdot d_{\bar{x}})$, since every eight levels the degree is doubled.

It remains to prove that there exists a semi algebraic condition such that its csa tree contains a path like the one described above.

Since all *possible* root functions of a polynomial must appear in the csa tree, a path of the form described above exists in any csa tree. It must be proven that one with length $4d$ exists, i.e., that at each Euclidean division the degree is decreased exactly by one.

The conditions constraining the decrease of the degree of the polynomials involved down the path are polynomial inequalities each requiring that some (not identically zero) polynomial form in the coefficients of f be nonzero. The system consisting of these inequalities, has at least one solution.

It might be worth repeating here that the strategy chosen in building a csa tree has no influence on the dimensions of the structure. □

One would be tempted to blame the complexity explosion on the choice of a very simple language and to add, for instance, a symbol for the valuation to fix the problem. But the treatment of conditions involving valuations would end up being essentially a rephrasing of the one presented here for the corresponding conditions (cf. Cohen's algorithm that, as mentioned, makes use of a very rich language).

Moreover, note that the lower bound given by the above lemma is clearly not optimal, since (for instance) each Euclidean division increases the degree of the coefficients as well.

If the maximum degree of polynomials in the conditions in $\mathcal{F}_r$ is an exponential in the maximum degree of the polynomials in the conditions in $\mathcal{F}_{r+1}$, the maximum exponent of the polynomials that the algorithm must deal with is a tower of exponentials whose height depends on the dimension of the space. Therefore

THEOREM 8.26. *The algorithm QuantifElim is nonelementary.*

9. Conclusions: comparing the real and the p-adic case

It is now interesting, to take a look at the real case to try and understand why the cylindric algebraic decomposition fails to give an elementary algorithm for $Th(\mathbf{Q}_p)$.

First, it will be helpful to give a brief intuitive summary from [7] of what is relevant to this discussion.

In Collins' algorithm, the analogue of the sets $\mathcal{F}_r$ are the (augmented) projections. The projection and augmented projection of a set of polynomials $\mathcal{A}$ in $r+1$ variables is obtained through quite standard polynomial manipulations. The operations involve taking the coefficients of the polynomials seen as polynomials in the $(r+1)$-st variable, taking reducta of the same polynomials (the n-th reductum is obtained by deleting the n non null terms of higher degree in the polynomial), taking principal subresultant coefficients of some Sylvester matrix, taking derivatives ([7], pp. 142 and 144). No structure similar to a csa tree is needed.

Decomposing the space into cells, in each of which every polynomial in the (augmented) projection of the set $\mathcal{A}$ is in a fixed coset of the squares, is sufficient to ensure that in each cell all root functions for the polynomials in $\mathcal{A}$ exist, are continuous, and the real-valued ones are pairwise distinct at each point ([7], Thms. 5 and 6). The roots themselves are then approximated via a procedure (ISOL) based on Sturm's theorem ([7], p. 148).

Now, turning to $Th(\mathbf{Q}_p)$, the sets $\mathcal{F}_r$ are a lot more complicated. Tracing the reason why csa trees appear in their definition, it can be seen that it depends on the way roots are approximated in the p-adic case.

The only tool here is Hensel's lemma, less powerful than Sturm's theorem and restrictive in that it works only for polynomials with integral coefficients. Moreover, it requires a bound on the valuation of the derivative. This condition not only implies the absence of double roots, a requirement also in the real case, but it is stronger than that.

It leads to a systematic separation of the valuation part (via the θ functions) from the angular component; but more crucially, it implies changes of variables at each stage of the inner recursion to get the roots out of the way.

Both in the real and in the p-adic case, it is necessary to build recursively a c.a.d. for the coefficients of the polynomial inspected. But after a change of variable occurs, in the algorithm for $Th(\mathbf{Q}_p)$, the coefficients of the original polynomial in $r+1$ variables are no longer purely polynomials in r variables, but polynomial forms in generalized root functions.

Building a c.a.d. for these functions is no longer straightforward. It requires first determining sets of purely algebraic conditions equivalent to each of the conditions, on the semi algebraic functions at issue. In other terms, each semi algebraic condition is to be replaced in the set $\mathcal{F}_r$ by those conditions that label the leaves of the csa tree whose root is labelled by that semi algebraic condition. The previous section shows that the complexity explosion occurs because of this.

It is not surprising that difficulties in the p-adic case appear, where the

real case benefits by its nicer topological properties, and makes use of Sturm's theorem.

Apologies. In [9] a double exponential space upper bound for the complexity of $Th(\mathbf{Q}_p)$ was claimed, via the algorithm discussed here. The subsequent reorganization of the algorithm exposed the complexity explosion. Notice, though, that the mentioned reorganization is not responsible for its nonelementary nature.

Acknowledgements

I am much obliged to Angus Macintyre who introduced me to the p-adic numbers, proposed that I study their complexity, and has given me precious advice in the course of following my work.

I feel indebted to the Logic Group at the Mathematics Department of the University of Turin for their interest in my research has helped me concentrate on the reorganization of the algorithm after a long period in which my attention had been diverted to other matters.

I am grateful to Bruno Codenotti for having read the final draft of my work and having given me much encouragement.

My gratitude also goes to Joachim von zur Gathen and to the referees to whom I owe some helpful remarks and suggestions.

Last but not least, I wish to thank Giovanni Faglia who has stood by me at all times while I carried out this research.

References

[1] J. Ax and S. Kochen, Diophantine problems over local fields I. *Amer. J. Math.* **87**, 1965, 605-630.

[2] J. Ax and S. Kochen, Diophantine problems over local fields II. *Amer. J. Math.* **87**, 1965, 631-648.

[3] J. Ax and S. Kochen, Diophantine problems over local fields III. *Ann. Math.* **83**, 1966, 437-456.

[4] M. Ben-Or, D. Kozen and J. Reif, The complexity of elementary algebra and geometry. *Proc. Sixteenth Ann. ACM Symp. Theor. Comput.*, 1984, 457-464.

[5] S.S. Brown, Bounds on transfer principles for algebraically closed and complete discretely valued fields. *Memoirs Amer. Math. Soc.* **15**, 204, 1978.

[6] P.J. Cohen, Decision procedures for real and p-adic fields. *Comm. Pure Appl. Math.* **XXII**, 1969, 131-151.

[7] G.E. Collins, Quantifier elimination for real closed fields by cylindrical algebraic decomposition. *Lecture Notes in Computer Science* **33**, 1975, 134-183.

[8] J. Denef, p-adic semi-algebraic sets and cell decomposition. *J. Reine Angew. Math.* **369**, 1986, 154-166.

[9] L. Egidi, The complexity of the theory of p-adic numbers. *Proc. 34th Ann. IEEE Symp. Found. Comput. Sci.*, 1993, 412-421.

[10] N. Koblitz, p-adic numbers, p-adic analysis, and the zeta-functions. *Graduate Texts in Mathematics*, **58**, Springer, 1977.

[11] A. Macintyre, On definable subsets of p-adic fields. *J. Symbolic Logic* **41**, 1976, 605-610.

[12] A. Macintyre, Twenty years of p-adic model theory. *Logic Colloquium '84*, Elsevier Science Publishers B.V. (North Holland), 1986, 121-153.

[13] A. Macintyre, Personal communication, 1991.

[14] P. Scowcroft and L. Van den Dries, On the structure of semialgebraic sets over p-adic fields. *J. Symbolic Logic* **53**, 1988, 1138-1164.

[15] V. Weispfenning, Quantifier elimination and decision procedures for valued fields. *Lecture Notes in Mathematics* **1103**, 1984, 419-472.

Manuscript received 10 March 1996

Lavinia Egidi
Dipartimento di Informatica
Università degli Studi di Torino
Corso Svizzera, 185
I-10149 Torino TO
Italy
lavinia@di.unito.it

Current address:
II Facoltà di Scienze M.F.N.
Università degli Studi di Torino
Sede di Alessandria
Corso T. Borsalino, 54
I-15100 Alessandria AL
Italy
lavinia@unial.it