Fields Institute Communications
Volume **00**, 0000

Dynamic computations inside the algebraic closure of a valued field

Franz-Viktor Kuhlmann
Mathematical Sciences Group, University of Saskatchewan, 106 Wiggins Road, Saskatoon, SK, S7N 5E6, Canada.
fvk@math.usask.ca

Henri Lombardi
Laboratoire de Mathématiques, (UMR CNRS 6623), UFR des Sciences et Techniques, Université de Franche-Comté, 25 030 BESANÇON cedex, France.
lombardi@math.univ-fcomte.fr

Hervé Perdry
Laboratoire de Mathématiques, (UMR CNRS 6623), UFR des Sciences et Techniques, Université de Franche-Comté, 25 030 BESANÇON cedex, France.
perdry@math.univ-fcomte.fr

Abstract. We explain how to compute in the algebraic closure of a valued field. These computations heavily rely on the Newton Polygon Algorithm. They are made in the same spirit as the dynamic algebraic closure of a field. They give a concrete content to the theorem saying that a valued field does have an algebraically closed valued extension. The algorithms created for that purpose can be used to perform an effective quantifier elimination for algebraically closed valued fields, which relies on a very natural geometric idea.

Contents

2000 *Mathematics Subject Classification.* Primary: 12J10, 12J20, 12Y05, 03C10; Secondary: 13P05, 68W30.

Introduction

We consider a valued field $\mathbf{K}$ with $\mathbf{V}$ its valuation ring and $\mathbf{S}$ a subring of $\mathbf{V}$ such that $\mathbf{K}$ is the quotient field of $\mathbf{S}$. We assume that $\mathbf{S}$ is an explicit ring and that divisibility inside $\mathbf{V}$ can be tested, for any two elements of $\mathbf{S}$. By explicit ring we mean a ring where algebraic operations and equality test are explicit. These are our minimal assumptions of computability. If we want more assumptions in certain cases we shall explicitly state them.

We let $\mathbf{K}^{\mathrm{ac}}$ denote the algebraic closure of $\mathbf{K}$ with $\mathbf{V}^{\mathrm{ac}}$ a valuation ring that extends $\mathbf{V}$. Our general purpose is the discussion of computational problems in $(\mathbf{K}^{\mathrm{ac}}, \mathbf{V}^{\mathrm{ac}})$ under our computability assumptions on $(\mathbf{K}, \mathbf{V})$.

Each computational problem we shall consider has as input *a finite family* $(c_i)_{i=1,\dots,n}$ *of parameters* in the ring $\mathbf{S}$. We call them the *coefficients of our computational problem.* Algorithms with the above minimal computability assumptions work uniformly. This means that some computations are made that give polynomials of $\mathbb{Z}[C_1, \dots, C_n]$, and that all our tests are of the two following types:

Is $P(c_1, \dots, c_n) = 0$? Does $Q(c_1, \dots, c_n)$ divide $P(c_1, \dots, c_n)$ in $\mathbf{V}$?

We are not interested in the way the answers to these tests are made. We may imagine these answers given either by some oracles or by some algorithms.

We shall denote the unit group by $\mathcal{U}_{\mathbf{V}}$ or $\mathbf{V}^{\times}$, $\mathcal{M}_{\mathbf{V}} = \mathbf{V} \setminus \mathcal{U}_{\mathbf{V}}$ will be the maximal ideal and $\mathcal{U}^1_{\mathbf{V}} = 1 + \mathcal{M}_{\mathbf{V}}$ is the group of units whose residue is equal to 1. We denote the value group $\mathbf{K}^{\times}/\mathcal{U}_{\mathbf{V}}$ by $\Gamma_{\mathbf{K}}$. We consider $\Gamma_{\mathbf{K}^{\mathrm{ac}}}$ as the divisible hull $\Gamma^{dh}_{\mathbf{K}}$ of $\Gamma_{\mathbf{K}}$, and the valuation $v_{\mathbf{K}^{\mathrm{ac}}}$ as an extension of $v_{\mathbf{K}}$. We shall denote the residue field $\mathbf{V}/\mathcal{M}_{\mathbf{V}}$ of $(\mathbf{K}, \mathbf{V})$ by $\overline{\mathbf{K}}$. By convention, $v(0) = \infty$ (this is not an element of $\Gamma_{\mathbf{K}}$).

We say that the value of some element x belonging to $\mathbf{K}^{\mathrm{ac}}$ is well determined if we know an integer m and two elements F and G of $\mathbb{Z}[C_1, \dots, C_n]$ such that, setting $f = F(c_1, \dots, c_n)$, with $f \neq 0$, and $g = G(c_1, \dots, c_n)$, there exists a unit u in $\mathbf{V}^{\mathrm{ac}}$ such that:

$$f x^m = u g$$

(a particular case is given by infinite value, i.e., when $x = 0$.)

We call $v(x)$ the value of x and we read the previous formula as:

$$m\, v(x) = v(g) - v(f)\,.$$

We shall use the notation $x \preceq y$ for $v(x) \leq v(y)$.

Example 0.1 Let us for example explain the computations that are necessary to compare $3v(x_1) + 2v(x_2)$ to $7v(x_3)$ when the values are given by

$$f_1 x_1^{m_1} = u_1 g_1,\ f_2 x_2^{m_2} = u_2 g_2,\ f_3 x_3^{m_3} = u_3 g_3, \quad (g_1, g_2, g_3 \neq 0)\,.$$

We consider the LCM $m = m_1 n_1 = m_2 n_2 = m_3 n_3$ of m_1, m_2, m_3. We have that

$$f_1^{n_1} x_1^m = u_1^{n_1} g_1^{n_1},\ f_2^{n_2} x_2^m = u_2^{n_2} g_2^{n_2},\ f_3^{n_3} x_3^m = u_3^{n_3} g_3^{n_3}\,.$$

So $3v(x_1) + 2v(x_2) \leq 7v(x_3)$ iff $g_1^{3n_1} g_2^{2n_2} f_3^{7n_3} \preceq f_1^{3n_1} f_2^{2n_2} g_3^{7n_3}$.

The reader can easily verify that computations we shall run in the value group are always meaningful under our computability asumptions on the ring $\mathbf{S}$.

In the same way, elements of the residue field will be in general defined from elements of $\mathbf{V}$. So computations inside the residue field are given by computations inside $\mathbf{S}$.

The constructive meaning of the existence of an algebraic closure $(\mathbf{K}^{\mathrm{ac}}, \mathbf{V}^{\mathrm{ac}})$ of $(\mathbf{K}, \mathbf{V})$ is that computations inside $(\mathbf{K}^{\mathrm{ac}}, \mathbf{V}^{\mathrm{ac}})$ never produce contradictions. The constructive proof of this constructive meaning can be obtained by considering classical proofs (of the existence of an algebraic closure) from the viewpoint of dynamical theories (see [2]).

The present paper can be read from a classical point of view as well as from a constructive one. Our results give a uniform way for computing inside $(\mathbf{K}^{\mathrm{ac}}, \mathbf{V}^{\mathrm{ac}})$ when we know how to compute inside $(\mathbf{K}, \mathbf{V})$.

In the first section we give some basic material for computation inside algebraically closed valued fields. The most important is the Newton Polygon Algorithm.

In section 2, we explain how the Newton Polygon Algorithm can be used in order to make explicit computations inside the algebraic closure of a valued field, even in the case where there is no factorization algorithm for one variable polynomials. It is sufficient to take the point of view of dynamic evaluations as in [3].

To conclude the paper, we give in section 3 a new quantifier elimination algorithm for the theory of algebraically closed valued fields (with fixed characteristic and residue field characteristic). The geometric idea for this algorithm is simple. It can be easily implemented after the work done in section 2.

1 Basic material

1.1 Multisets. A *multiset* is a set with (nonnegative) multiplicities, or equivalently, a list defined up to permutation. In particular, the roots of a polynomial $P(X)$ form a multiset in the algebraic closure of the base field. We shall use the notation $[x_1, \dots, x_d]$ for the multiset corresponding to the list $(x_1, \dots, x_d)$. The *cardinality* of a multiset is the length of a corresponding list, i.e., the sum of multiplicities occurring in the multiset.

We shall use the natural (associative commutative) additive notation for “disjoint unions” of multisets, e.g.,

$$[b, a, c, b, b, a, b, d, a, c, b] = 3[a, b] + [b, b, d] + 2[c] = 3[a] + 5[b] + 2[c] + [d]\ .$$

We call *a pairing between two multisets* what remains of a bijection between two corresponding lists when one forgets the ordering of the lists. E.g., if we consider the two lists

$$(a, a, a, a', a', a', a'') = (a_i)_{i=1,\dots,7} \quad \text{and} \quad (b, b, b', b', b'', b'', b'') = (b_i)_{i=1,\dots,7}$$

corresponding to the multisets

$$3[a] + 3[a'] + [a''] \quad \text{and} \quad 2[b] + 2[b'] + 3[b'']\,,$$

and the bijection

$$a_1 \mapsto b_3,\ a_2 \mapsto b_4,\ a_3 \mapsto b_1,\ a_4 \mapsto b_6,\ a_5 \mapsto b_5,\ a_6 \mapsto b_7,\ a_7 \mapsto b_2\ ,$$

then what remains can be described as

$$2[a \mapsto b'] + [a \mapsto b] + 3[a' \mapsto b''] + [a'' \mapsto b]\ ,$$

or equivalently as

$$2[(a,b')] + [(a,b)] + 3[(a',b'')] + [(a'',b)] \,.$$

This is a multiset of pairs that gives by the canonical projections the initial multisets $3[a] + 3[a'] + [a'']$ and $2[b] + 2[b'] + 3[b'']$.

This notion can be extended to r multisets $M_1, \dots, M_r$ with same cardinality k: a pairing between the M_i's is a multiset of r-tuples that gives by the canonical projections the initial multisets $M_1, \dots, M_r$.

The notion of multisets is a natural one when dealing with roots of a polynomial in an abstract setting. Multiplicity is relevant, but in general there is no canonical ordering of the roots.

Dynamic evaluation in [3, 4] can be understood as a way of computing with root multisets.

1.2 The Newton Polygon. Here we recall the well known Newton Polygon Algorithm.

The Newton polygon of a polynomial $P(X) = \sum_{i=0,\dots,d} p_i X^i \in \mathbf{K}[X]$ (where $p_d \neq 0$) is obtained from the list of pairs in $\mathbb{N} \times (\Gamma_{\mathbf{K}} \cup \{\infty\})$

$$((0, v(p_0)), (1, v(p_1)), \dots, (d, v(p_d))) \,.$$

The Newton polygon is "the bottom convex hull" of this list. It can be formally defined as the extracted list $((0, v(p_0)), \dots, (d, v(p_d)))$ verifying: two pairs $(i, v(p_i))$ and $(j, v(p_j))$ are two consecutive vertices of the Newton polygon iff:

$$\begin{array}{rl} \text{if } 0 \le k < i \text{ then} & (v(p_j) - v(p_i))/(j-i) \;>\; (v(p_i) - v(p_k)/(i-k)) \\ \text{if } i < k < j \text{ then} & (v(p_k) - v(p_i))/(k-i) \;\geq\; (v(p_j) - v(p_i))/(j-i) \\ \text{if } j < k \le d \text{ then} & (v(p_k) - v(p_j))/(k-j) \;>\; (v(p_j) - v(p_i))/(j-i) \end{array}$$

Let $P(X) = p_d \prod_{i=1}^{d} (X - x_i)$ in $\mathbf{K}^{\mathrm{ac}}[X]$. It is easily shown that if $(i, v(p_i))$ and $(j, v(p_j))$ are two consecutive vertices in the Newton polygon of the polynomial P, then the zeros of P in $\mathbf{K}^{\mathrm{ac}}$ whose value in $\Gamma_{\mathbf{K}}^{dh}$ equals $(v(p_i) - v(p_j))/(j-i)$ form a multiset with cardinality $j - i$.

Proof Order the x_i's in non-decreasing order of the values $v(x_i)$. We give the proof for an example. Assume for instance that

$$\nu_1 = v(x_1) = v(x_2) < \nu_3 = v(x_3) = v(x_4) = v(x_5) < \nu_6 = v(x_6) \cdots$$

Let us express p_{d-j}/p_d as a symmetric function of the roots. We see immediately that

$$\begin{array}{rcl} v(p_{d-1}) & \geq & v(p_d) + \nu_1 \\ v(p_{d-2}) & = & v(p_d) + 2\nu_1 \\ v(p_{d-3}) & \geq & v(p_d) + 2\nu_1 + \nu_3 > v(p_d) + 3\nu_1 \\ v(p_{d-4}) & \geq & v(p_d) + 2\nu_1 + 2\nu_3 \\ v(p_{d-5}) & = & v(p_d) + 2\nu_1 + 3\nu_3 \\ v(p_{d-6}) & \geq & v(p_d) + 2\nu_1 + 3\nu_3 + \nu_6 > v(p_d) + 2\nu_1 + 4\nu_3 \end{array}$$

So the two last edges of the Newton polygon are $((d-2, v(p_{d-2})), (d, v(p_d)))$ with slope $-2\nu_1$ and $((d-5, v(p_{d-5})), (d-2, v(p_{d-2})))$ with slope $-3\nu_3$, giving the wanted result. $\square$

Now we can give an answer to the following problem.

Computational problem 1.1 (Multiset of values of roots of polynomials)
Input: *A polynomial* $P \in \mathbf{K}[X]$ *over a valued field* $(\mathbf{K}, \mathbf{V})$.
Output: *The multiset* $[v(x_1), \dots, v(x_n)]$ *where* $[x_1, \dots, x_n]$ *is the multiset of roots of* P *in* $\mathbf{K}^{\mathrm{ac}}$.

This problem is solved by the following algorithm, which is widely used in the sequel.

Newton Polygon Algorithm The number n_∞ of roots equal to 0 (i.e., with infinite value) is read off from P. Let $P_0 := P/X^{n_\infty}$. Compute the Newton polygon of P_0, compute the slopes of the edges and output the answer. □

1.3 Generalized Tschirnhaus transformation. We recall a well known way of computing in algebraic extensions, which we will use freely in our paper. We call this method the *generalized Tschirnhaus transformation.*

Let $\mathbf{K}$ be a field, $(P_j)_{j=1,\dots,r}$ be a family of monic polynomials in $\mathbf{K}[X]$, and

$$P_j(X) = (X - \xi_{j,1}) \cdot \ldots \cdot (X - \xi_{j,d_j})$$

their factorizations in $\mathbf{K}^{\mathrm{ac}}[X]$. Take $Q(X_1, \dots, X_r) \in \mathbf{K}[X_1, \dots, X_r]$, and let $d = d_1 \cdots d_r$. We claim that the polynomial

$$T_Q(Z) = (Z - Q(\xi_{1,1}, \dots, \xi_{r,1})) \cdot \ldots \cdot (Z - Q(\xi_{1,d_1}, \dots, \xi_{r,d_r}))$$

of degree d is the characteristic polynomial of A_Q , where A_Q is the matrix of the multiplication by $Q(x_1, \dots, x_r)$ inside the d-dimensional $\mathbf{K}$-algebra

$$\mathbf{K}[x] := \mathbf{K}[X_1, \dots, X_r] / \langle P_1(X_1), \dots, P_r(X_r) \rangle \ .$$

We give a proof of this well known fact, for which we found no reference. We prove a slightly more general result, which deals with roots of so-called *triangular systems.* Moreover, the computation works in arbitrary commutative rings.

Definition 1.2 *Let* $\mathbf{A} \subset \mathbf{B}$ *be commutative rings.*

1. *Take a system of polynomials*
$$\overline{P} = (P_1, \dots, P_r) \quad \textit{where} \ \ P_1(X_1) \in \mathbf{A}[X_1],$$
$$P_2(X_1, X_2) \in \mathbf{A}[X_1, X_2], \dots, P_r(X_1, \dots, X_r) \in \mathbf{A}[X_1, \dots, X_r] \ .$$
This system is called a triangular system *if each* P_i *is monic w.r.t.* X_i.
2. *The* quotient algebra *is* $\mathbf{A}[X_1, \dots, X_r] / \langle P_1, \dots, P_r \rangle = \mathbf{A}[x_1, \dots, x_r]$ *where* x_i *is the class of* X_i. *We denote it by* $\mathbf{A}_{\overline{P}}$. *Let* $d_i = \deg_{X_i}(P_i)$. *Then* $\mathbf{A}_{\overline{P}}$ *is a free* $\mathbf{A}$*-module of rank* $d_1 d_2 \cdots d_r$ *with "monomial basis"* $(x_1^{\mu_1} \cdots x_r^{\mu_r})_{\mu_i < d_i}$. *Note that we may assume w.l.o.g. that* $\deg_{X_j}(P_k) < d_j$ *for* $k > j$.
3. *A vector* $\alpha = (\alpha_1, \dots, \alpha_k) \in \mathbf{B}^r$ *is called a* root vector of $\overline{P}$ *(or* a solution of $\overline{P}$*) if*
$$P_1(\alpha_1) = P_2(\alpha_1, \alpha_2) = \ldots = P_k(\alpha_1, \dots, \alpha_r) = 0 \,.$$
4. *Assume for simplicity that* $r = 3$. *We say that the system* $\overline{P}$ *fully splits in* $\mathbf{B}$ *if* $\mathbf{B}$ *contains elements* ξ_i $(i \le d_1)$, $\xi_{i,j}$ $(i \le d_1,\, j \le d_2)$, *and* $\xi_{i,j,k}$ $(i \le d_1,\, j \le d_2,\, k \le d_3)$ *such that*
$$\left.\begin{array}{rcll} P_1(X) & = & \prod_{i \le d_1} (X - \xi_i) & \\ P_2(\xi_i, Y) & = & \prod_{j \le d_2} (Y - \xi_{i,j}) & (i \le d_1) \\ P_3(\xi_i, \xi_{i,j}, Z) & = & \prod_{k \le d_3} (Z - \xi_{i,j,k}) & (i \le d_1,\, j \le d_2) \end{array}\right\} \qquad (1)$$

5. *When* $\mathbf{A} = \mathbf{K}$ *and* $\mathbf{B} = \mathbf{K}^{\mathrm{ac}}$*, two systems with the same variables are called* coprime systems *if they have no common root vector.*

In order to simplify notations, we give our result for the case $r = 3$.

Proposition 1.3 *Let* $\mathbf{A} \subset \mathbf{B}$ *be commutative rings and* $\overline{P} = (P_1, P_2, P_3)$ *a triangular system over* $\mathbf{A}$ *which fully splits in* $\mathbf{B}$ *with equations* (1)*. Let* $Q(x_1, x_2, x_3) \in \mathbf{A}_{\overline{P}}$*,* μ_Q *be the* $\mathbf{A}$*-linear endomorphism of* $A_{\overline{P}}$ *representing multiplication by* Q*, and* $C_Q(Z)$ *the characteristic polynomial of* μ_Q*. Then we have*

$$C_Q(Z) = \prod_{i \le d_1,\, j \le d_2,\, k \le d_3} (Z - Q(\xi_i, \xi_{i,j}, \xi_{i,j,k})) \tag{2}$$

Proof Note that we could have chosen $Q \in \mathbf{A}[X_1, X_2, X_3]$. But if $(\alpha,\, \beta,\, \gamma)$ is a root vector of $\overline{P}$ in an extension of $\mathbf{A}$, it is clear that $Q(\alpha, \beta, \gamma)$ depends only of the class of Q in $\mathbf{A}_{\overline{P}}$, so equation (2) is meaningful.

By Cayley-Hamilton $C_Q(\mu_Q) = 0$ and since $\mu_Q(1) = Q$, $C_Q(Q) = 0$. This implies that $C_Q(Q(\alpha, \beta, \gamma)) = 0$ each time we have a root vector $(\alpha,\, \beta,\, \gamma)$ of $\overline{P}$ in an extension of $\mathbf{A}$ since $\mathbf{A}[\alpha, \beta, \gamma]$ is a homomorphic image of $\mathbf{A}_{\overline{P}}$.

So the proposition is proved in the "good case" where $\mathbf{B}$ is a domain and all the root vectors in (1) give distinct values for $Q(\xi_i, \xi_{i,j}, \xi_{i,j,k})$: the RHS and LHS in (2) are monic univariate polynomials with the same roots, all being distinct.

Now we give the proof for the "generic case" where the $\xi_i, \xi_{i,j}, \xi_{i,j,k}$ and the coefficients $q_{i,j,k}$ of Q are indeterminates. This means that $\mathbf{B}$ can be replaced by a ring generated over $\mathbb{Z}$ by these indeterminates, and $\mathbf{A}$ can be replaced by the subring of $\mathbf{B}$ generated by the coefficients of Q and by the coefficients of P_1, P_2, P_3 which are defined by equations (1). In this generic case, $\mathbf{B}$ is an integral domain and all the $Q(\xi_i, \xi_{i,j}, \xi_{i,j,k})$ are distinct. So the generic case is a good case and we are done.

Finally, note that all non-generic cases are homomorphic images of the generic case. □

We give another slight generalization, which can be proved in a similar way. Let $Q, R \in \mathbf{A}[X_1, \dots, X_r]$ with $R(\xi)$ invertible in $\mathbf{B}$ for all the root vectors in (1). Let $F = Q/R$. Then A_R is an invertible matrix (over $\mathbf{B}$) and the polynomial

$$T_F(Z) = \prod_{i \le d_1,\, j \le d_2,\, k \le d_3} (Z - F(\xi_i, \xi_{i,j}, \xi_{i,j,k}))$$

is the characteristic polynomial of $A_Q(A_R)^{-1}$.

2 Dynamic computations in the algebraic closure

Dynamic computations in the algebraic closure of a valued field are an extension of dynamic computations in the algebraic closure of a field as explained in [3, 4]. First let us recall these ones.

2.1 Dynamic algebraic closure. The following algorithms tell us how to compute dynamically in the algebraic closure of $\mathbf{K}$ when we do not want to (or we cannot) use factorization algorithms in $\mathbf{K}[X]$.

First we examine the problem of adding one root of a monic polynomial without factorization algorithm. If we are able to compute in the field so created, then we are able to compute recursively in any finite extension given by adding one after the other roots of several polynomials. In fact, since there is a priori an ambiguity about what root we have introduced (distinct roots give in general non-isomorphic fields), we have to compute all possible cases.

Computational problem 2.1 (computational problem à la D5)
Input: *Let P (of degree ≥ 2) and Q be polynomials in $\mathbf{K}[X]$.*
Output: *Give correct answers to the following questions:*

(1) *Is Q zero at each root of P in $\mathbf{K}^{\mathrm{ac}}$?*
(2) *Is Q nonzero at each root of P in $\mathbf{K}^{\mathrm{ac}}$?*
(3) *If the two answers are "No", compute two factors P_1 and P_2 of P and two polynomials U_1, U_2 such that:*
— *Q is zero at each root of P_1 in $\mathbf{K}^{\mathrm{ac}}$,*
— *Q is nonzero at each root of P_2 in $\mathbf{K}^{\mathrm{ac}}$,*
— *P_1 and P_2 are coprime, $P_1U_1 + P_2U_2 = 1$,*
— *each root of P in $\mathbf{K}^{\mathrm{ac}}$ is a root of P_1P_2.*

We give two natural solutions of the previous problem.

Algorithm SquarefreeD5 *(solving computational problem 2.1 when P is a squarefree polynomial)*
Assume that P is squarefree.
Compute the monic GCD P_1 of P and Q.
If $P_1 = 1$ then answer "Yes" to the second question;
else if $\mathrm{lc}(P)P_1 = P$ then answer "Yes" to the first question;
else return P_1, $P_2 := P/P_1$ and polynomials U_1, U_2 s.t. $P_1U_1+P_2U_2 = 1$. □

Algorithm BasicD5 *(solving computational problem 2.1)*
Compute the monic GCD P_1 of P and Q.
If $P_1 = 1$ then answer "Yes" to the second question;
else compute the monic polynomial P_2 such that:
P_2 divides P, $\mathrm{GCD}(P_1, P_2) = 1$ and P divides $P_1^m P_2$ (for some m);
if $P_2 = 1$ then answer "Yes" to the first question, and replace P by P_1;
else return P_1, P_2 and polynomials U_1, U_2 s.t. $P_1U_1 + P_2U_2 = 1$. □

The replacement of P by P_1 is not used in the algorithm itself, but is meant for use by subsequent algorithms because if $P_2 = 1$ then P_1 has the same roots as P but possibly smaller degree.

Remark 2.2 Observe that $P_2 = P/\gcd(P_1^k, P) = P/\gcd(Q^k, P)$ where $k = 1 + \deg(P) - \deg(P_1)$. We can also get P_2 by iteration of the process: start with $R = P$; replace R by $R/\gcd(R, Q)$ (here $\gcd(R, Q)$ means the monic GCD of R and Q), until the GCD is 1.

If P is monic and the ring $\mathbf{S}$ is normal then P_1 and P_2 are in $\mathbf{S}[X]$, but it is not always easy to make this result explicit. Nevertheless we can always compute P_1 and P_2 using coefficients in the quotient field of $\mathbf{S}$: the GCD computation may use pseudo divisions instead of divisions. The use of subresultant polynomials may improve the efficacity of the algorithm.

We can understand the previous algorithms as breaking the set of roots of a polynomial in distinct subsets anytime that some objective distinction may be done between the roots. Their stupendous simplicity is certainly the main reason explaining their non-universal use in the literature about algebraic extensions of fields.

Remark 2.3 If we see the roots of P as a multiset, and if we want to keep the information concerning multiplicities, the output

- (P_1, P_2) with P_1, P_2 coprime and each root of P in $\mathbf{K}^{\mathrm{ac}}$ is a root of P_1P_2.

is not the good one. We need in this case one of the two following outputs:

- (P_1, P_2) with P_1, P_2 coprime and $P_1P_2 = P$.

or in a more economic way for future computations:

- (P_1, P_2) with P_1, P_2 coprime, $P_1P_2 = P$ and a decomposition of each P_i as a product of powers of coprime polynomials.

The computational problem corresponding to the first output can be solved by the following slight variant of **BasicD5**.

Algorithm MultisetD5 *(solving a multiset variant of computational problem 2.1).*
Input: Let P (of degree ≥ 2) and Q be polynomials in $\mathbf{K}[X]$.
Output: (P_1, P_2) with P_1, P_2 coprime, $P_1P_2 = P$, Q is zero at each root of P_1 in $\mathbf{K}^{\mathrm{ac}}$, Q is nonzero at each root of P_2 in $\mathbf{K}^{\mathrm{ac}}$.
Compute the monic GCD R_1 of P and Q.
If $R_1 = 1$ then return $P_1 = 1$, $P_2 = P$
  else compute the monic polynomial P_2 such that:
    P_2 divides P, $\mathrm{GCD}(R_1, P_2) = 1$ and P divides $R_1^m P_2$ (for some m).
  return P_2, $P_1 = P/P_2$ and polynomials U_1, U_2 such that $P_1U_1 + P_2U_2 = 1$. $\square$

We now explain the recursive use of algorithms **SquarefreeD5** and **BasicD5**. Note that root vectors of a triangular system $\overline{P}$ as in definition 1.2 form a multiset of cardinality $d = \prod_i \deg_{X_i}(P_i)$.

Computational problem 2.4 (computing in extensions generated by several successive algebraic elements)
Input:

- *A triangular system of polynomials* $\overline{P} = (P_1, \ldots, P_n)$*:*

$$P_1(X_1) \in \mathbf{K}[X_1],\ P_2(X_1, X_2) \in \mathbf{K}[X_1, X_2],\ \ldots, P_k(X_1, \ldots, X_k) \in \mathbf{K}[X_1, \ldots, X_k]\,.$$

- *A finite list of polynomials* $Q_1, \ldots, Q_r$ *in* $\mathbf{K}[X_1, \ldots, X_k]$.

Output:

- *A list of coprime triangular systems* $\overline{S^{(1)}}, \ldots, \overline{S^{(\ell)}}$ *whose root vectors form a partition of the set of all solutions of the initial triangular system* $\overline{P}$*, such that for each* j*, the* r*-tuple of signs for the tuple* $(Q_1(\overline{x}), \ldots, Q_r(\overline{x}))$ *(the sign of* y *is either* 0 *if* $y = 0$ *or* 1 *if* $y \neq 0$*), is the same for every root vector* $\overline{x} = (x_1, \ldots, x_k)$ *of* $\overline{S^{(j)}}$ *.*
- *For each triangular system* $\overline{S^{(j)}}$*, this fixed* r*-tuple of signs.*

In the general case, we can solve the previous problem in the following way.

Algorithm TriangularBasicD5 *(solving computational problem 2.4)*
Use **BasicD5** recursively. More precisely, consider that Q and P_k are polynomials in the variable X_k with parameters $(x_1, \ldots, x_{k-1})$. When making the computations of **BasicD5** we have to solve some tests

“ Is $R(x_1, \ldots, x_{k-1})$ equal to zero or not ? ”

for some polynomials R given by the computation. So we have to solve the same kind of problem with one variable less. Hence, a recursive computation will produce the answer. $\square$

In the case of a perfect field, we can use **SquarefreeD5** recursively. To see why this works, we have to recall how to compute the squarefree part of a polynomial in one variable in this case.

Algorithm SquarefreePart *(compute the squarefree part of a polynomial in one variable in the case of a perfect field)*
We assume that $\mathbf{K}$ is a perfect field. In the characteristic p case we assume that getting p-th roots is explicit inside $\mathbf{S}$.

Input: A polynomial $P \in \mathbf{S}[X]$.
Output: P_1 the squarefree part of P.

If the characteristic is zero then $P_1 = P/\gcd(P, P')$.

If the characteristic is p then let $P_1 = 1$ and:
Iterate the following process:
  Beginning with $R = P$ iterate the following process:
    If $R = Q(X^p)$ then replace R by $R^{1/p}$ else replace R by $R/\gcd(R, R')$
  until you find $\gcd(R, R') = 1$.
  Replace P_1 by $P_1 \cdot R$
  Iterate the following process:
    Replace P by $P/\gcd(P, R)$
  until you find $\gcd(P, R) = 1$
until $P = 1$. $\square$

We suggest that the reader apply the algorithm to a polynomial of the form $Q_1(X^p)^2 Q_2(X^{p^2})$ with $p \neq 2$, in order to see why the loops in this algorithm are necessary.

AlgorithmPerfectTriangularD5 *(solving computational problem 2.4 in the case of a perfect field)*
We assume that $\mathbf{K}$ is a perfect field. In the characteristic p case we assume that getting p-th roots is explicit inside $\mathbf{S}$.

In a first big step we replace the initial system by a disjunction of coprime systems that are "squarefree".
For each polynomial in the triangular system, we use **SquarefreePart** and (recursively) **SquarefreeD5** to replace it by a "squarefree" polynomial.
More precisely, first we replace P_1 by its squarefree part S_1.
Then we try to apply **SquarefreePart** to the polynomial P_2 as if the quotient algebra $\mathbf{K}[X_1]/S_1(X_1)$ were a field. If this is not possible, **SquarefreeD5** produces a splitting of S_1. In each branch so created the computation is possible and we can replace P_2 by its squarefree part.
For example, we may get three branches with the following properties. In the first one, the squarefree polynomial $P_{1,1}$ replaces P_1, and P_2 is already squarefree, so that $P_{2,1} = P_2$. In the second one, the squarefree polynomial $P_{1,2}$ replaces P_1, and the squarefree part of P_2 is given by $P_{2,2}$ with degree $\deg(P_2) - 1$. In the third one, $P_{1,3}$ replaces P_1 and the squarefree part of P_2 is given by $P_{2,3}$ with degree $\deg(P_2) - 4$. Then we introduce P_3 in every branch previously created and try to apply **SquarefreePart** to the polynomial P_3 as if the corresponding quotient

algebra $\mathbf{K}[X_1, X_2]/\langle P_{1,i}(X_1), P_{2,i}(X_1, X_2)\rangle$ were a field. If this is not possible, **SquarefreeD5** produces a splitting of $P_{1,i}$ or $P_{2,i}$.
And so on.
When we have introduced all P_i's, we get a tree. Each leaf of the tree corresponds to a new triangular system where all successive polynomials replacing the P_i's are "strongly squarefree" (the squarefreeness is certified by a Bezout identity in the suitable quotient algebra). Distinct leaves correspond to coprime triangular systems. So the set of root vectors of $\overline{P}$ is partitioned into distinct subsets, each one corresponding to a leaf of the tree.

Now we describe the second "big step". At each leaf of the tree we search for the signs of the Q_j's using **SquarefreeD5** as if the corresponding quotient algebra were a field. If this is not possible, new splittings are produced. □

Remark 2.5 Slight variants of the above algorithms give a partition of the *multiset* of solutions of the triangular system $\overline{P}$ in disjoint multisets that are defined by coprime triangular systems $\overline{S'^{(j)}}$, each Q_i having a constant sign at the zeros of each $\overline{S'^{(j)}}$.

Remark 2.6 The above algorithms can be generalized in order to search systematically for solutions of any system of sign conditions: equalities need not be in a triangular form. So they can be seen as quantifier elimination algorithms in the first order theory of algebraically closed extensions of some explicitly given field $\mathbf{K}$.

In the following subsection we show that the same kind of computations are possible in the case of valued fields.

2.2 Dynamic algebraic closure of a valued field.

Roots of one polynomial. The valued algebraic closure of $(\mathbf{K}, \mathbf{V})$ is well determined up to isomorphism. So the following computational problem makes sense.

Computational problem 2.7 (Simultaneous values)
Input: *polynomials P (monic) and $Q_1, \ldots, Q_r$ in $\mathbf{K}[X]$. Call $[x_1, \ldots, x_d]$ the multiset of roots of P in $\mathbf{K}^{\mathrm{ac}}$.*
Output: *The multiset $[(v(x_i), v(Q_1(x_i)), \ldots, v(Q_r(x_i)))]_{i=1,\ldots,d}$ of $(r+1)$-tuples of values.*

This problem is solved by the following algorithm.

Algorithm SimVal *(solving computational problem 2.7)*
We start with the case $r = 1$. Assume w.l.o.g. that $P(0) \neq 0$. The multiset $[\nu_i]_{i=1,\ldots,d}$ of (finite) values of the x_i's is given by the Newton Polygon Algorithm for P.

For $m, n \in \mathbb{N}$, the polynomial

$$S_{m,n}(X) = (X - x_1^m Q_1(x_1)^n) \cdot \ldots \cdot (X - x_d^m Q_1(x_d)^n)$$

is the characteristic polynomial of the matrix $A^m(Q_1(A))^n$ where A is the companion matrix of P.
So, using the Newton polygon of $S_{m,n}$ we know the multiset

$$[m\, v(x_i) + n\, v(Q_1(x_i))]_{i=1,\ldots,d} = [m\, \nu_i + n\, \nu_{1,i}]_{i=1,\ldots,d}$$

for any (m, n).
We compute first the multiset $[\nu_{1,i}]_{i=1,\ldots,d}$.

We want to compute the correct pairing between the two multisets $[\nu_i]_{i=1,\dots,d}$ and $[\nu_{1,i}]_{i=1,\dots,d}$.
Assume first that no $\nu_{1,i}$ is infinite.
Let us call a *bad coincidence for* n_1 an equality

$$\nu_i + n_1\,\nu_{1,h} = \nu_j + n_1\,\nu_{1,k} \quad \text{with} \quad \nu_i \neq \nu_j, \quad i,j,h,k \in \{1,\dots,d\}\ .$$

If there is no bad coincidence for some n_1 then we can state this fact by considering the two sets $\{\nu_i\ : i = 1,\dots,d\}$ and $\{\nu_{1,i}\ : i = 1,\dots,d\}$. Note also that there are at most $(d(d-1)/2)^2$ "bad values" of n_1. So we can find a "good" n_1 by a finite number of computations. Fix a "good" n_1. From the multisets $[\nu_i]_{i=1,\dots,d}$ and $[\nu_{1,i}]_{i=1,\dots,d}$ we deduce the multiset $[\nu_i + n_1\,\nu_{1,j}]_{i=1,\dots,d,j=1,\dots,d}$. Now, n_1 being "good", the multiset $[\nu_i + n_1\,\nu_{1,i}]_{i=1,\dots,d}$ (obtained by the Newton Polygon Algorithm applied to S_{1,n_1}) can be read as a submultiset of $[\nu_i + n_1\,\nu_{1,j}]_{i=1,\dots,d,j=1,\dots,d}$. This gives us the pairing between the multisets $[\nu_i]_{i=1,\dots,d}$ and $[\nu_{1,i}]_{i=1,\dots,d}$.
For example, assume that

$$[\nu_i]_{i=1,\dots,9} = 3[\alpha_1] + 4[\alpha_2] + 2[\alpha_3], \qquad [\nu_{1,i}]_{i=1,\dots,9} = 2[\beta_1] + 2[\beta_2] + 2[\beta_3] + 3[\beta_4]$$

and that the number 5 is good, i.e., the twelve values $\alpha_i + 5\beta_k$ are distinct. Computing the multiset $[\nu_i + 5\,\nu_{1,i}]_{i=1,\dots,9}$, we find, e.g.,

$$\begin{aligned}&[\alpha_1 + 5\beta_1] + 2[\alpha_1 + 5\beta_4] + [\alpha_2 + 5\beta_4] + 2[\alpha_2 + 5\beta_2] + \\ &\quad + [\alpha_2 + 5\beta_3] + [\alpha_3 + 5\beta_1] + [\alpha_3 + 5\beta_3],\end{aligned}$$

and we get the pairing

$$[(\alpha_1,\beta_1)] + 2[(\alpha_1,\beta_4)] + [(\alpha_2,\beta_4)] + 2[(\alpha_2,\beta_2)] + [(\alpha_2,\beta_3)] + [(\alpha_3,\beta_1)] + [(\alpha_3,\beta_3)]\ .$$

Comment: the multiset $[x_i]_{i=1,\dots,d}$ is, as a root multiset, made of "indiscernible elements". The knowledge of the multiset $[\nu_i]_{i=1,\dots,d}$ introduces some distinction between the roots (if the ν_i's are not all equal). The knowledge of the multiset $[\nu_i + n_1\,\nu_{1,i}]_{i=1,\dots,d}$ (with a "good" n_1) induces a finer distinction between the roots.

We remark that the case where some $Q_1(x_i)$'s equal zero can also be done correctly by a slight modification of the previous algorithm. Nevertheless, when such a case appears, it seems more natural to use the technique of dynamical evaluation (see [3] and section 2.1). If not all $Q_1(x_i)$'s equal zero (which is a trivial case), then one can compute a factorization of P in a product of two coprime polynomials P_1 and P_2 by applying algorithm **BasicD5** to P and Q_1. Then we can study separately the roots of these two polynomials. Moreover, the following steps of the algorithm are clearer if all $Q_1(x_i)$'s are distinct from zero.

Next we show that analogous arguments work for the general case. It will be sufficient to show how the case $r = 2$ works. Set $\nu_{2,i} = v(Q_2(x_i))$. We have computed the correct pairing $[(\nu_1,\nu_{1,1}),(\nu_2,\nu_{1,2}),\dots,(\nu_d,\nu_{1,d})]$ between the multisets $[\nu_i]_{i=1,\dots,d}$ and $[\nu_{1,i}]_{i=1,\dots,d}$. We know also a "good" integer n_1. We can assume w.l.o.g. that all $\nu_{1,i}$'s and $\nu_{2,i}$'s are finite. We compute first the multiset $[\nu_{2,i}]_{i=1,\dots,d}$. Let us call a *bad coincidence for* n_2 an equality

$$\nu_i + n_1\,\nu_{1,i} + n_2\,\nu_{2,h} = \nu_j + n_1\,\nu_{1,j} + n_2\,\nu_{2,k} \quad \text{with} \quad \nu_i + n_1\,\nu_{1,i} \neq \nu_j + n_1\,\nu_{1,j}\ .$$

If there is no bad coincidence for some n_2 then we can state this fact by considering the two sets $\{\nu_i + n_1\,\nu_{1,i}\ : i = 1,\dots,d\}$ and $\{\nu_{2,i}\ : i = 1,\dots,d\}$. We choose such an integer n_2. And so on. $\square$

Remark 2.8 Assume that P is a squarefree polynomial, so the x_i's are in the separable closure $\mathbf{K}^{\mathrm{sep}}$ of $(\mathbf{K}, \mathbf{V})$. Assume that algorithm **SimVal** has shown that some list of values $(\nu_i, \nu_{1,i}, \ldots, \nu_{r,i})$ corresponds to only one root of P. It is clear from the abstract definition of the henselization that such a "discernible" element over $(\mathbf{K}, \mathbf{V})$ is inside the henselization $\mathbf{K}^{\mathrm{h}}$ of $(\mathbf{K}, \mathbf{V})$. A perhaps surprising computational consequence is that, since the henselization is an immediate extension, when algorithm **SimVal** isolates (or discerns) some root of P, then the corresponding list of values is made only of "integer values", i.e., values of elements of $\mathbf{K}$ "without integer denominator". We can prove this constructively:
First, using computations in the henselization $\mathbf{K}^{\mathrm{h}}$ as defined in [5], one can prove (cf. [6]) the following lemma:

Lemma 2.9 *If the polynomial $P \in \mathbf{K}^{\mathrm{h}}[X]$ has roots $x_1, \ldots, x_d$ and if the d-tuple $[v(Q(x_1)), \ldots, v(Q(x_d))]$ (provided by* **SimVal** *applied to P, Q or by any other way) is equal to $d_1[\alpha_1] + \cdots + d_k[\alpha_k]$, with $\alpha_i \neq \alpha_j$ (for $i \neq j$), then one can factorize $P = P_1 \ldots P_k$ in $\mathbf{K}^{\mathrm{h}}[X]$ ($\deg P_i = d_i$), such that, if the roots of P_i are $y_1, \ldots, y_{d_i}$ the d_i-tuple $[v(Q(y_1), \ldots, v(Q(y_{d_i}))]$ is equal to $d_i[\alpha_i]$.*

Then if some list of values $(\nu_i, \nu_{1,i}, \ldots, \nu_{r,i})$ corresponds to only one root of P, we let
$n_0 = \#\{j : \nu_j = \nu_i\}$,
$n_1 = \#\{j : \nu_j = \nu_i \text{ and } \nu_{1,j} = \nu_{1,i}\}$,
$\ldots$
$n_r = \#\{j : \nu_j = \nu_i \text{ and } \nu_{k,j} = \nu_{k,i} \quad k = 1, \ldots, r\} = 1$

The previous result applied to $P(X)$ and $Q(X) = X$ provides a factor P_0 of P, with degree n_0; then applied to $P_0(X)$ and $Q_1(X)$, it provides a factor P_1 with degree n_1, and so on. Finally, we obtain a factor P_r of degree $n_r = 1$. So the corresponding root is in $\mathbf{K}^{\mathrm{h}}$. The computations in $\mathbf{K}^{\mathrm{h}}$ prove that the list of values is made only of "integer values"; one can compute explicitly elements of $\mathbf{K}$ having the same value. More precisely, one can compute $z_0, z_1, \ldots, z_r \in \mathbf{K}$, such that $x_i = z_0(1 + \nu_0)$, $Q_1(x_i) = z_1(1 + \nu_1), \ldots, Q_r(x_i) = z_r(1 + \nu_r)$, with $v(\nu_i) > 0$ for all i.

Root vectors of triangular systems. Algorithm **SimVal** says that "we can compute in $\mathbf{K}[x]$" where x is a root of P satisfying certain "compatible value conditions". We know how many roots of P correspond to a system of compatible value conditions. Computing in $\mathbf{K}[x]$ means that we can get "any brute information concerning the valuation in this field", more precisely, we can decide, for any new polynomial Q, if the value of $Q(x)$ is well determined or not. And we can compute the value(s). When several possibilities for $v(Q(x))$ appear, choosing one possible value, we refine our description of $\mathbf{K}[x]$.

So even if $\mathbf{K}[x]$ is not a priori a completely well determined valued field, we can nevertheless always do as if it was completely well determined. And we get recursively the following computations, exactly as in section 2.1.

More precisely, our computational problem is the following.

Computational problem 2.10
(computing in extensions generated by several successive algebraic elements)
Input:

- *A triangular system of polynomials $\overline{P} = (P_1, \ldots, P_n)$:*

$P_1(X_1) \in \mathbf{K}[X_1]$, $P_2(X_1, X_2) \in \mathbf{K}[X_1, X_2]$, $\ldots$, $P_k(X_1, \ldots, X_k) \in \mathbf{K}[X_1, \ldots, X_k]$.

- *A finite list of polynomials* $Q_1, \dots, Q_r$ *in* $\mathbf{K}[X_1, \dots, X_k]$.

Output:

- *The multiset of* $(k+r)$*-tuples of values*

$$[(v(x_1), \dots, v(x_k), v(Q_1(\overline{x})), \dots, v(Q_r(\overline{x})))]_{\overline{x}=(x_1,\dots,x_k)\in R}$$

where R *is the multiset of root vectors of* $\overline{P}$ *(this multiset has cardinality* $d = \prod_i \deg_{X_i}(P_i)$*).*

This problem is solved by the following algorithm.

Algorithm TriangularSimVal Use recursively algorithm **SimVal**. □

Graph of roots. The following algorithm can be seen as a particular case of the previous one. We denote by $\mu(P, a)$ the multiplicity of a as root of the univariate polynomial P (if $P(a) \neq 0$ we let $\mu(P, a) = 0$).

Computational problem 2.11
(computing the ultrametric graph of roots of a family of univariate polynomials)
Input:

- *A finite family of univariate polynomials* $\overline{P} = (P_1, \dots, P_s)$ *in* $\mathbf{K}[X]$.

Output:

- *The number* N *of distinct roots of* $P_1 \cdots P_n$.
- *For some ordering* $(x_1, \dots, x_N)$ *of these roots the finite family*

$$\left((\mu(P_i, x_j))_{i\in[1,s], j\in[1,N]}, (v(x_j - x_\ell))_{1\le j<\ell\le N}\right) .$$

Note that there are many possible answers, by changing the order of the roots. All correct answers are isomorphic.

Algorithm GraphRoots First a recursive use of **BasicD5** allows to find a finite multiset of pairwise coprime polynomials $(R_1, \dots, R_r)$ such that each P_i is a product of some R_k's. So we can assume w.l.o.g. that the P_i's are pairwise coprime. If $\deg(P_i) = n_i$ we introduce the roots $x_{i,1}, \dots, x_{i,n_i}$ of P_i through the triangular system

$$\begin{array}{rcl}
P_{i,1}(X_{i,1}) & = & P_i(X_{i,1}) \\[2ex]
P_{i,2}(X_{i,1}, X_{i,2}) & = & \frac{P_{i,1}(X_{i,2})-P_{i,1}(X_{i,1})}{X_{i,2}-X_{i,1}} \\[2ex]
P_{i,3}(X_{i,1}, X_{i,2}, X_{i,3}) & = & \frac{P_{i,2}(X_{i,1},X_{i,3})-P_{i,2}(X_{i,1},X_{i,2})}{X_{i,3}-X_{i,2}} \\
\vdots & \vdots & \vdots \\
P_{i,n_i}(X_{i,1}, \dots, X_{i,n_i}) & = & \frac{P_{i,n_i-1}(X_{i,1},\dots,X_{i,n_i-2},X_{i,n_i})-P_{i,n_i-1}(X_{i,1},\dots,X_{i,n_i-2},X_{i,n_i-1})}{X_{i,n_i}-X_{i,n_i-1}}
\end{array}$$

$$\begin{array}{rcl}
P_{i,1}(x_{i,1}) & = & 0 \\
P_{i,2}(x_{i,1}, x_{i,2}) & = & 0 \\
P_{i,3}(x_{i,1}, x_{i,2}, x_{i,3}) & = & 0 \\
\vdots & \vdots & \vdots \\
P_{i,n}(x_{i,1}, \dots, x_{i,n_i}) & = & 0
\end{array}$$

The $P_{i,k}$'s give all together a triangular system and we can apply **TriangularSimVal** for finding the values $v(x_{i,k} - x_{i',k'})$. We remark that we can use a simplified form of **TriangularSimVal** since all possible results are isomorphic and we need only one of these results. E.g., in the first step we compute the multiset $[(v(x_{1,k} - x_{1,k'})_{1 \leq k < k' \leq n_1}]$ but we select arbitrarily one value as the good one w.r.t. some ordering of the roots, and so on. □

Remark 2.12 There are probably some shortcuts allowing to give this ultrametric graph in a quicker way: for example, for a single polynomial, it is easy to compute the multiset of values $[v(x_i - x_j)]_{i \neq j}$ *without* knowing exactly to which edge each value corresponds; there might be a way (at least in a great number of cases) to reconstruct the graph (up to isomorphism).

3 Quantifier elimination

The aim of this section is to give a transparent proof of the following well known theorem (cf. [8]).

Theorem 3.1 *The theory of algebraically closed valued fileds (with fixed characteristics) admits quantifier elimination.*

First we give a sketch of the proof of this theorem. Our algorithm is a kind of "cylindric algebraic decomposition" (in the real closed case see, e.g., [1]). Given a finite set of multivariate polynomials, we choose a variable as being the main variable and we consider the other ones as parameters.

We settle in subsection 3.2 an existential decision procedure for a quantifier free formula with only one variable: given a finite set S of univariate polynomials, we give a complete description of the "valued line $\mathbf{K}^{\mathrm{ac}}$" w.r.t. S.

More precisely, we give first a formal name to each root of each polynomial in S, and we compute the ultrametric distance between each pair of these roots. We compute also the multiplicities of these roots and all the values $v(P_i(x_j))$ for each root x_j and each polynomial P_i. This job is done by algorithm **GraphRoots**.

Next, from these datas, we are able to test if a given conjunction of elementary assertions concerning the $v(P_i(\xi))$'s is realizable by some ξ of the line $\mathbf{K}^{\mathrm{ac}}$. In order to make this test we need a key geometric lemma, concerning *ultrametric graphs.* We explain this lemma in section 3.1.

The structure of our existential univariate decision procedure is very simple. This implies a kind of uniformity in such a way that the algorithm can be performed "with parameters", exactly as **BasicTriangularD5** is nothing but a parametrized version of **Basic D5**. This gives a good way for eliminating the quantifier in a formula with only one existential quantifier. So the work done in our final section 3.3 will be a careful verification of uniformity for the algorithms used in section 3.2.

Finally, the general elimination procedure follows by usual tricks.

We now give general explanations about notations and technical tools needed in the algorithms.

As in [8] we use a two-sorted language, $L = (L_F, L_\Gamma, v)$. The language of fields $L_F = \{0, 1, +, -, .\}$ is the F-sort. The language L_Γ is the Γ-sort. There is one more symbol, v, which is a function symbol for the valuation. The language L_Γ consists of the language $L'_\Gamma = \{0, \infty, +, -, <\}$ of ordered Abelian groups with last element ∞ together with a family of symbols $\{\frac{\cdot}{q} : q \in \mathbb{N}^*\}$.

By convention $a - \infty = 0$ for all $a \in \Gamma$. But there are some ambiguities as $a - (b - c)$ may not be equal to $a - b + c$. In fact, it is possible to avoid the sign $-$ for Γ-formulas, using case distinctions. For example, we can replace $a - b = c$ by $(b = \infty \wedge c = 0) \vee a = b + c$. So any quantifier free formula Φ is equivalent to a formula written without the Γ-sign $-$. In the sequel we assume w.l.o.g. that Γ-terms are always written without using the Γ-sign $-$.

Note also that we have no function symbol for the inverse of a nonzero element inside the field. This is not a restriction. The introduction of this function symbol would imply some trouble as the necessity of some strange convention as $x/0 = 0$ for any x.

The theory of algebraically closed non-trivial valued fields is $\mathbf{ACVF}(L)$. Recall that the formal theory specifies the characteristic of the field and of the residue field. In our formulas there are F-variables and Γ-variables, F-terms and Γ-terms, and, more important, F-quantifiers and Γ-quantifiers.

The rules of building terms are the natural ones. We see that the F-terms are formal polynomials in $\mathbb{Z}[x_1, \cdots, x_n]$. For the Γ-terms, we avoid the Γ-sign $-$. Take $r_1, \ldots, r_k \in \mathbb{Q}^{>0}$, and let $f_1, \ldots, f_\ell$ (with $\ell \leq k$) be F-terms; then

$$r_1 \cdot v(f_1) + \cdots + r_\ell \cdot v(f_\ell) + r_{\ell+1} \cdot a_{\ell+1} + \cdots + r_k \cdot a_k \tag{3}$$

(where each a_i is a Γ-variable or a Γ-constant) is a general Γ-term. Moreover we remark that such a Γ-term can be easily rewritten as

$$\frac{1}{N}\left(v(f) + s_{\ell+1} \cdot a_{\ell+1} + \cdots + s_k \cdot a_k\right)$$

where $N, s_j \in \mathbb{Z}^{>0}$.

When we want to make computations inside the algebraic closure of some explicitly given valued field $(\mathbf{K}, \mathbf{V})$ we have to use the theory $\mathbf{ACVF}(\mathbf{K}, \mathbf{V})$ where the elements of $\mathbf{K}$ and $\Gamma_{\mathbf{K}}$ are added as constants and the diagram of the valued field $(\mathbf{K}, \mathbf{V})$ is added as a set of axioms.

The theory $\mathbf{DOAG}_\infty$ of divisible ordered Abelian groups with last element ∞ admits quantifier elimination; hence it is sufficient to eliminate the F-quantifiers from an L-formula ϕ: we obtain an F-quantifier free L-formula ϕ' (most of the time, this formula has more Γ-quantifiers than ϕ), and we can conclude using the quantifier elimination of $\mathbf{DOAG}_\infty$.

This strategy allows us to get a new algorithmic proof of theorem 3.1, which is the topic of the third section of [8]: *The theory* $\mathbf{ACVF}(L)$ *admits quantifier elimination.*

3.1 Ultrametric Graphs. To prove theorem 3.1, we will need a lemma about *ultrametric graphs.* Let Γ be the divisible ordered Abelian group $\Gamma_{\mathbf{K}^{\mathrm{ac}}}$. A graph of vertices $p_1, \ldots, p_n$ is a subset G of $\{p_1, \ldots, p_n\}^2$ such that if $(p_i, p_j) \in G$, then $(p_j, p_i) \in G$. If $(p_i, p_j) \in G$, then it is an *edge* of G. The graph will be called *complete* if every pair (p_i, p_j) is an edge.

We consider graphs labeled by elements of $\Gamma \cup \{\infty\}$: to each edge (p_i, p_j) we associate an element $\varepsilon_{ij} \in \Gamma \cup \{\infty\}$, and we impose that $\varepsilon_{ij} = \varepsilon_{ji}$. Such a graph is called *ultrametric* if every triangle in it is an *ultrametric triangle*, that is, has two vertices labeled by the same element of Γ, and the third one is labeled by a greater or equal element. We can put $\varepsilon_{ii} = \infty$ as a convention, so that degenerated triangles are ultrametric.

If we define

$$t(\varepsilon_{ij}, \varepsilon_{ik}, \varepsilon_{jk}) \;:\Leftrightarrow\; (\varepsilon_{ij} = \varepsilon_{ik}) \wedge (\varepsilon_{ij} \le \varepsilon_{jk})\,,$$

then

$$T(\varepsilon_{ij}, \varepsilon_{ik}, \varepsilon_{jk}) \;:\Leftrightarrow\; t(\varepsilon_{ij}, \varepsilon_{ik}, \varepsilon_{jk}) \vee t(\varepsilon_{ik}, \varepsilon_{jk}, \varepsilon_{ij}) \vee t(\varepsilon_{jk}, \varepsilon_{ij}, \varepsilon_{ik})$$

is the formula asserting that (p_i, p_j, p_k) is an ultrametric triangle inside the graph G.

The complete graph of vertices $p_1, \ldots, p_n$ with edges labeled by ε_{ij} is ultrametric if the following formula is true:

$$\bigwedge_{i<j<k} T(\varepsilon_{ij}, \varepsilon_{ik}, \varepsilon_{jk})\,.$$

In an algebraically closed valued field, let $a_1, \ldots, a_n$ be fixed elements. Let $\varepsilon_{ij} = v(a_i - a_j)$. Then the complete graph of vertices $a_1, \ldots, a_n$ and of edges (a_i, a_j) labeled by ε_{ij} is ultrametric.

Lemma 3.2 (Ultrametric graphs) *In any formal theory of valued fields implying that the residue field is infinite, the assertion*

$$\exists_F\, x \bigwedge_{i=1,\ldots,n} v(x - a_i) = \beta_i$$

is equivalent to the formula expressing that the complete graph of vertices $a_1, \ldots, a_n$ and x, with edges (a_i, x) labeled by β_i, is ultrametric. The triangles (a_i, a_j, a_k) being ultrametric, this is equivalent to $\bigwedge_{i<j} T_{ij}$ where T_{ij} is $T(\varepsilon_{ij}, \beta_i, \beta_j)$.

Proof Let $S_i(x)$ be the formula $v(x - a_i) = \beta_i$. We prove that

$$\left(\exists_F\, x \bigwedge_i S_i(x)\right) \Longleftrightarrow \bigwedge_{i<j} T_{ij}\,.$$

The implication $\Longrightarrow$ is clear.
For the reverse implication $\Longleftarrow$, we first note that

$$(T_{ij} \wedge (\beta_j < \beta_i)) \Longrightarrow \beta_j = \varepsilon_{ij}\,,$$

and that

$$(\beta_j = \varepsilon_{ij} \wedge (\beta_j < \beta_i) \wedge S_i(x)) \Longrightarrow S_j(x)\,.$$

Thus we have the following implication:

$$(T_{ij} \wedge (\beta_j < \beta_i) \wedge S_i(x)) \Longrightarrow S_j(x)\,. \tag{4}$$

Hence we need to keep only those indices i for which β_i is maximal among $\beta_1, \ldots, \beta_n$. Let $\beta = \max\{\beta_1, \ldots, \beta_n\}$ and $I_1 = \{i \in \{1, \ldots, n\} : \beta_i = \beta\}$. Assume w.l.o.g. that $1 \in I_1$. We have

$$\bigwedge_{i<j} T_{ij} \wedge \bigwedge_{i\in I_1} S_i(x) \Longrightarrow \bigwedge_{1=1,\ldots,n} S_i(x)$$

Note that for $i, j \in I_1$, T_{ij} is equivalent to $\varepsilon_{ij} \ge \beta$, and that $S_i(x)$ is the formula $v(x - a_i) = \beta$. We show that

$$\bigwedge_{i<j,\ i,j\in I_1} T_{ij} \Longrightarrow \exists_F\, x \bigwedge_{i\in I_1} v(x - a_i) = \beta\,.$$

If $\beta = \infty$, we have $T_{ij} \Longrightarrow (a_i = a_j)$ for all $i, j \in I_1$, and in this case we take $x = a_i$ for any $i \in I_1$. Now assume that $\beta < \infty$. If $\varepsilon_{ij} > \beta$, we obtain $(S_i(x) \wedge T_{ij}) \Longrightarrow S_j(x)$. We consider the following case distinction:

• If $\varepsilon_{ij} > \beta$ for all $i, j \in I_1$ then $\left(\bigwedge_{i<j\ i,j\in I_1} T_{ij} \wedge S_1(x)\right) \implies \bigwedge_{i\in I_1} S_i(x)$. The formula $\exists_F\, x\ S_1(x)$ being always true, we have $\bigwedge_{i<j} T_{ij} \implies \exists_F\, x\ \bigwedge_i S_i(x)$.
• Else, we take in I_1 a subset I_2 which is maximal for the property that $\varepsilon_{ij} = \beta$ for all indices $i, j \in I_2$. It suffices to show that $\exists_F\, x \bigwedge_{i\in I_2} S_i(x)$, since from the definition of I_2 we have

$$\left(\bigwedge_{i<j,\ i,j\in I_1} T_{ij} \wedge \bigwedge_{i\in I_2} S_i(x)\right) \implies \bigwedge_{i\in I_1} S_i(x)\,.$$

We can assume w.l.o.g. that $1 \in I_2$. We denote the natural map from $\mathbf{V}^{\rm ac}$ to $\mathbf{V}^{\rm ac}/\mathcal{M}_{\mathbf{V}^{\rm ac}} = \overline{\mathbf{K}^{\rm ac}}$ by $x \mapsto \operatorname{res} x$. We fix $z \in \mathbf{K}^{\rm ac}$ such that $v(z) = \beta$. The field $\overline{\mathbf{K}^{\rm ac}}$ is infinite since it is algebraically closed; thus we can choose $x \in \mathbf{K}^{\rm ac}$ such that

$$\bigwedge_{i\in I_2} \operatorname{res}\left(\frac{x-a_1}{z}\right) \neq \operatorname{res}\left(\frac{a_i-a_1}{z}\right).$$

This x verifies $v(x-a_i) = \beta$, for all $i \in I_2$. This concludes the proof. $\square$

Remark 3.3 We can give a geometric description of the set

$$S = \{x \in \mathbf{K}^{\rm ac} : \bigwedge_{i=1,\ldots,n} v(x-a_i) = \beta_i\}\,.$$

We use the notations of the proof. Set $C_\beta(a) = \{x : v(x-a) = \beta\}$. We have

$$S = \bigcap_{i=1,\ldots,n} C_{\beta_i}(a_i) = \bigcap_{i\in I_1} C_\beta(a_i)\,.$$

If $\beta = \infty$, S is reduced to one element in K. Now suppose $\beta < \infty$. If $\varepsilon_{ij} > \beta$ for all $i, j \in I_1$, then $S = C_{\beta_i}(a_i)$ for all $i \in I_1$. If for some $i, j \in I_1$, $\varepsilon_{ij} = \beta$, take I_2 as in the proof. We have $S = \bigcap_{i\in I_2} C_\beta(a_i)$. Suppose that $1 \in I_2$. The set $C_\beta(a_1)$ is an infinite disjoint union of open disks $B^\circ_\beta(\zeta) = \{x : v(x-\zeta) > \beta\}$, where $v(\zeta - a_1) = \beta$. There is a bijection between the disks $B^\circ_\beta(\zeta)$ and the residue field of $\mathbf{K}^{\rm ac}$, given by

$$B^\circ_\beta(\zeta) \mapsto f(\zeta) = \operatorname{res}\left(\frac{\zeta - a_1}{z}\right).$$

We have the following equality:

$$S = \bigcap_{i\in I_2} C_\beta(a_i) = \bigcup_{\substack{v(\zeta-a_1)=\beta \\ \forall i\in I_2\setminus\{1\}\ f(\zeta)\neq f(a_i)}} B^\circ_\beta(\zeta)\,.$$

This union is nonempty because there are infinitely many values possible for $f(\zeta)$, but only finitely many for $f(a_i)$.

Remark 3.4 Another formulation of lemma 3.2 is that we have a quantifier elimination for *linear formulas* in $\mathbf{ACVF}(L)$: given a formula

$$\exists_F\, x\ \bigwedge_i v(x-x_i) = \beta_i$$

we put $\varepsilon_{ij} = v(x_i - x_j)$, and the above formula is equivalent to

$$\bigwedge_{i<j} T(\varepsilon_{ij}, \beta_i, \beta_j)\,.$$

An easy consequence is the following lemma:

Lemma 3.5 *Take any complete ultrametric graph of vertices $p_1, \ldots, p_n$, with edges labeled by $\varepsilon_{ij} \in \Gamma \cup \{\infty\}$, and elements $x_1, \ldots, x_l \in \mathbf{K}^{\mathrm{ac}}$ (with $l < n$), such that $v(x_i - x_j) = \varepsilon_{ij}$ for all $i, j \le l$. Then there exist $x_{l+1}, \ldots, x_n \in \mathbf{K}^{\mathrm{ac}}$ such that $v(x_i - x_j) = \varepsilon_{ij}$ for all i, j.*

3.2 Univariate existential decision procedure. We are going to prove that existential problems in a single variable x can be solved in $(\mathbf{K}^{\mathrm{ac}}, \mathbf{V}^{\mathrm{ac}})$.

Definition 3.6 *We define* univariate F-conditions *by*

(i) *For any $P(X) \in \mathbf{K}[X]$, the condition $\Phi(x) \;:\Leftrightarrow\; P(x) = 0$ is a univariate F-condition.*

(ii) *Take any $\gamma, \delta \in \Gamma_{\mathbf{K}}$, $q, r \in \mathbb{Q}^{>0}$, and any $P(X), Q(X) \in \mathbf{K}[X]$. The condition $\Phi(x) \;:\Leftrightarrow\; v(P(x)) + q \cdot \gamma \;\Box\; v(Q(x)) + r \cdot \delta$, where $\Box$ is either $=$ or $<$, is a univariate F-condition.*

(iii) *Take any $P(X) \in \mathbf{K}[X]$. The condition $\Phi(x) \;:\Leftrightarrow\; v(P(x)) < \infty$ is a univariate F-condition.*

(iv) *If $\Phi(x), \Psi(x)$ are univariate F-conditions, then $\Phi(x) \wedge \Psi(x)$ and $\Phi(x) \vee \Psi(x)$ are univariate F-conditions.*

Conditions of the form **(i)**, **(ii)** *and* **(iii)** *are called* atomic F-conditions.

Definition 3.7 *We define* Γ-conditions *by*

(i) *For any $\delta \in \Gamma_{\mathbf{K}} \cup \{\infty\}$, $q_1, \ldots, q_n \in \mathbb{Q}^{>0}$, $r \in \{1, n\}$, the condition $\Phi(\overline{a}) \;:\Leftrightarrow\; q_1 \cdot a_1 + \cdots + q_r \cdot a_r \;\Box\; q_{r+1} \cdot a_{r+1} + \cdots + q_n \cdot a_n + \delta$, where $\Box$ is either $=$, $>$ or $<$, is a Γ-condition on $\overline{a}$.*

(ii) *If $\Phi(\overline{a}), \Psi(\overline{a})$ are Γ-conditions on $\overline{a}$, then so are $\Phi(\overline{a}) \wedge \Psi(\overline{a})$ and $\Phi(\overline{a}) \vee \Psi(\overline{a})$.*

Conditions of the form **(i)** *are called* atomic Γ-conditions.

It is well known that such conditions are equivalent to some condition of the following form, which is by definition a *disjunctive normal form*:

$$\bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} \Phi_{ij} \,,$$

where the Φ_{ij} are atomic conditions. Moreover, given any univariate condition $\Phi(x)$, there is an algorithm which computes a disjunctive normal form for $\Phi(x)$.

We say that $\xi \in \mathbf{K}^{\mathrm{ac}}$ satisfies a univariate F-condition $\Phi(x)$ if $\Phi(\xi)$ holds in $\mathbf{K}^{\mathrm{ac}}$, and that $\alpha_1, \ldots, \alpha_n \in \Gamma_{\mathbf{K}^{\mathrm{ac}}}$ satisfy a Γ-condition $\Phi(\overline{a})$ if $\Phi(\overline{\alpha})$ holds in $\Gamma_{\mathbf{K}^{\mathrm{ac}}}$.

We recall the following result without proof. See Theorem 5.6 in [2] or Corollary 3.1.17 in [7].

Proposition 3.8 (Existential Decision Procedure in $\mathbf{DOAG}_\infty$) *Let $\Phi(\overline{a})$ be a Γ-condition. Then there is an algorithm to decide whether there are some $\alpha_1, \ldots, \alpha_n \in \Gamma_{\mathbf{K}^{\mathrm{ac}}}$ satisfying $\Phi(\overline{a})$ or not. If the answer is yes, the algorithm provides such a n-tuple. We call it a* witness *of the condition.*

We now prove the following theorem:

Theorem 3.9 (Univariate Existential Decision Procedure in ACVF) *Let $\Phi(x)$ be a univariate condition. Then we have an algorithm to decide whether there is some $\xi \in \mathbf{K}^{\mathrm{ac}}$ satisfying $\Phi(x)$ or not. If the answer is yes, the algorithm gives a description of a witness $\xi \in \mathbf{K}^{\mathrm{ac}}$ such that $\Phi(\xi)$ holds; the algorithm decides whether ξ is unique or not, and if this is the case then ξ is in $\mathbf{K}^{\mathrm{h}}$.*

Proof We give an existential decision procedure for a conjunction

$$\Phi(x) \ : \ \bigwedge_{i=1}^{n} \Phi_i(x)$$

where the Φ_i's are atomic conditions. It suffices to use it several times to obtain an existential decision procedure for a univariate condition put in a disjunctive normal form, and hence for every univariate condition.

• **First case:** One of the $\Phi_i(x)$ (let's say $\Phi_1(x)$) is of the form $P(x) = 0$. Let $k = \deg P$, and $\xi_1, \ldots, \xi_k$ be the roots of P. Let $Q_1(x), \ldots, Q_r(x) \in K[x]$ be the polynomials appearing in the other $\Phi_i(x)$'s. We can use **SimVal** to obtain the multiset of $(r+1)$-tuples of values $[(v(\xi_i), v(Q_1(\xi_i)), \ldots, v(Q_r(\xi_i)))]_{i=1,\ldots,k}$.

It suffices now to check, for each $(\nu, \nu_1, \ldots, \nu_r)$ in this list, whether the conditions $\Phi_1, \ldots, \Phi_n$ are verified:

- for a Φ_k of the form $Q_i(x) = 0$, test whether $\nu_i = \infty$,
- for a Φ_k of the form $v(Q_i(x)) + q\cdot\gamma \,\square\, v(Q_j(x)) + r\cdot\delta$, test whether $\nu_i + q\cdot\gamma \,\square\, \nu_j + r\cdot\delta$ (where $\square$ is either $=$, $>$ or $<$).
- for a Φ_k of the form $v(Q_i(x)) < \infty$, test whether $\nu_i < \infty$.

If there are no $(r+1)$-tuples in this multiset such that these conditions are verified, then there is no $\xi \in \mathbf{K}^{\mathrm{ac}}$ satisfying $\Phi(x)$; if there are $m \leq k$ of these multisets satisfying these conditions, we know that m of the roots of P can be chosen for ξ.

If $m = 1$, then remark 2.8 shows that the corresponding root of P is in $\mathbf{K}^{\mathrm{h}}$.

• **Second case:** Assume now that there is no condition $\Phi_i(x)$ of the form $P(x) = 0$ among the $\Phi_i(x)$. For each i, let $P_i(x)$ and $Q_i(x)$ be the polynomials appearing in atomic formulas $\Phi_i \ : \ v(P_i(x)) + q_i\cdot\gamma_i \,\square_i\, v(Q_i(x)) + r_i\cdot\delta_i$ (where $\square_i$ is either $=$, $>$ or $<$), and $\Phi_i \ : \ v(P_i(x)) < \infty$ (in that case, set $Q_i = 1$, $q_i = r_1 = 1$, $\gamma_i = 0$ and $\delta_i = \infty$ for the sequel).

We construct the following formulas:

$$\Phi'(x, \overline{c}, \overline{d}) \ : \ \left(\bigwedge_{i=1}^{n} v(P_i(x)) = c_i \wedge v(Q_i(x)) = d_i \right)$$

$$\Phi''(\overline{c}, \overline{d}) \ : \ \left(\bigwedge_{i=1}^{n} c_i + q_i\cdot\gamma_i \,\square_i\, d_i + r_i\cdot\delta_i \right) .$$

The variables $\overline{c} = c_1, \ldots, c_n$ and $\overline{d} = d_1, \ldots, d_n$ stand for elements of $\Gamma_{\mathbf{K}^{\mathrm{ac}}}$. We have

$$\exists x \in \mathbf{K}^{\mathrm{ac}} \ \Phi(x) \Longleftrightarrow \exists \overline{c}, \overline{d} \in \Gamma_{\mathbf{K}^{\mathrm{ac}}} \, \exists x \in \mathbf{K}^{\mathrm{ac}} \ \Phi'(x, \overline{c}, \overline{d}) \wedge \Phi''(\overline{c}, \overline{d}) \, .$$

Consider a problem of the following form:

$$\Psi(x, \overline{b}) \ : \ \exists x \in \mathbf{K}^{\mathrm{ac}} \ \bigwedge_{i=1}^{m} v(R_i(x)) = b_i \, ,$$

where each $R_i(X)$ is a polynomial of $\mathbf{K}[x]$, and the b_i's are indeterminates.

We introduce all the roots $r_1, \ldots, r_N$ of the polynomials $R_1, \ldots, R_m$. We can compute N with the algorithm **GraphRoots**, as well as the values $\varepsilon_{ij} = v(r_i - r_j)$, for all i, j, and the multiplicity μ_{jk} of r_k as a root of R_j. We have an equivalence

$$\Psi(x, \overline{b}) \Longleftrightarrow \exists \, x \in \mathbf{K}^{\mathrm{ac}} \, \exists a_1 \cdots a_N \in \Gamma_{\mathbf{K}^{\mathrm{ac}}} \ \bigwedge_{i=1}^{N} v(x - r_i) = a_i \wedge \Psi_1(\overline{a}, \overline{b}) \, ,$$

where Ψ_1 is a conjuction of formulas of the form $b_j = \sum_k \mu_{jk} \cdot a_k$.

From the ultrametric graph lemma we have

$$\exists x \in \mathbf{K}^{\mathrm{ac}} \bigwedge_{i=1}^{N} v(x - r_i) = a_i \Longleftrightarrow \bigwedge_{i<j} T(\varepsilon_{ij}, a_i, a_j) \; .$$

Hence we can write that $\Psi(x, \overline{b})$ is equivalent to a problem in $\Gamma_{\mathbf{K}^{\mathrm{ac}}}$:

$$\Psi(x, \overline{b}) \Longleftrightarrow \exists \overline{a} \in \Gamma_{\mathbf{K}^{\mathrm{ac}}} \bigwedge_{i<j} T(\varepsilon_{ij}, a_i, a_j) \wedge \Psi_1(\overline{a}, \overline{b}) \; .$$

Now we can do that for $\Psi = \Phi'$. We obtain that $\exists x \in \mathbf{K}^{\mathrm{ac}}\, \Phi'(x, \overline{c}, \overline{d})$ is equivalent to $\exists \overline{a} \in \Gamma_{\mathbf{K}^{\mathrm{ac}}}\, \Phi'''(\overline{a}, \overline{c}, \overline{d})$, where $\Phi'''(\overline{a}, \overline{c}, \overline{d})$ is a Γ-condition. We have proved

$$\exists x \in \mathbf{K}^{\mathrm{ac}}\, \Phi(x) \Longleftrightarrow \exists \overline{a}, \overline{c}, \overline{d} \in \Gamma_{\mathbf{K}^{\mathrm{ac}}}\, \Phi'''(\overline{a}, \overline{c}, \overline{d}) \wedge \Phi''(\overline{c}, \overline{d}) \; .$$

We can apply the existential decision procedure for $\mathbf{DOAG}_\infty$ to this formula. If there is no solution, then there is no $\xi \in \mathbf{K}^{\mathrm{ac}}$ satisfying $\Phi(x)$. If there is a solution, we can use it together with lemma 3.2 to describe an element $\xi \in \mathbf{K}^{\mathrm{ac}}$ satisfying $\Phi(x)$. Of course, there is no unicity in that case. □

Remark 3.10 The first case of our proof can in fact be treated as a particular case of the second, replacing $P(x) = 0$ by $v(P(x)) = \infty$: in that case the existential decision procedure in $\mathbf{DOAG}_\infty$ will give $a_i = \infty$ for some i, and then $v(x - r_i) = \infty$ implies $\xi = r_i$. However, the proof is clearer with this distinction. Moreover, it would be less easy to show that in the case of unicity, the witness is in $\mathbf{K}^{\mathrm{h}}$.

3.3 Quantifier Elimination. Quantifier elimination algorithms very often come from existential decision procedures in the one variable case. If such a decision procedure is "uniform" it can be performed "with parameters". This gives a good way for eliminating the quantifier in a formula with only one existential quantifier. For the real algebraic case see, e.g., [1] chapter 1. In the present section, we will treat the case of algebraically closed valued fields.

Definition 3.11 *Take $n \in \mathbb{N}$, and denote by $\overline{y}$ an n-tuple $(y_1, \ldots, y_n)$ of F-variables. Let $C_1(\overline{y}), \ldots, C_m(\overline{y})$ be atomic L-formulas with $y_1, \ldots, y_n$ as the only free variables.*
1. *We say that $\bigvee_i C_i(\overline{y})$ is a* finite exclusive disjunction *if*

$$\forall_F \overline{y} \quad \bigvee_{i=1}^{m} C_i(\overline{y}) \;\; \wedge \;\; \bigwedge_{i \neq j} \neg C_i(\overline{y}) \vee \neg C_j(\overline{y})$$

holds. In that case we write

$$\mathfrak{C}_i = \{\overline{y} \in \mathbf{K}^n : C_i(\overline{y})\} \; .$$

Then K^n is the disjoint union of $\mathfrak{C}_1, \ldots, \mathfrak{C}_m$. The family $\mathfrak{C}_i$ is a definable partition *of the space $\mathbf{K}^n$. Note that we allow that some $\mathfrak{C}_i$ may be empty.*
2. *Let $D_{ij}(\overline{y})$, for $i = 1, \ldots, m$ and $j = 1, \ldots, \ell_i$, be atomic L-formulas such that $\bigvee_{ij} D_{ij}(\overline{y})$ is a finite exclusive disjunction. We say that $\bigvee_{ij} D_{ij}$ is a refinement of $\bigvee_i C_i$ if for all i, we have*

$$C_i(\overline{y}) \Longleftrightarrow \bigvee_{j=1}^{\ell_i} D_{ij}(\overline{y}) \; ,$$

or, equivalently

$$\mathfrak{C}_i = \bigcup_{j=1}^{\ell_i} \mathfrak{D}_{ij} ,$$

where $\mathfrak{D}_{ij} = \{\overline{y} \in \mathbf{K}^n : D_{ij}(\overline{y})\}$. *Note that this union is a disjoint union.*

We denote by $\overline{Y}$ an n-tuple of indeterminates $Y_1, \ldots, Y_n$. The ring $\mathbf{K}\left[\overline{Y}\right]$ is $\mathbf{K}[Y_1, \ldots, Y_n]$. We can apply the algorithms given in the previous section to polynomials with parameters. Consider $P(\overline{Y}, X) \in \mathbf{K}\left[\overline{Y}, X\right]$ as a polynomial in X with parameters $\overline{Y}$.

Proposition 3.12 (Algorithms with parameters)
1. *The Newton Polygon Algorithm applied to* P *provides*

- **(i)** *a finite exclusive disjunction* $\bigvee_i C_i(\overline{y})$,
- **(ii)** *for each* i, *an integer* k_i *and a multiset* $[t_1(\overline{y}), \ldots, t_{k_i}(\overline{y})]$, *where each* $t_j(\overline{y})$ *is an* L_Γ*-term,*

such that for all $\overline{y} \in \mathfrak{C}_i$, $k_i = \deg_X P(\overline{y}, X)$, *and if* $[\xi_1, \ldots \xi_{k_i}]$ *denotes the multiset of roots of* $P(\overline{y}, X)$, *then* $[t_1(\overline{y}), \ldots, t_{k_i}(\overline{y})]$ *is* $[v(\xi_1), \ldots, v(\xi_{k_i})]$. *In other words, in each case of the above exclusive disjunction, the algorithm computes the values of the roots of* $P(\overline{y}, X)$.
2. *Keep the notation of the previous statement. Let* $Q_1, \ldots, Q_r \in \mathbf{K}\left[\overline{Y}, X\right]$ *be polynomials in* X *with parameters* $\overline{Y}$. *The algorithm* **SimVal** *applied to* $P, Q_1, \ldots, Q_r$ *provides*

- **(i)** *a refinement* $\bigvee_{ij} D_{ij}(\overline{y})$ *of* $\bigvee_i C_i(\overline{y})$,
- **(ii)** *for each case* i, j *(with* $j \in \{1, \ldots, \ell_i\}$*) a multiset of* $(r+1)$*-tuples of* L_Γ*-terms* $\left[\left(t_s(\overline{y}), u_s^1(\overline{y}), \ldots, u_s^r(\overline{y})\right)\right]_{s=1,\ldots,k_i}$,

such that for all $\overline{y} \in \mathfrak{D}_{ij}$, *if* $[\xi_1, \ldots \xi_{k_i}]$ *is the multiset of roots of* $P(\overline{y}, X)$, *then* $\left[\left(t_s(\overline{y}), u_s^1(\overline{y}), \ldots, u_s^r(\overline{y})\right)\right]_{s=1,\ldots,k_i}$ *is* $[(v(\xi_s), v(Q_1(\xi_s), \ldots, v(Q_r(\xi_s)))]_{s=1,\ldots,k_i}$.
3. *Take* $P_1, \ldots, P_s \in \mathbf{K}\left[\overline{Y}, X\right]$. *The algorithm* **GraphRoots** *applied to* $P_1, \ldots, P_s$ *provides*

- **(i)** *a finite exclusive disjunction* $\bigvee_i C_i(\overline{y})$,
- **(ii)** *for each* i, *an integer* N_i *and a finite family* $\left((\mu_{jk})_{j\in[1,s],k\in[1,N_i]}, (t_{k,\ell}(\overline{y}))_{1\le j<\ell\le N}\right)$, *where the* μ_{jk} *are integers and the* $t_{k,\ell}(\overline{y})$ *are* L_Γ*-terms,*

such that for all $\overline{y} \in \mathfrak{C}_i$, N_i *is the number of roots of* $P_1 \cdot \cdots \cdot P_s$, *and for some ordering* $(\xi_1, \ldots, \xi_{N_i})$ *of these roots,* μ_{jk} *is the multiplicity of* ξ_k *as a root of* P_j, *and* $t_{k,\ell}(\overline{y})$ *is* $v(\xi_k - \xi_\ell)$.

Proof For the first statement, write $P(\overline{Y}, X) = q_n(\overline{y}) \cdot X^n + \cdots + q_0(\overline{y})$. Consider the exclusive disjunction

$$\Big(q_0(\overline{y}) = \ldots = q_n(\overline{y}) = 0\Big) \ \vee \ \bigvee_{i=0}^{n} \left(v(q_i(\overline{y})) < \infty \ \wedge \ \bigwedge_{j=i+1}^{n} q_{n-j}(\overline{y}) = 0 \right) .$$

In each case of this disjunction the degree in X of $P(\overline{y}, X)$ is fixed. We are going to refine it to obtain the desired disjunction. Apply the Newton Polygon Algorithm in any fixed case of this disjunction: its result depends naturally on a new disjunction, each case of it expressing a different shape for the Newton Polygon of P. More precisely, if $m > 0$ is $\deg_X P(\overline{y}, X)$, for each $\ell \le m + 1$ and each ℓ-tuple

$(k_1, \ldots, k_\ell)$ of non-negative integers such that $0 = k_1 < \cdots < k_\ell = m$, we can write a formula $C_{m,\ell,k_1,\ldots,k_\ell}(\overline{y})$ expressing that $(k_1, v(q_{k_1}(\overline{y}))), \ldots, (k_\ell, v(q_{k_e ll}(\overline{y})))$ are the consecutive vertices of the Newton Polygon of P. In each fixed case $C_{m,\ell,k_1,\ldots,k_\ell}$, the values of the roots are the L_Γ-terms $\frac{1}{k_{i+1}-k_i}(v(q_{k_i}(\overline{y})) - v(q_{k_{i+1}}(\overline{y})))$.

Example: Set $R(\overline{Y}, X) = a(\overline{Y})X^2 + b(\overline{Y})X + c(\overline{Y})$; we omit the parameters $\overline{Y}$ in the sequel: a stands for $a(\overline{Y})$, and so on.

- If $v(a) < \infty$, and $2v(b) \geq v(a) + v(c)$, then $\xi_1, \xi_2 \in \mathbf{K}^{\mathrm{ac}}$, the roots of R considered as a polynomial in X, both have value $\frac{1}{2}(v(c) - v(a))$.
- If $v(a) < \infty$, and $2v(b) < v(a) + v(c)$, then there is one root of value $v(b) - v(a)$ and the other of value $v(c) - v(b)$.
- If $a = 0$ and $v(b) < \infty$, then there is a single root, of value $v(c) - v(b)$.
- If $a = 0$ and $b = 0$ and $v(c) < \infty$, then there is no root.
- If $a = b = c = 0$, then $\forall x \in \mathbf{K}^{\mathrm{ac}}$, $R(\overline{y}, x) = 0$.

Now we turn to the second statement. The algorithm **SimVal** applies the Newton Polygon Algorithm to P: this is our first disjunction. Then it computes some Tschirnhaus transformation of P. The degree of P being fixed in each case of the disjunction, this can be done without refining it. The results of this computations are new polynomials in $K\left[\overline{Y}, X\right]$. We apply the Newton Polygon Algorithm to each of these polynomials, after refining the disjunction. We obtain some lists of L_Γ-terms, from which we can construct the list we want, under a few conditions to eliminate "bad coincidences" (cf. 2.7); these conditions give rise to a new refinement of the disjunction.

For the third statement, just note that **GraphRoots** uses **SimVal** iteratedly; then the result comes from the second statement. □

Now we are able to prove theorem 3.1.

Proof of theorem 3.1 We recall that there are classical and easy arguments ([8]) showing that it suffices to eliminate an F-quantifier $\exists_F\, x$ in a formula such as $\exists_F\, x\ \bigwedge_{k=1,\ldots,n} \Phi_k(\overline{y}, x)$, where each $\Phi_k(\overline{y}, x)$ is either an atomic F-formula like $P(\overline{y}, x) = 0$ with $P(\overline{y}, x) \in \mathbb{Z}[\overline{y}, x]$, or an atomic Γ-formula. Note that an atomic F-formula $P(\overline{y}, x) \neq 0$ can be replaced by the Γ-formula $v(P(\overline{y}, x)) < \infty$. So we are done if we prove the following proposition. □

Proposition 3.13 *There is an algorithmic procedure that computes, from a formula $\exists_F\, x\ \bigwedge_{k=1,\ldots,n} \Phi_k(\overline{y}, x)$ (where each $\Phi_k(\overline{y}, x)$ is either an atomic F-formula like $P(\overline{y}, x) = 0$ with $P(\overline{y}, x) \in \mathbb{Z}[\overline{y}, x]$, or an atomic Γ-formula), an equivalent quantifier free formula $\Psi(\overline{y})$.*

A geometric form of this proposition is the following (for the real algebraic case see, e.g., theorem 2.2.1 of [1]). Let $\mathbf{K}$ be a subfield of $\mathbf{L}$. A *basic v-constructible set defined over* $\mathbf{K}$ in $\mathbf{L}^n$ is a set of the form $\{\overline{x} \in \mathbf{L}^n : \Phi(\overline{x})\}$ where $\Phi(\overline{x})$ is either an atomic F-formula like $P(\overline{x}) = 0$ with $P(\overline{x}) \in \mathbf{K}[\overline{x}]$, or an atomic Γ-formula (which is built by using only constants in $\mathbf{K}$ and $v(\mathbf{K})$). A *v-constructible set defined over* $\mathbf{K}$ in $\mathbf{L}^n$ is any boolean combination of basic v-constructible sets defined over $\mathbf{K}$.

Proposition 3.14 *Let $\mathbf{L}$ be an algebraically closed valued field, and $\mathbf{K}$ a subfield. Then the image $\pi(S)$ of a v-constructible set S defined over $\mathbf{K}$ under the canonical projection from $\mathbf{L}^n$ onto $\mathbf{L}^{n-1}$ is again a v-constructible set defined over*

K. *Moreover, there is an algorithmic procedure that uses only computations inside* **K** *to get a description of* $\pi(S)$ *from a description of* S.

Proof of proposition 3.13 We can apply our univariate decision procedure (theorem 3.9) with parameters in order to eliminate x. This procedure uses **Sim-Val** and **GraphRoots** with parameters: it will provide an exclusive disjunction $\bigvee_i C_i(\overline{y})$, and in each case of this exclusive disjunction, a formula $\Psi_i(\overline{y})$ without F-quantifiers (but perhaps with some new Γ-quantifiers if for $\overline{y} \in \mathfrak{C}_i$ we are in the second case of the proof of 3.9) such that

$$\forall \overline{y} \in \mathfrak{C}_i,\ \exists_F\, x \bigwedge_{k=1,\ldots,n} \Phi_k(\overline{y}, x) \Longleftrightarrow \Psi_i(\overline{y})\ .$$

Thus we have

$$\exists_F\, x \bigwedge_{k=1,\ldots,n} \Phi_k(\overline{y}, x) \Longleftrightarrow \bigvee_i C_i(\overline{y}) \wedge \Psi_i(\overline{y})\ .$$

This concludes the proof. □

Remark 3.15 The strategy used in [8] was first to give an elimination for linear formulas, and then a procedure which decreases the degrees of polynomials. There was no geometric idea at first sight, although there may be a geometric content hidden in the proof. We believe that the two procedures are in fact different.

When we use this quantifier elimination with the theory **ACVF**(**K**, **V**) we get as a particular case a decision procedure for a closed formula with coefficients in a valued field **K** given as in the introduction.

Theorem 3.16 *Take a formula*

$$\Theta(\overline{y})\ :\ Q_F^1 x_1 \ldots Q_F^n x_n\ \Phi(\overline{\alpha}, \overline{y}, \overline{x})$$

where each Q_F^i *is* $\forall_F$ *or* $\exists_F$ *and* $\overline{\alpha} = \alpha_1, \ldots, \alpha_m$ *are elements of* **K**. *We have an algorithm for computing a quantifier free formula* $\Psi(\overline{y})$ *equivalent to* $\Theta(\overline{y})$. *As a particular case, when* $\overline{y}$ *is the empty sequence, we can decide whether the formula* $\Theta(\overline{y})$ *is true in* $\mathbf{K}^{\mathrm{ac}}$ *or not. Moreover, if the formula is purely existential, i.e.,* $Q_F^1, \ldots, Q_F^n$ *are existential quantifiers* $\exists_F$, *then the algorithm provides a witness* $\overline{\xi} \in (\mathbf{K}^{\mathrm{ac}})^n$ *such that* $\Phi(\overline{\xi})$ *is true. If we have a result of unicity such as*

$$\forall_F \overline{x}, \overline{y}\ (\Phi(\overline{\alpha}, \overline{x}) \wedge (\Phi(\overline{\alpha}, \overline{y}) \Longrightarrow \overline{x} = \overline{y})\ ,$$

then this witness is in $(\mathbf{K}^{\mathrm{h}})^n$.

Proof Let us explain how we get the test point. We apply the quantifier elimination procedure to

$$Q_F^1 x_1 \ldots Q_F^n x_n\ \Phi(\overline{a}, \overline{x})$$

obtained after replacement of each α_i by a new indeterminate a_i. The result is a quantifier-free formula $\Psi(\overline{a})$, such that

$$Q_F^1 x_1 \ldots Q_F^n x_n\ \Phi(\overline{a}, \overline{x}) \Longleftrightarrow \Psi(\overline{a})\ .$$

It suffices to test whether $\Psi(\overline{\alpha})$ is true or not.

If all quantifiers Q_F^i are existential, we can find formulas $\Psi_k(\overline{a}, x_1, \ldots, x_k)$ for $k = 1$ to $n-1$, such that

$$\begin{array}{cc} & \exists_F\, x_1 \ldots \exists_F\, x_n\ \Phi(\overline{a}, x_1, \ldots, x_n) \\ \Longleftrightarrow & \exists_F\, x_1 \ldots \exists_F\, x_{n-1}\ \Psi_{n-1}(\overline{a}, x_1, \ldots, x_{n-1}) \\ \vdots & \vdots \\ \Longleftrightarrow & \exists_F\, x_1\ \Psi_1(\overline{a}, x_1) \\ \Longleftrightarrow & \Psi(\overline{a}) \end{array}$$

If $\Psi(\overline{\alpha})$ is true and we apply the decision procedure of theorem 3.9 to the sentence $\exists_F\, x_1\ \Psi_1(\overline{\alpha}, x_1)$, we find $\xi_1 \in \mathbf{K}^{\mathrm{ac}}$ such that $\Psi_1(\overline{\alpha}, \xi_1)$ holds. We apply again the decision procedure to $\exists_F\, x_2\ \Psi_2(\overline{\alpha}, \xi_1, x_2)$ and we find $\xi_2 \in \mathbf{K}^{\mathrm{ac}}$ such that $\Psi_2(\overline{\alpha}, \xi_1, \xi_2)$ holds, and so on. In this way, we find $\xi_1, \ldots, \xi_n \in \mathbf{K}^{\mathrm{ac}}$ such that $\Phi(\overline{\alpha}, \xi_1, \ldots, \xi_n)$ holds.

If the n-tuple $(\xi_1, \ldots, \xi_n)$ satisfying $\Phi(\overline{\alpha}, x_1, \ldots, x_n)$ is unique, then ξ_1 satisfying $\Psi_1(\overline{\alpha}, x_1)$ is unique and theorem 3.9 shows that $\xi_1 \in \mathbf{K}^{\mathrm{h}}$. Repeating this argument n times, we conclude that, in this case, $\xi_1, \ldots, \xi_n \in \mathbf{K}^{\mathrm{h}}$. □

References

[1] J. Bochnak, M. Coste, M.-F. Roy: *Géométrie algébrique réelle*, Springer (1987)

[2] M. Coste, H. Lombardi, M.-F. Roy: *Dynamical method in algebra: Effective Nullstellensätze*, Annals of Pure and Applied Logic **111** (2001), 203–256

[3] J. Della Dora, C. Dicrescenzo, D. Duval: *About a new method for computing in algebraic number fields*, in: Proceedings Eurocal'85, Springer Lecture Notes in Computer Science **204** (1985), 289–290

[4] C. Dicrescenzo, D. Duval: *Algebraic extensions and algebraic closure in Scratchpad*, in: Symbolic and algebraic computation (ISSAC 88), Springer Lecture Notes in Computer Science **358** (1989), 440–446

[5] F.-V. Kuhlmann, H. Lombardi: *Construction du hensélisé d'un corps valué*, Journal of Algebra **228** (2000), 624–632

[6] H. Perdry: *Aspects constructifs de la théorie des corps valués*, Thèse de doctorat en Mathématiques et Applications de l'Université de Franche-Comté (2001)

[7] D. Marker: *Model Theory: An Introduction*, Graduate Texts in Mathematics **217**, Springer (2002)

[8] V. Weispfenning: *Quantifier elimination and decision procedure for valued fields*, in: Models and sets, Springer Lecture Notes in Math. **1103** (1984), 419–472