On Triangular Decompositions of Algebraic Varieties

Marc Moreno Maza [1]

Ontario Research Center for Computer Algebra, University of Western Ontario, Canada

Abstract

We propose an efficient algorithm for computing triangular decompositions of algebraic varieties. It is based on an incremental process and produces components in order of decreasing dimension. The combination of these two major features is obtained by means of lazy evaluation techniques and a lifting property for calculations modulo regular chains. This allows a good management of the intermediate computations, as confirmed by several implementations and applications of this work. Our algorithm is also well suited for parallel execution.

Introduction

Triangular decompositions are one of the major tools for solving polynomial systems. For systems of algebraic equations, they provide a convenient way to describe complex solutions and a step towards isolation of real roots or decomposition into irreducible components. Combined with other techniques, they are used for these purposes by several computer algebra systems such as AXIOM, ALDOR, MAPLE and SINGULAR. For systems of partial differential equations, they provide the only practicable way for determining a symbolic solution. Moreover, thanks to Rosenfeld's Lemma, triangulation methods for the algebraic case apply to the differential one.

In this paper, we propose an innovative algorithm for computing triangular decompositions of algebraic varieties. It is based on an incremental process and produces components in order of decreasing dimension. The combination of these two major features is obtained by means of lazy evaluation techniques and a lifting property for calculations modulo regular chains. This allows a

[1] moreno@orcca.on.ca

good management of the intermediate computations. In particular, redundant branches can be cut at an early stage of the decomposition process. The efficiency of this approach is confirmed by several implementations and applications of this work. In addition, our algorithm relies on a paradigm of task management and is well suited for parallel execution.

Triangular decompositions were introduced by Ritt (1932) for solving systems of partial differential equations based on the concept of a characteristic set of a polynomial ideal. Ritt's decompositions involved only characteristic sets of prime ideals (Ritt, 1950). In this case, they have good properties and Aubry et al. (1999) proved that a characteristic set of a prime ideal is no harder to compute than a lexicographical Gröbner basis. The first complexity estimates for the computation of a characteristic set and a triangular decomposition were given by Gallo and Mishra (1990) and Szántó (1999).

Following the work of Ritt, Wu (1986) proposed an algorithm for solving systems of algebraic equations by means of characteristic sets of not necessarily prime ideals. Wu's method was used for geometric problems where degenerate solutions are not interesting, but his decompositions may contain inconsistent components, and thus redundant components.

The consistency problem was solved by Kalkbrener (1991) who considered particular characteristic sets, called regular chains. Aubry et al. (1999) showed that the good properties of characteristic sets of prime ideals generalize to regular chains. Kalkbrener gave also an algorithm for computing an unmixed-dimensional decomposition $V_1 \cup \cdots \cup V_m$ of an algebraic variety V, where each V_i is given by a regular chain, representing the generic points of the irreducible components of V_i. However, Kalkbrener's decompositions may contain redundant components and do not represent the points of V that are not generic.

The redundancy problem is studied by Lazard (1991) who strengthened the notion of a regular chain. Without giving a correctness proof, he also proposed a new method to obtain Wu-like decompositions where all the points, generic or not, are represented. Following the work of Seidenberg (1956) another approach was given by Wang (1993). By associating a set of inequations to every characteristic set, Wang's method avoids redundant components but this technique makes the output sometimes hard to handle. By considering special triangular systems, called simple systems, this last point has been improved by Wang (1998). Simple systems are closely related to Lazard regular chains and Dellière (1999) showed that simple systems could represent the triangular systems involved in Dynamic Evaluation (Gómez Díaz, 1994).

Another approach for computing equidimensional decompositions of algebraic varieties is that of *geometric resolution* initiated by Giusti, Heintz, Pardo and collaborators (Giusti et al., 1995; Pardo, 1995). In (Lecerf, 2003), follow-

ing (Lecerf, 2000), Lecerf extends these techniques. His algorithm proceeds also by solving incrementally subsystems of the input equations, and uses Newton's iterator, notably in its version of (Lecerf, 2002), to work only with zero- and one-dimensional varieties. This algorithm is probabilistic, as it requires generic enough coordinates.

The four methods of Wu, Lazard, Wang and Kalkbrener are compared from an experimental point of view by Aubry and Moreno Maza (1999). It appears that the methods based on regular chains perform better than the others, and, among them, Kalkbrener's method have better timings. However, Kalkbrener's decompositions are not suitable for algebraic varieties of positive dimension where all the points, generic or not, are needed. It appears also that the construction of Lazard regular chains may be expensive in positive dimension. Finally, it appears that the methods based on regular chains generate many redundant intermediate components. Therefore the challenge of providing an efficient algorithm for controlling the intermediate computations and for computing triangular decompositions in the sense of Lazard remained.

The algorithm presented in this paper decomposes the variety V by means of regular chains. All the zeros of V, generic or not, are represented. However, our algorithm can be restricted to represent generic zeros only. Moreover, it can produce Lazard's regular chains, if required.

A preliminary implementation (which does not contain most of the improvements given in this paper) is distributed since the release 2.2 of AXIOM and a technical report (Moreno Maza, 1999) describes the algorithm. Experiments are reported by Aubry and Moreno Maza (1999). They show that this implementation is comparable in performances with that of Kalkbrener, although our method provides a complete decomposition of V. Moreover we obtain much better results than with our implementation of Lazard's method.

An optimized implementation was realized in ALDOR. Foursov and Moreno Maza (2002) used it for classifying coupled integrable equations. and Kogan and Moreno Maza (2002) used it for studying equivalence classes of ternary cubics under general linear changes of variables. The release 10 of MAPLE contains also an implementation of our algorithm. It is reported in Lemaire et al. (2005) and is used by Boulier et al. (2004) in nonlinear control theory.

The algorithms and complexity estimates of (Schost, 2003) and (Dahan and Schost, 2004) together with the notion of equiprojectable decomposition (Dahan et al., 2004) have initiated the study of modular algorithms for computing triangular decompositions. Combined with the algorithms presented in this paper, this has led to the first of such an algorithm (Dahan et al., 2005).

Our decomposition strategy is based on a procedure called decompose that computes the intersection of a hypersurface with a quasi-component (the set

of the generic points associated with a regular chain). More precisely, this operation computes the quasi-components with maximal dimension contained in this intersection, and the necessary information in order to obtain those of lower dimension when they are needed.

This lazy evaluation feature is exploited by a top-level procedure which manages the calls to the procedure decompose such that the output quasi-components are produced in order of decreasing dimension and such that the redundant quasi-components are removed as soon as possible, by means of an inclusion test. The procedure decompose itself relies on a procedure for polynomial gcd computations modulo regular chains that lazily produces a *main result* and a way to terminate some *degenerated computations* when they are needed. This strategy leads to the following difficulty. Since we proceed in order of decreasing dimension, we cannot use a top-down elimination process (eliminating the greatest variables first) as in Wang's algorithm or a bottom-up construction as in Kalkbrener's algorithm. In fact, we need to perform projections into subspaces and then reconstructions in the whole space where the algebraic variety V is to be decomposed. We solve this problem with Theorem 23 which provides a lifting property for polynomial computations modulo regular chains.

The paper is organized as follows.

Section 1. For the reader's convenience we review the concept of a regular chain.

Section 2. We give some fundamental properties of regular chains. They may be deduced from most text books in commutative algebra. For the reader's convenience we give direct proofs, because we have not found any reference where they are explicit.

Section 3. We introduce the notions needed for the description of our decomposition process in order of decreasing dimension. Our lifting property concludes this section.

Section 4. We assume that we are given an operation for computing polynomial gcds modulo regular chains and an operation for testing the regularity of a polynomial w.r.t. a regular chain. Then, we state and prove our algorithm decompose for computing the quasi-components of maximal dimension in the intersection of a hypersurface and a quasi-component.

Section 5. We show that the assumptions of the previous sections hold and we prove that the whole decomposition process terminates and is correct. Our arguments rely only on the properties of regular chains given in (Aubry et al., 1999) and the basic properties of prime ideals (Samuel and Zariski, 1967).

Section 6. We give several improvements of the algorithms of the previous sections. We show how to obtain Kalkbrener's decomposition by means of our method. Lazard regular chains have two additional properties w.r.t. general regular chains: we also show how to obtain them with our method. Then

we discuss the problem of deciding whether a quasi-component is contained in another. This question is crucial for removing the redundant intermediate quasi-components during our decomposition process. Finally, we explain how our decomposition process must be conducted in order to remove the redundant intermediate quasi-components at an early stage.

1 Notations and Basic Definitions

We review in this section the notions related to triangular sets and regular chains. See (Aubry, 1999; Aubry et al., 1999; Moreno Maza, 1997; Boulier et al., 2001) for more details. We shall use some notions from commutative algebra (such as the dimension of an ideal) and refer for instance to (Samuel and Zariski, 1967) for this subject.

Let $\mathbf{k}$ be a field and let $x_1 < x_2 < \cdots < x_n$ be n ordered variables. For every $i = 1 \cdots n$ we define $\mathbf{P}_i = \mathbf{k}[x_1, \ldots, x_i]$ and we put $\mathbf{P}_0 = \mathbf{k}$. Let $q \in \mathbf{P}_n$ be a non-constant polynomial i.e. $q \notin \mathbf{k}$. The leading coefficient and the degree of p regarded as univariate polynomial in the variable x_i will be denoted by $\mathsf{lc}(p, x_i)$ and $\mathsf{deg}(p, x_i)$ respectively. The greatest variable occurring in q is called the *main variable* of q and it is denoted by $\mathsf{mvar}(q)$. The degree, the leading coefficient, the (monic) leading monomial, the leading term, and the reductum of q regarded as a univariate polynomial in $\mathsf{mvar}(q)$ are called the *main degree*, the *initial*, the *rank*, the *head* and the *tail* of q; they are denoted by $\mathsf{mdeg}(q)$, $\mathsf{init}(q)$, $\mathsf{rank}(q)$, $\mathsf{head}(q)$ and $\mathsf{tail}(q)$ respectively. Hence we have $q = \mathsf{head}(q) + \mathsf{tail}(q)$ and $\mathsf{head}(q) = \mathsf{init}(q)\,\mathsf{rank}(q)$. We call the *separant* of q the derivative of q w.r.t. $\mathsf{mvar}(q)$. For $p \in \mathbf{P}_n$ we denote by $\mathsf{prem}(p, q)$ and $\mathsf{pquo}(p, q)$ the pseudo-remainder and the pseudo-quotient of p by q as univariate polynomials in $\mathsf{mvar}(q)$. We say that p is *smaller than* q and we write $p \prec q$ if either $p \in \mathbf{k}$ and $q \notin \mathbf{k}$ or both are non-constant polynomials such that either $\mathsf{mvar}(p) < \mathsf{mvar}(q)$ or $\mathsf{mvar}(p) = \mathsf{mvar}(q)$ and $\mathsf{mdeg}(p) < \mathsf{mdeg}(q)$. We write $p \sim q$ if neither $p \prec q$ nor $q \prec p$ hold.

Let $\mathcal{I}$ be an ideal of $\mathbf{P}_n$. We denote by $\mathbf{Ass}(\mathcal{I})$ the set of the prime ideals associated with $\mathcal{I}$ and by $\sqrt{\mathcal{I}}$ the radical of $\mathcal{I}$. We denote by $\mathcal{I} : p^\infty$ the *saturated ideal* of $\mathcal{I}$ w.r.t. p, that is the set of the polynomials q such that there exists an integer $m \geq 0$ such that $p^m q \in \mathcal{I}$. Let $\mathbf{K}$ be an algebraic closure of $\mathbf{k}$ and let F be a subset of $\mathbf{P}_n$. We denote by $\mathcal{I}(F)$ the ideal generated by F in $\mathbf{P}_n$ and by $\mathbf{V}(F)$ the affine variety associated with F in $\mathbf{K}^n$. If $F = \emptyset$ we define $\mathbf{V}(F) = \mathbf{K}^n$. If F consists of a single element f, then f will also denote the set F when no confusion is possible. For any $W \subseteq \mathbf{K}^n$ we denote by $\overline{W}$ the Zariski closure of W w.r.t. $\mathbf{k}$.

A subset T of $\mathbf{P}_n$ is called a *triangular set* if $T \cap \mathbf{k}$ is empty and if for every

$t \in T$ and every $t' \in T$ such that $t \neq t'$ we have $\mathsf{mvar}(t) \neq \mathsf{mvar}(t')$. A variable x_i is *algebraic* w.r.t. T if there exists $t \in T$ such that $x_i = \mathsf{mvar}(t)$. We denote by $\mathsf{alg}(T)$ the set of the algebraic variables of T. For $x_i \in \mathsf{alg}(T)$ we denote by T_{x_i} the polynomial in T whose main variable is x_i. We denote by $T^-_{x_i}$ and $T^+_{x_i}$ the sets of polynomials $t \in T$ such that $\mathsf{mvar}(t) < x_i$ and $\mathsf{mvar}(t) > x_i$ respectively. We say that T is *purely algebraic* if for every variable v occurring in T there exists $t \in T$ such that $v = \mathsf{mvar}(t)$. The pseudo-remainder of p w.r.t. T is denoted by $\mathsf{prem}(p, T)$.

A point $\zeta \in \mathbf{V}(T)$ is called a *regular zero* of T if for every $t \in T$ the initial of t does not vanish at ζ. The set of the regular zeros of T is denoted by $\mathbf{W}(T)$. We denote by $\mathbf{Z}(F, T)$ the intersection of $\mathbf{W}(T)$ and $\mathbf{V}(F)$.

Let i be in the range $0 \cdots n$. Assume $T \cap \mathbf{P}_i \neq \emptyset$. We denote by $\mathsf{sat}_i(T)$ the ideal $\mathcal{I}(T \cap \mathbf{P}_i) : h_i^\infty$ of $\mathbf{P}_i$ where h_i is the product of the initials of $T \cap \mathbf{P}_i$. If $T \cap \mathbf{P}_i = \emptyset$ we define $\mathsf{sat}_i(T) = \{0\}$. The *saturated ideal* of T is the ideal of $\mathbf{P}_n$ denoted by $\mathbf{Sat}(T)$ and defined by $\mathbf{Sat}(T) = \mathsf{sat}_n(T)$. We denote by $\mathsf{dim}(T)$ the dimension of the ideal $\mathbf{Sat}(T)$. For any triangular set T we have:

$$\overline{\mathbf{W}(T)} = \mathbf{V}(\mathbf{Sat}(T)). \tag{1}$$

Moreover if $\mathbf{W}(T) \neq \emptyset$ then for every $i = 0 \cdots n$ we have:

$$\mathbf{Ass}(\mathbf{Sat}(T) \cap \mathbf{P}_i) \;=\; \{\mathcal{P} \cap \mathbf{P}_i \;\mid\; \mathcal{P} \in \mathbf{Ass}(\mathbf{Sat}(T))\}. \tag{2}$$

We introduce here the *reduced form* of the polynomial p w.r.t. the ideal $\mathcal{I}$ as the polynomial denoted by $\mathsf{red}(p, \mathcal{I})$ and defined as follows. If $p \in \mathcal{I}$ then $\mathsf{red}(p, \mathcal{I}) = 0$, otherwise if $p \in \mathbf{k}$ then $\mathsf{red}(p, \mathcal{I}) = p$, otherwise if $\mathsf{head}(p) \in \mathcal{I}$ then $\mathsf{red}(p, \mathcal{I}) = \mathsf{red}(\mathsf{tail}(p), \mathcal{I})$, otherwise $\mathsf{red}(p, \mathcal{I}) = \mathsf{red}(\mathsf{init}(p), \mathcal{I})\ \mathsf{rank}(p) + \mathsf{red}(\mathsf{tail}(p), \mathcal{I})$. It is important to observe that p and $\mathsf{red}(p, \mathcal{I})$ are equal modulo $\mathcal{I}$. We shall use $\mathsf{red}(p, T)$ as a short hand for $\mathsf{red}(p, \mathbf{Sat}(T))$.

Let us denote by $\mathsf{rank}(T)$ the set of the $\mathsf{rank}(t)$ for $t \in T$. Let S be a second triangular set. We say that T has *smaller rank* than S and we write $T \prec S$ if there exists $v \in \mathsf{alg}(T)$ such that $\mathsf{rank}(\ T^-_v) = \mathsf{rank}(\ S^-_v)$ and either $v \notin \mathsf{alg}(S)$ or $v \in \mathsf{alg}(S)$ and $T_v \prec S_v$. We say that T and S have the *same rank* and we write $T \sim S$ if $\mathsf{rank}(T) = \mathsf{rank}(S)$. Recall that any sequence of triangular sets which is strictly decreasing w.r.t. $\prec$ is finite.

The polynomial $p \in \mathbf{P}_n$ is *regular* w.r.t. (or modulo) the ideal $\mathcal{I}$ if for every $\mathcal{P} \in \mathbf{Ass}(\mathcal{I})$ we have $p \notin \mathcal{P}$. The polynomial $p \in \mathbf{P}_n$ is *regular* w.r.t. the triangular set T if it is regular w.r.t. $\mathbf{Sat}(T)$. The triangular set $T \subseteq \mathbf{P}_n$ is a *regular chain* if for any $x_i \in \mathsf{alg}(T)$ with $i \geq 2$ the initial of T_{x_i} is regular w.r.t. $\mathsf{sat}_{i-1}(T)$. It is shown in (Aubry, 1999) that T is a regular chain iff for every $i = 0 \cdots n$ we have $\mathsf{sat}_{i-1}(T) = \mathbf{Sat}(T) \cap \mathbf{P}_i$. Moreover in (Moreno Maza, 1997) we show that T is a regular chain iff for every $p \in \mathbf{P}_n$ we have $p \in$

$\mathbf{Sat}(T) \iff \mathsf{prem}(p, T) = 0$. A subset W of $\mathbf{K}^n$ is a *quasi-component* if there exists a regular chain T such that $\mathbf{W}(T) = W$. It is shown in (Kalkbrener, 1991) that if T is a regular chain with r elements, then $\mathbf{W}(T) \neq \emptyset$ and $\mathbf{Sat}(T)$ is unmixed-dimensional with dimension $n-r$. Let k be an integer in the range $0 \cdots n-1$. Let $p \in \mathbf{P}_n$ with $p \notin \mathbf{k}$ and $\mathsf{mvar}(p) = x_{k+1}$. Assume that T is a regular chain contained in $\mathbf{P}_k$ and that $\mathsf{init}(p)$ is regular w.r.t. T. Then we have:

$$\mathbf{Sat}(T \cup p) \;=\; (\mathcal{I}(p) + \mathbf{Sat}(T)) : \mathsf{init}(p)^{\infty}. \tag{3}$$

See Proposition 4.3.2. of (Aubry, 1999). This property has an immediate and very useful consequence. Let T' be a second regular chain of $\mathbf{P}_n$ contained in $\mathbf{P}_k$ such that $T' \cup p$ is a regular chain and $\mathbf{Sat}(T) \subseteq \mathbf{Sat}(T')$. Relation (3) shows that we have $\mathbf{Sat}(T \cup p) \subseteq \mathbf{Sat}(T' \cup p)$.

The polynomial p is *normalized* w.r.t. T if either $p \in \mathbf{k}$ or $\mathsf{mvar}(p) \notin \mathsf{alg}(T)$ and $\mathsf{init}(p)$ is normalized w.r.t. T. The triangular set T is *normalized* if for every $x_i \in \mathsf{alg}(T)$ the polynomial T_{x_i} is normalized w.r.t $T^{-}_{x_i}$. Any normalized triangular set is a regular chain. For the concepts of *square-free* regular chain and *Lazard* triangular set, please see (Aubry and Moreno Maza, 1999).

To state our algorithms, we use the syntax of the programming language of the Computer Algebra System AXIOM (Jenks and Sutor, 1992). In particular we use indentation for denoting blocks of instructions. However we use # instead of $--$ to start a line of comments. Let $\mathtt{B} = \{I_1, \ldots, I_\ell\}$ be a block of instructions. Assume that some instruction I_j has the form $\mathtt{Bool} \Longrightarrow \mathtt{C}$ where `Bool` evaluates to a boolean value and `C` is a block of instructions. Then `B` is equivalent to: $\{I_1, \ldots, I_{j-1}$, **if** `Bool` **then** `C` **else** $\{I_{j+1}, \ldots, I_\ell\}\}$. Assume now that `B` is the block of instructions of a loop. If `C` consists of the single instruction **iterate**, then `B` is equivalent to: $\{I_1, \ldots, I_{j-1}$, **if not** `Bool` **then** $\{I_{j+1}, \ldots, I_\ell\}\}$.

An instruction **return**(x) inside a body of a function means: return x and leave the body. Most of our algorithms return a sequence of values. One way to implement this is to manage a list S of the output items and to use **return**(S) when S is complete. Another approach is to use the concept of a *generator* which is supported by ALDOR, the AXIOM compiler (Broadbery et al., 1994). Instead of inserting each output item s in a list we throw s directly in the flow of output with the statement **output**(s) and we use the key-word **exit** to leave the body.

2 Properties of Quasi-Components under Factorization

Most of the algorithms described in this paper take a regular chain T as part of their input and return a sequence $T_1, \ldots, T_\ell$ of regular chains as part of their output. We give in this section various relations on regular chains and

quasi-components. More precisely, we study in Lemma 4 and Theorem 7 the possible *splits* of the quasi-component $\mathbf{W}(T)$ when a polynomial of T has a non-trivial factorization. We consider two kinds of relations. The following definition clarifies this latter point.

Definition 1 *Let $W, W_1, \ldots, W_\ell$ be subsets of the affine space $\mathbf{K}^n$. We say that the sequence $(W_1, \ldots, W_\ell)$ is a* Kalkbrener split *of W and we write*

$$W \longrightarrow_K (W_1, \ldots, W_\ell)$$

if W and the union $W_1 \cup \cdots \cup W_\ell$ have the same closure w.r.t. Zariski topology. We say that $(W_1, \ldots, W_\ell)$ is a Lazard split *of W and we write*

$$W \longrightarrow_L (W_1, \ldots, W_\ell)$$

if this is a Kalkbrener split of W and if the union $W_1 \cup \cdots \cup W_\ell$ contains W. By convention, if W is empty, the empty sequence is a Lazard split of W.

The terminology follows from the specification of the algorithms of Kalkbrener Kalkbrener (1993) and Lazard Lazard (1991) as they are presented in Aubry and Moreno Maza (1999). Let us point out two useful relations that follow immediately from Definition 1.

Proposition 2 *Let $W, W_1, \ldots, W_\ell, W_{1,1}, \ldots, W_{1,n_1}, \ldots, W_{\ell,1}, \ldots, W_{\ell,n_\ell}$ be subsets of $\mathbf{K}^n$ such that $W \longrightarrow_L (W_1, \ldots, W_\ell)$. The following properties hold.*

(*a*) *If for every $i = 1 \cdots \ell$ the relation $W_i \longrightarrow_L (W_{i,1}, \ldots, W_{i,n_i})$ is satisfied, then we have:*

$$W \longrightarrow_L (W_{1,1}, \ldots, W_{1,n_1}, \ldots, W_{\ell,1}, \ldots, W_{\ell,n_\ell}).$$

(*b*) *Assume that W is an algebraic variety. Let V be another algebraic variety such that the relation $V \cap W_i \longrightarrow_L (W_{i,1}, \ldots, W_{i,n_i})$ holds for every $i = 1 \cdots \ell$. Then we have:*

$$V \cap W \longrightarrow_L (W_{1,1}, \ldots, W_{1,n_1}, \ldots, W_{\ell,1}, \ldots, W_{\ell,n_\ell}).$$

Observe that relation (*a*) holds also for Kalkbrener splits. In Section 3 we will introduce another kind of splits that satisfies a similar property (Proposition 13) and we will often refer to it as the *Composition Property*. Of course, without this nice behavior, a notion of *split* would have a limited interest.

Property (*b*) is crucial for decomposing an algebraic variety into quasi-components in an incremental manner. We shall explain this in Section 3. Unfortunately property (*b*) does not hold for Kalkbrener splits. Indeed, consider a variety $W \subseteq \mathbf{K}^n$ and a regular chain $T \subseteq \mathbf{P}_n$ such that $\overline{\mathbf{W}(T)} = W$ and $\mathbf{W}(T) \neq W$ hold. Assume that for some $t \in T$ the initial h of t satisfies

$(W \setminus \mathbf{W}(T)) \cap \mathbf{V}(h) \neq \emptyset$. We have $W \longrightarrow_K (\mathbf{W}(T))$ and $W \cap \mathbf{V}(h) \neq \emptyset$. However we have $\mathbf{W}(T) \cap \mathbf{V}(h) = \emptyset$. Hence $(\mathbf{W}(T) \cap \mathbf{V}(h))$ is not a Kalkbrener split of $W \cap \mathbf{V}(h)$. Note that $(\mathbf{W}(T))$ is not a Lazard split of W. In Lemma 4 and in Theorem 7 we shall give fundamental examples of Kalkbrener and Lazard splits by means of Definition 3. The proof of Lemma 4 follows easily from the relations (1), (2) and (3) of Section 1.

Definition 3 *Let $T, T_1, \ldots, T_\ell$ be regular chains in $\mathbf{P}_n$. We say that $(T_1, \ldots, T_\ell)$ is a* Kalkbrener split *(resp.* Lazard split*) of T if $(\mathbf{W}(T_1), \ldots, \mathbf{W}(T_\ell))$ is a Kalkbrener split (resp. Lazard split) of $\mathbf{W}(T)$. We will use the same notations to denote splits of regular chains as those for splits of subsets of $\mathbf{K}^n$.*

Lemma 4 *Let $T \subseteq \mathbf{P}_n$ be a regular chain. Let $t, c \in \mathbf{P}_n$ be two non-constant polynomials such that c is regular w.r.t. T and $\mathsf{mvar}(c) < \mathsf{mvar}(t)$ holds. Then $T \cup t$ is a regular chain if and only if $T \cup c\,t$ is a regular chain. Moreover, if $T \cup t$ is a regular chain, then the following relations hold:*

(a) $T \cup t \longrightarrow_K (T \cup c\,t)$,
(b) $T \cup c\,t \longrightarrow_L (T \cup t)$,
(c) $\mathbf{W}(T \cup t) \longrightarrow_L (\mathbf{W}(T \cup c\,t), \mathbf{Z}(c, T \cup t))$.

Roughly speaking, Lemma 4 shows how to split the quasi-component $\mathbf{W}(T)$ under the primitive factorization of t regarded as univariate w.r.t. $\mathsf{mvar}(t)$. We are interested now in the case where t factors into several polynomials with $\mathsf{mvar}(t)$ as main variable. This problem is treated in Theorem 7 which requires the following preliminary results.

Proposition 5 *Let T and U be two regular chains of $\mathbf{P}_n$. Assume that the radical of $\mathbf{Sat}(T)$ is contained in the radical of $\mathbf{Sat}(U)$. Then we have:*

(a) $\mathsf{dim}(U) \leq \mathsf{dim}(T)$,
(b) *If $\mathsf{dim}(U) = \mathsf{dim}(T)$ then every polynomial h regular w.r.t. T is also regular w.r.t. U.*

PROOF. Our assumption means that the intersection of the prime ideals associated with $\mathbf{Sat}(T)$ is contained in the intersection of the prime ideals associated with $\mathbf{Sat}(U)$. Now recall that if a prime ideal contains an intersection of ideals then it contains one of them. Therefore every prime ideal associated with $\mathbf{Sat}(U)$ contains a prime ideal associated with $\mathbf{Sat}(T)$. Since the saturated ideal of a regular chain is unmixed, property (a) is proved. Now assume that $\mathsf{dim}(U) = \mathsf{dim}(T)$ holds. This implies that every prime ideal associated with $\mathbf{Sat}(U)$ is a prime ideal associated with $\mathbf{Sat}(T)$. So if a polynomial h is regular w.r.t. T, then it is regular w.r.t. U. $\square$

Lemma 6 *Let $T \subseteq \mathbf{P}_n$ be a triangular set. Let a, b, t be non-constant polynomials with the same main variable x_{k+1} (with k in the range $0 \cdots n-1$). We denote by h_a the initial of a. We assume that the following properties hold:*

(i) $T \cup t$ *is a regular chain,*
(ii) $\mathsf{prem}(t, a) \in \mathbf{Sat}(T) \cap \mathbf{P}_k$ *and* $b = \mathsf{pquo}(t, a)$,
(iii) h_a *is regular w.r.t.* T.

Then $T \cup a$ and $T \cup b$ are regular chains. Moreover we have:

(a) $\mathbf{Sat}(T \cup t) \subseteq \mathbf{Sat}(T \cup a)$,
(b) $\mathbf{Sat}(T \cup t) \subseteq \mathbf{Sat}(T \cup b)$,
(c) $\sqrt{\mathbf{Sat}(T \cup t)} = \sqrt{\mathbf{Sat}(T \cup a)} \cap \sqrt{\mathbf{Sat}(T \cup b)}$.

PROOF. Let us fix some notations. We define $h = \mathsf{init}(t)$ and $h_b = \mathsf{init}(b)$. The fact that the polynomials a, b, t have main variable x_{k+1} together with assumption (ii) implies that there exist an integer $e \geq 1$ and a polynomial $r_t \in \mathbf{Sat}(T) \cap \mathbf{P}_k$ such that: ${h_a}^e\, t = a\, b + r_t$.

First we assume that $T \subseteq \mathbf{P}_k$ holds. Assumption (iii) implies that $T \cup a$ is a regular chain. We prove that $T \cup b$ is a regular chain too. Observe that the product ${h_a}^e\, h$ equals the product $h_a\, h_b$. Since h_a and h are regular w.r.t. T, then so is h_b. Hence $T \cup b$ is a regular chain. We prove that $\mathbf{Sat}(T \cup t) \subseteq \mathbf{Sat}(T \cup a)$. Let f be in $\mathbf{Sat}(T \cup t)$. Thus $\mathsf{prem}(f, t)$ lies in $\mathbf{Sat}(T)$. Then there exits a non-negative integer e', a polynomial $q_f \in \mathbf{P}_{k+1}$ and a polynomial $r_f \in \mathbf{Sat}(T) \cap \mathbf{P}_k$ such that ${h^{e'}}\, f = q_f\, t + r_f$. Therefore we have:

$$ {h_a}^e\, {h^{e'}}\, f = q_f\, a\, b + q_f\, r_t + {h_a}^e\, r_f. \tag{4} $$

This shows that ${h_a}^e\, {h^{e'}}\, f \in \mathcal{I}(a) + \mathbf{Sat}(T)$ and thus that ${h^{e'}}\, f \in (\mathcal{I}(a) + \mathbf{Sat}(T)) : h_a^\infty$. With relation (3) page 7, we obtain ${h^{e'}}\, f \in \mathbf{Sat}(T \cup a)$. Thus f lies in $\mathbf{Sat}(T \cup a) : h^\infty$. Since $h \in \mathbf{P}_k$ and since h is regular w.r.t. $\mathbf{Sat}(T)$, it cannot belong to a prime ideal associated with $\mathbf{Sat}(T \cup a)$ (we are using here relation (2) page 6). Therefore we have $\mathbf{Sat}(T \cup a) : h^\infty = \mathbf{Sat}(T \cup a)$ and we obtain $f \in \mathbf{Sat}(T \cup a)$. Now we prove that $f \in \mathbf{Sat}(T \cup b)$. Recall that we have ${h_a}^e\, h = h_a\, h_b$. Thus equation 4, after multiplication by h, shows that $h_a\, h_b\, {h^{e'}}\, f$ lies in $\mathcal{I}(b) + \mathbf{Sat}(T)$. As above this leads to $h_a\, {h^{e'}}\, f \in \mathbf{Sat}(T \cup b)$. Since $h_a\, {h^{e'}}$ is regular w.r.t. T, we conclude that $f \in \mathbf{Sat}(T \cup b)$.

We come back to the general case and assume that T contains at least one polynomial with main variable greater than x_{k+1}. We define $v = x_{k+1}$. Let p be the polynomial of T_v^+ with smallest main variable. We know that $T_v^- \cup t$, $T_v^- \cup a$, and $T_v^- \cup b$ are regular chains such that the saturated ideal of the first one is contained in the saturated ideals of the other ones. Since they all have the same dimension, property (b) of Proposition 5 shows that $\mathsf{init}(p)$ is regular

w.r.t. $T_v^- \cup a$ and $T_v^- \cup b$. Hence $T_v^- \cup \{a, p\}$ and $T_v^- \cup \{b, p\}$ are regular chains. Moreover relation (3) page 7 shows that their saturated ideals contain that of $T_v^- \cup \{t, p\}$. Continuing in this manner with the other polynomials of T_v^+, it follows that $T \cup a$ and $T \cup b$ are regular chains and that both the relations (a) and (b) hold. Finally, proving relation (c) is easy. $\square$

Theorem 7 *With the same notations and assumptions as in Lemma 6, the following properties hold:*

(a) $T \cup t \longrightarrow_K (T \cup a, T \cup b)$,
(b) $\mathbf{W}(T \cup t) \longrightarrow_L (\mathbf{W}(T \cup a), \mathbf{W}(T \cup b), \mathbf{Z}(h_a, T \cup t))$.

PROOF. Property (a) follows from relation (1) page 6 and property (c) of Lemma 6. We prove property (b). Using the notations of the proof of Lemma 6, observe first that the polynomial ${h_a}^e\, t - a\, b$ vanishes at any point ζ of the algebraic variety associated with $\mathbf{Sat}(T)$. Since $\mathbf{V}(\mathbf{Sat}(T))$ contains $\mathbf{W}(T)$ we get $\mathbf{Z}(h_a\, t, T) = \mathbf{Z}(a\, b, T)$. Since the initials of ${h_a}^e\, t$ and ab are equal, we deduce: $\mathbf{W}(T \cup h_a\, t) = \mathbf{W}(T \cup a\, b)$. This can be easily written as:

$$\mathbf{W}(T \cup h_a\, t) \;=\; \mathbf{W}(T \cup h_b\, a) \cup \mathbf{W}(T \cup h_a\, b) \tag{5}$$

where each triangular set is a regular chain by Lemmas 4 and 6. Now observe that we have:

$$\mathbf{W}(T \cup t) \;=\; \mathbf{W}(T \cup h_a\, t) \;\cup\; \mathbf{Z}(h_a, T \cup t).$$

Thus with relation (5) and property (a) we obtain:

$$\begin{aligned}
\mathbf{W}(T \cup t) &= \mathbf{W}(T \cup h_b\, a) \;\cup\; \mathbf{W}(T \cup h_a\, b) \;\cup\; \mathbf{Z}(h_a, T \cup t) \\
&\subseteq \mathbf{W}(T \cup a) \;\cup\; \mathbf{W}(T \cup b) \;\cup\; \mathbf{Z}(h_a, T \cup t) \\
&\subseteq \overline{\mathbf{W}(T \cup a)} \;\cup\; \overline{\mathbf{W}(T \cup b)} \;\cup\; \overline{\mathbf{Z}(h_a, T \cup t)} \\
&\subseteq \overline{\mathbf{W}(T \cup t)}.
\end{aligned}$$

The theorem is proved. $\square$

Corollary 8 *With the above hypothesis, let* $(\mathbf{W}(T_1), \ldots, \mathbf{W}(T_\ell))$ *be a Lazard split of* $\mathbf{Z}(h_a, T \cup t)$ *then* $(T \cup a, T \cup b, T_1, \ldots, T_\ell)$ *is a Lazard split of* $T \cup t$.

3 Incremental Triangular Decompositions

Let F denote a subset of $\mathbf{P}_n$ and let V be its associated algebraic variety in $\mathbf{K}^n$. Our aim is to decompose V into a finite union of regular chains by solving each of the following problems:

(K) Compute regular chains $T_1, \ldots, T_\ell$ such that $V \longrightarrow_K (\mathbf{W}(T_1), \ldots, \mathbf{W}(T_\ell))$,
(L) Compute regular chains $T_1, \ldots, T_\ell$ such that $V \longrightarrow_L (\mathbf{W}(T_1), \ldots, \mathbf{W}(T_\ell))$.

Since $V = \overline{V}$, solving each problem leads to a decomposition of V. The former brings a decomposition into varieties and the latter brings a decomposition into quasi-components. Clearly, a solution of problem (L) is a solution of problem (K) but the converse is false. Roughly speaking, a solution of problem (K) gives only the generic zeros (in the sense of van der Waerden (1991)) of the irreducible components of V. See for instance (Aubry and Moreno Maza, 1999) for several examples. Another interesting feature of problem (L) is that it can be solved in an incremental manner; this follows from Proposition 2. So we will concentrate now on problem (L). In Section 6 we shall show how to modify our algorithm for solving problem (L) in order to get an efficient method for solving problem (K).

The goal of the following three sections will be to describe an operation denoted by solve such that solve(F) computes a Lazard split of V. Proceeding in an incremental manner, we shall reduce the computation of solve(F) into more elementary computations as follows. Assume that we are given an operation intersect with the following specification as in (Lazard, 1991). For every polynomial $p \in \mathbf{P}_n$ and every regular chain $T \subseteq \mathbf{P}_n$, the output of intersect(p, T) is a sequence $(T_1, \ldots, T_d)$ of regular chains of $\mathbf{P}_n$ such that $\mathbf{Z}(p, T) \longrightarrow_L (\mathbf{W}(T_1), \ldots, \mathbf{W}(T_d))$. By using Proposition 2, this operation leads to the straightforward algorithm.

Algorithm 1

- **Input:** F a finite set of polynomials of $\mathbf{P}_n$.
- **Output:** Regular chains $T_1, \ldots, T_d$ such that $(\mathbf{W}(T_1), \ldots, \mathbf{W}(T_d))$ is a Lazard split of $\mathbf{V}(F)$.
- solve(F) ==
  $C := [\emptyset]$
  # C is a list of regular chains
  while $F \neq \emptyset$ **repeat**
    choose and remove a polynomial p from F
    $C' := [\,]$
    for $T \in C$ **repeat**
      $C' :=$ concat(intersect$(p, T), C')$
    $C := C'$
  return C

Assume that $F = \{p_1, \ldots, p_m\}$ where p_i is the i-th polynomial chosen by Algorithm 1. Then observe that the i-th iteration of the **while** loop computes a split of Lazard of $\mathbf{V}(p_1, \ldots, p_i)$. As pointed out by Lazard (1991) the order in which the regular chains are output should be in decreasing dimension.

Indeed, if the quasi-component $\mathbf{W}(T)$ is contained in the quasi-component $\mathbf{W}(T')$ then we have $\mathsf{dim}(T) \leq \mathsf{dim}(T')$. Hence, assuming that we have an inclusion test for quasi-components, producing regular chains by decreasing dimension allows us to remove redundant quasi-components as soon as possible and to control the number of unnecessary computations. Therefore one would like to compute first the quasi-components with maximal dimension and to delay the computation of the others. This leads to the new notion of splits of Definition 11 together with the following concept.

Definition 9 *A* process *of* $\mathbf{P}_n$ *is a pair* $[F, T]$ *where* F *is a subset of* $\mathbf{P}_n$ *and* T *is a triangular set of* $\mathbf{P}_n$*. We say that the process* $[F_1, T_1]$ *of* $\mathbf{P}_n$ *has a lower rank than the process* $[F_2, T_2]$ *of* $\mathbf{P}_n$ *and we write* $[F_1, T_1] \prec [F_2, T_2]$ *if either* $T_1 \prec T_2$ *holds or* $T_1 \sim T_2$ *holds and there exists a polynomial* f *in* F_1 *with smaller rank w.r.t.* $\prec$ *than any polynomial in* F_2*.*

If F is empty or if T is not a regular chain, then the process $[F, T]$ represents a computation that needs to be done in order to describe $\mathbf{Z}(F, T)$. We will use processes to represent these computations that we want to delay. To establish the termination of our algorithm we need the following Proposition 10, whose proof is straightforward.

Proposition 10 *Let* S *be a sequence of processes in* $\mathbf{P}_n$*. If* S *is strictly decreasing w.r.t. the ordering* $\prec$*, then* S *is finite.*

Definition 11 *Let* $[F_1, T_1], \ldots, [F_d, T_d]$ *and* $[F, T]$ *be processes of* $\mathbf{P}_n$*. We say that the sequence* $(\mathbf{Z}(F_1, T_1), \ldots, \mathbf{Z}(F_d, T_d))$ *is a* delayed split *of* $\mathbf{Z}(F, T)$ *and we write:*

$$\mathbf{Z}(F, T) \longrightarrow_D (\mathbf{Z}(F_1, T_1), \ldots, \mathbf{Z}(F_d, T_d))$$

if for every $i = 1 \cdots d$ *the following conditions hold:*

(D_1) $[F_i, T_i] \prec [F, T]$,
(D_2) $\mathbf{Z}(F, T) \subseteq \mathbf{Z}(F_1, T_1) \cup \cdots \cup \mathbf{Z}(F_d, T_d)$,
(D_3) $\mathbf{Sat}(T) \subseteq \mathbf{Sat}(T_i)$,
(D_4) $F_i \neq \emptyset \implies F \subseteq F_i$,
(D_5) $F_i = \emptyset \implies \mathbf{W}(T_i) \subseteq \mathbf{V}(F)$.

If this holds and if for every $i = 1 \cdots d$ *the triangular set* T_i *is a regular chain, we say that* $(\mathbf{Z}(F_1, T_1), \ldots, \mathbf{Z}(F_d, T_d))$ *is a* regular delayed split *of* $\mathbf{Z}(F, T)$*. By convention, if* $\mathbf{Z}(F, T)$ *is empty, we say that the empty sequence is a regular delayed split of* $\mathbf{Z}(F, T)$*.*

Let $[F, T], [F_1, T_1], \ldots, [F_d, T_d]$ be processes of $\mathbf{P}_n$ such that $(\mathbf{Z}(F_1, T_1), \ldots, \mathbf{Z}(F_d, T_d))$ is a delayed split of $\mathbf{Z}(F, T)$. We give first an informal explanation of Definition 11. Condition (D_1) means that each $\mathbf{Z}(F_i, T_i)$ is better described than $\mathbf{Z}(F, T)$. Condition (D_2) means that no information is lost. Conditions (D_3) to (D_5) mean essentially that the possible extraneous information is

kept under control, as we shall see now. So let us have a closer look at these three conditions. Condition (D_3) implies that each $\mathbf{V}(\mathbf{Sat}(T_i))$ is contained in $\overline{\mathbf{W}(T)}$. Condition (D_4) implies that if F_i is not empty then $\mathbf{V}(F_i) \subseteq \mathbf{V}(F)$ and $\mathbf{Z}(F_i, T_i) \subseteq \mathbf{W}(T_i) \cap \mathbf{V}(F)$ hold. Condition (D_5) implies that if F_i is empty then $\mathbf{W}(T_i) \subseteq \mathbf{V}(F)$ and again $\mathbf{Z}(F_i, T_i) \subseteq \mathbf{W}(T_i) \cap \mathbf{V}(F)$ hold. Therefore with condition (D_2) we obtain:

$$\mathbf{V}(F) \cap \mathbf{W}(T) \subseteq \mathbf{Z}(F_1, T_1) \cup \cdots \cup \mathbf{Z}(F_d, T_d) \subseteq \mathbf{V}(F) \cap \overline{\mathbf{W}(T)}.$$

Observe that $\mathbf{Z}(F_1, T_1) \cup \cdots \cup \mathbf{Z}(F_d, T_d)$ may not be contained in the closure of $\mathbf{Z}(F, T)$ and the sequence $(\mathbf{Z}(F_1, T_1), \ldots, \mathbf{Z}(F_d, T_d))$ may be neither a Kalkbrener split nor a Lazard split of $\mathbf{Z}(F, T)$. It is important to remark that a Kalkbrener split or a Lazard split may not be a delayed split, obviously because of condition (D_1) but also because of condition (D_3). Indeed these former notions of splits impose only conditions on radicals whereas (D_3) is a condition on ideals. Proposition 12 gives an important example of (regular) delayed splits. Its proof follows immediately from Lemma 6 and Theorem 7. Recall that $\mathbf{Z}(\emptyset, T)$ and $\mathbf{W}(T)$ denote the same object.

Proposition 12 *With the same notations and assumptions as in Lemma 6, we have:*

$$\mathbf{W}(T \cup t) \longrightarrow_D (\mathbf{W}(T \cup a),\ \mathbf{W}(T \cup b),\ \mathbf{Z}(h_a, T \cup t)).$$

Following the previous discussion, we need to explain why we can use these splits even though they may lead to extraneous solutions. In fact we shall see that if the starting point of a cascade of delayed splits is a variety, then this bad situation never happens. The first step is to show that we have a *composition property* similar to property (a) of Proposition 2. This is the goal of Proposition 13 whose proof is straightforward. The second step is to show that delayed splits have a *nice behavior* with algebraic varieties, a property similar to point (b) of Proposition 2. This will be done in Proposition 14.

Proposition 13 *Let $[F, T], [F_1, T_1], \ldots, [F_d, T_d], [F_{1,1}, T_{1,1}], \ldots, [F_{1,e}, T_{1,e}]$ be processes of $\mathbf{P}_n$ such that $F_1 \neq \emptyset$. Assume that the following relations hold:*

(i) $\mathbf{Z}(F, T) \longrightarrow_D (\mathbf{Z}(F_1, T_1), \ldots, \mathbf{Z}(F_d, T_d))$,
(ii) $\mathbf{Z}(F_1, T_1) \longrightarrow_D (\mathbf{Z}(F_{1,1}, T_{1,1}), \ldots, \mathbf{Z}(F_{1,e}, T_{1,e}))$.

Then we have:

$$\mathbf{Z}(F, T) \longrightarrow_D (\mathbf{Z}(F_{1,1}, T_{1,1}), \ldots, \mathbf{Z}(F_{1,e}, T_{1,e}), \mathbf{Z}(F_2, T_2), \ldots, \mathbf{Z}(F_d, T_d)).$$

Proposition 14 *Let $F, F_1, \ldots, F_d, F_{1,1}, \ldots, F_{1,e}$ be subsets of $\mathbf{P}_n$ such that $F_1 \neq \emptyset$. Let $T_1, \ldots, T_d$, $T_{1,1}, \ldots, T_{1,e}$ be triangular sets of $\mathbf{P}_n$. Assume that the following relations hold:*

(i) $\mathbf{V}(F) \subseteq \mathbf{Z}(F_1, T_1) \cup \cdots \cup \mathbf{Z}(F_d, T_d)$,
(ii) *for every* $i = 1 \cdots d$ *we have* $F_i = \emptyset \implies \mathbf{W}(T_i) \subseteq \mathbf{V}(F)$,
(iii) *for every* $i = 1 \cdots d$ *we have* $F_i \neq \emptyset \implies \mathbf{W}(T_i) \subseteq \mathbf{V}(F \setminus F_i)$,
(iv) $\mathbf{Z}(F_1, T_1) \longrightarrow_D (\mathbf{Z}(F_{1,1}, T_{1,1}), \ldots, \mathbf{Z}(F_{1,e}, T_{1,e}))$.

Then the properties below hold:

(a) *for every* $j = 1 \cdots e$ *we have* $F_{1,j} = \emptyset \implies \mathbf{W}(T_{1,j}) \subseteq \mathbf{V}(F)$,
(b) *for every* $j = 1 \cdots e$ *we have* $F_{1,j} \neq \emptyset \implies \mathbf{W}(T_{1,j}) \subseteq \mathbf{V}(F \setminus F_{1,j})$,
(c) *If* $F_2, \ldots, F_d, F_{1,1}, \ldots, F_{1,e}$ *are all empty then we have:*

$$\mathbf{V}(F) = \cup_{i=2}^{d} \mathbf{W}(T_i) \cup \cup_{j=1}^{e} \mathbf{W}(T_{1,j}).$$

PROOF. Let j be in the range $1 \cdots e$. We prove property (a). Assume $F_{1,j} = \emptyset$. By condition (D_5) we have $\mathbf{W}(T_{1,j}) \subseteq \mathbf{V}(F_1)$. By condition (D_3) we have $\mathbf{W}(T_{1,j}) \subseteq \overline{\mathbf{W}(T_1)}$. Since $F_1 \neq \emptyset$ by hypothesis (iii) we have $\mathbf{W}(T_1) \subseteq \mathbf{V}(F \setminus F_1)$. Thus $\mathbf{W}(T_{1,j}) \subseteq \mathbf{V}(F_1) \cap \mathbf{V}(F \setminus F_1)$. Therefore $\mathbf{W}(T_{1,j}) \subseteq \mathbf{V}(F)$ and property (a) is proved. Assume now that $F_{1,j} \neq \emptyset$ holds. Thus $F_1 \subseteq F_{1,j}$ holds. Since $\mathbf{W}(T_1) \subseteq \mathbf{V}(F \setminus F_1)$ and $\mathbf{W}(T_{1,j}) \subseteq \overline{\mathbf{W}(T_1)}$ hold, we easily obtain $\mathbf{W}(T_{1,j}) \subseteq \mathbf{V}(F \setminus F_{1,j})$ and property (b) is proved. We prove property (c). Assume that $F_2, \ldots, F_d, F_{1,1}, \ldots, F_{1,e}$ are empty. By hypothesis (i) and (iv) the union of all $\mathbf{W}(T_{i,j})$ and $\mathbf{W}(T_i)$ contains $\mathbf{V}(F)$. By hypothesis (ii) and property (a) the same union is contained in $\mathbf{V}(F)$. Property (c) is proved. $\square$

Delayed splits involving only processes of the form $[\emptyset, T]$ play a special role in the algorithms of Sections 4 and 5. This leads us to Definition 15, Proposition 16 and its Corollary 17 which enhance the results of Section 2.

Definition 15 *Let* $T, T_1, \ldots, T_d$ *be triangular sets of* $\mathbf{P}_n$. *We say that the sequence* $(T_1, \ldots, T_d)$ *is a* delayed split *of* T *and we write:*

$$T \longrightarrow_D (T_1, \ldots, T_d)$$

if either $d = 1$ *and* $T = T_d$, *or* $\mathbf{W}(T) \longrightarrow_D (\mathbf{W}(T_1), \ldots, \mathbf{W}(T_d))$. *If this holds and if for every* $i = 1 \cdots d$ *the triangular set* T_i *is a regular chain, we say that* $(T_1, \ldots, T_d)$ *is a* regular delayed split *of* T. *By convention, if* $\mathbf{W}(T) = \emptyset$ *holds, we say that the empty sequence is a regular delayed split of* T.

Proposition 16 *Let* $T, T_1, \ldots, T_d$ *be regular chains of* $\mathbf{P}_n$. *Let* p *be a polynomial of* $\mathbf{P}_n$ *regular w.r.t.* T *and such that* $(\mathbf{W}(T_1), \ldots, \mathbf{W}(T_d))$ *is a delayed split of* $\mathbf{Z}(p, T)$. *Then for every* $i = 1 \cdots d$ *we have* $\mathsf{dim}(T_i) < \mathsf{dim}(T)$.

PROOF. Indeed, if we had $\mathsf{dim}(T_i) = \mathsf{dim}(T)$, then condition (D_3) of delayed splits together with Proposition 5 shows that p would be regular w.r.t. T_i.

Therefore p would not belong to any prime ideal associated with $\mathbf{Sat}(T_i)$. However condition (D_5) of delayed splits implies that p must belong to the radical of $\mathbf{Sat}(T_i)$. A contradiction. $\square$

Corollary 17 *With the same assumptions as in Lemma 6, let $T_1, \ldots, T_d$ be regular chains of $\mathbf{P}_n$ such that $\mathbf{Z}(h_a, T \cup t) \longrightarrow_D (\mathbf{W}(T_1), \ldots, \mathbf{W}(T_d))$. Then we have: $T \cup t \longrightarrow_L (T \cup a,\ T \cup b,\ T_1, \ldots, T_d)$. Moreover, for every $i = 1 \cdots d$ we have $\mathsf{dim}(T_i) < \mathsf{dim}(T \cup t)$.*

Proposition 18 states a last technical result before presenting our algorithm for solving problem (L) by means of delayed splits. The proof of this proposition is straightforward. From now on we assume that we are given an operation decompose defined below in Specification 1. This leads to Algorithm 2 which provides an operation solve (as in Algorithm 1) based on delayed splits.

Proposition 18 *Let $[F, T], [F_1, T_1], \ldots, [F_d, T_d]$ be processes of $\mathbf{P}_n$ such that $F \neq \emptyset$. Let p be a polynomial in F minimal w.r.t. $\prec$ Assume that $(\mathbf{Z}(H_1, T_1),$ $\ldots,$ $\mathbf{Z}(H_d, T_d))$ is a delayed split of $\mathbf{Z}(p, T)$. Then $(\mathbf{Z}(F \cup H_1, T_1), \ldots, \mathbf{Z}(F \cup H_d, T_d))$ is a delayed split of $\mathbf{Z}(F, T)$.*

Specification 1 *For any regular chain $T \subseteq \mathbf{P}_n$ and any polynomial $p \in \mathbf{P}_n$ such that $p \notin \mathbf{k}$ and $p \notin \mathbf{Sat}(T)$ the operation $\mathsf{decompose}(p, T)$ returns a regular delayed split of $\mathbf{Z}(p, T)$.*

Algorithm 2

- **Input:** F a finite set of polynomials of $\mathbf{P}_n$.
- **Output:** Regular chains $T_1, \ldots, T_d$ such that $(\mathbf{W}(T_1), \ldots, \mathbf{W}(T_d))$ is a Lazard split of $\mathbf{V}(F)$.
- $\mathsf{solve}(F) ==$

  $R := [[F, \emptyset]]$
  # R is a list of processes
  while $R \neq [\,]$ **repeat**
    choose and remove a process $[F_1, U_1]$ from R
    $F_1 = \emptyset \Longrightarrow$ **output** U_1
    choose a polynomial $p \in F_1$ minimal w.r.t. $\prec$
    $G := F_1 \setminus \{p\}$
    $p := \mathsf{red}(p, U_1)$
    $p = 0 \Longrightarrow R := \mathsf{cons}\ ([G, U_1], R)$
    $p \in \mathbf{k} \Longrightarrow$ **iterate**
    for $[H, T] \in \mathsf{decompose}(p, U_1)$ **repeat**
      $R := \mathsf{cons}\ ([F_1 \cup H, T], R)$
  exit

Proposition 19 *If every call to the operation* decompose *satisfies Specification 1, then Algorithm 2 terminates and satisfies its specification.*

PROOF. We denote by L_i the i-th iteration of the **while** loop. Let $[F_1, U_1]$, $\ldots$, $[F_r, U_r]$ be the processes in R before L_i. Let $T_1, \ldots, T_s$ be the regular chains that have returned via the **output** statement before L_i. Assume that during L_i the process $[F_1, U_1]$ is chosen and removed from R. We first prove that the above algorithm terminates. If F_1 is empty, note that the number of processes has decreased after L_i. If F_1 is not empty and if the polynomial p reduces to a constant w.r.t. U_1, then again the number of processes has decreased. Thus if the **while** loop is infinite, the **for** loop needs to be performed an infinite number of times. Now assume that the **for** loop is performed during L_i. Observe that the new processes have smaller rank than $[F_1, U_1]$ (because of the choice of p and because of condition (D_1) of Definition 11). If the **for** loop was performed an infinite number of times, then the successive values of R could be organized as an ordered tree w.r.t. $\prec$ with at least one infinite chain. But this would contradict Proposition 10. Therefore the algorithm terminates.

We check now that the above algorithm is correct. We assume that before L_i the following relations holds:

(M_i) $\mathbf{V}(F) \subseteq \mathbf{Z}(F_1, U_1) \cup \cdots \cup \mathbf{Z}(F_r, U_r) \cup \mathbf{W}(T_1) \cup \cdots \cup \mathbf{W}(T_s)$,
(m_i) $\mathbf{W}(T_1) \cup \cdots \cup \mathbf{W}(T_s) \subseteq \mathbf{V}(F)$,
(S_i) for every $j = 1 \cdots r$ we have $\mathbf{W}(U_j) \subseteq \mathbf{V}(F \setminus F_j)$.

Note that this assumption is clearly satisfied for every $i = 1$, so we next prove that this still holds for every $i + 1$. Assume that F_1 is empty. Then $\mathbf{Z}(F_1, U_1) = \mathbf{W}(U_1)$. Thus (M_{i+1}) clearly holds. Now U_1 becomes a new output regular chain. From S_i we know that $\mathbf{W}(U_1) \subseteq \mathbf{V}(F)$. Hence relation (m_{i+1}) holds. Trivially (S_{i+1}) also holds. So consider now that a polynomial p is extracted from F_1. Assume that p reduces to zero w.r.t. U_1. This means that $p \in \mathbf{Sat}(U_1)$. Hence $\mathbf{W}(U_1) \subseteq \mathbf{V}(p)$ and $\mathbf{Z}(F_1, U_1) = \mathbf{Z}(F_1 \setminus \{p\}, U_1)$. Thus the relations (M_{i+1}), (m_{i+1}) and (S_{i+1}) hold again. Assume now that p reduces to a non-zero constant. This means that $\mathbf{Z}(p, U_1) = \emptyset = \mathbf{Z}(F_1, U_1)$. Thus we can delete $\mathbf{Z}(F_1, U_1)$ from (M_i). Then it follows easily that the relations (M_{i+1}), (m_{i+1}) and (S_{i+1}) hold. Now assume that the **for** loop is performed. The conclusion follows by applying Proposition 18 and then Proposition 14. Finally we observe that if L_i is the last iteration of the **while** loop, then Proposition 14 shows that our algorithm returns a Lazard split of $\mathbf{V}(F)$. $\square$

Remark 20 Let p be in $\mathbf{P}_n$ and let $T \subseteq \mathbf{P}_n$ be a regular chain. Note that $([\emptyset, T])$ may never be a delayed split of $\mathbf{Z}(p, T)$ because of condition (D_1). This justifies the input specification of $\mathsf{decompose}(p, T)$ (Specification 1) and the design of Algorithm 2. Observe now that Algorithm 2 can easily be adapted in order to implement an operation $\mathsf{intersect}$ with the following specification: for a polynomial $p \in \mathbf{P}_n$ and a regular chain $T \subseteq \mathbf{P}_n$ such that $p \notin \mathbf{Sat}(T)$ we denote by $\mathsf{intersect}(p, T)$ a sequence $(T_1, \ldots, T_d)$ of regular chains in $\mathbf{P}_n$ such that $\mathbf{Z}(p, T) \longrightarrow_D (\mathbf{W}(T_1), \ldots, \mathbf{W}(T_d))$. Indeed, it suffices to replace the line R

$:= [[F, \emptyset]]$ of Algorithm 2 by $R := [\{p\}, T]$. The termination and the correctness of $\mathsf{intersect}(p, T)$ result from Propositions 10 and 13 respectively. Note that this operation $\mathsf{intersect}$ has weaker specification than the one of Lazard's algorithm. This is because the union of our $\mathbf{W}(T_i)$ may not be contained in the closure of $\mathbf{Z}(p, T)$. However this will never lead us to any trouble (i.e. extraneous solutions) since we shall always use this operation $\mathsf{intersect}$ in the context of Corollary 17 (to compute a regular delayed split of $\mathbf{Z}(h_a, T \cup t)$).

Algorithm 2 reduces problem (L) to the following task. Given a polynomial $p \in \mathbf{P}_n$ and a regular chain $T \subseteq \mathbf{P}_n$ such that $p \notin \mathbf{Sat}(T)$ compute a regular delayed split of the intersection $\mathbf{Z}(p, T)$ of the hypersurface $\mathbf{V}(p)$ and the quasi-component $\mathbf{W}(T)$. We will achieve this task in Sections 4 and 5. First we will show that with additional conditions on p and T, a regular delayed split of $\mathbf{Z}(p, T)$ can be obtained. Then we will provide a way *to replace* T by a regular delayed split $(T_1, \ldots, T_d)$ of T such that p and each T_i satisfy our additional conditions. In order for this strategy to work we need a *composition property*. Proposition 22 will provide the appropriate result after Definition 21.

When computing a regular delayed split of $\mathbf{Z}(p, T)$ we will often be led to consider a *projection*. More precisely, if p has main variable x_{k+1}, we will consider $\mathbf{Z}(p, T) \cap \mathbf{P}_{k+1}$. Therefore we need a way to *lift* the resulting computations from $\mathbf{P}_{k+1}$ to $\mathbf{P}_n$. This is the goal of Theorem 23, concluding this section.

Definition 21 Given a subset F of $\mathbf{P}_n$ and a polynomial $p \in \mathbf{P}_n$ we define the subset $F \cdot p$ of $\mathbf{P}_n$ as follows. If $F = \emptyset$ then $F \cdot p = \emptyset$ otherwise $F \cdot p = F \cup p$.

Proposition 22 *Let $T, T_1, \ldots, T_d$ be triangular sets of $\mathbf{P}_n$ such that $(T_1, \ldots, T_d)$ is a delayed split of T. Let p be a polynomial of $\mathbf{P}_n$ such that $p \notin \mathbf{Sat}(T)$. For every $i = 1 \cdots d$ let $([F_{i,1}, T_{i,1}], \ldots, [F_{i,e_i}, T_{i,e_i}])$ be processes of $\mathbf{P}_n$ such that one of the following conditions holds:*

- (A_i) *$p \in \mathbf{Sat}(T_i)$, $e_i = 1$, $F_{i,1} = \emptyset$ and $T_{i,1} = T_i$,*
- (B_i) *$p \notin \mathbf{Sat}(T_i)$ and $\mathbf{Z}(q_i, T_i) \longrightarrow_D (\mathbf{Z}(F_{i,1}, T_{i,1}), \ldots, \mathbf{Z}(F_{i,e_i}, T_{i,e_i}))$ where q_i is a non-constant polynomial equal to p modulo $\mathbf{Sat}(T_i)$ and with no higher rank than p w.r.t. $\prec$.*

Then we have:

$$\mathbf{Z}(p, T) \longrightarrow_D (\mathbf{Z}(F_{1,1} \cdot p, T_{1,1}), \ldots, \mathbf{Z}(F_{1,e_1} \cdot p, T_{1,e_1}), \ldots, \mathbf{Z}(F_{d,1} \cdot p, T_{d,1}), \ldots, \mathbf{Z}(F_{d,e_d} \cdot p, T_{d,e_d})).$$

PROOF. Since $(T_1, \ldots, T_d)$ is a delayed split of T two cases arise: either $d = 1$ and $T = T_d$ hold or $\mathbf{W}(T) \longrightarrow_D (\mathbf{W}(T_1), \ldots, \mathbf{W}(T_d))$ holds. Proving the proposition in the first case is easy, so we restrict ourselves to the second one.

First, we prove that condition (D_1) holds. Let i be in the range $1 \cdots d$. We can assume that assumption (B_i) holds otherwise there is nothing to prove. Since $(\mathbf{W}(T_1), \ldots, \mathbf{W}(T_d))$ is a delayed split of $\mathbf{W}(T)$, we have $T_i \prec T$ which implies $[p, T_i] \prec [p, T]$. Let j be in the range $1 \cdots e_i$. Since $(\mathbf{Z}(F_{i,1}, T_{i,1}), \ldots, \mathbf{Z}(F_{i,e_i}, T_{i,e_i}))$ is a delayed split of $\mathbf{Z}(q_i, T_i)$ we have $[F_{i,j}, T_{i,j}] \prec [q_i, T_i]$. We need to check that $[F_{i,j} \cdot p, T_{i,j}] \prec [p, T]$ holds. Two cases arise. If $T_{i,j} \prec T_i$, then the result is obvious. If $T_{i,j} \sim T_i$ then $F_{i,j}$ must contain a polynomial smaller w.r.t. $\prec$ than q_i and the result is clear again since q_i has no higher rank than p. Therefore property (D_1) holds. We now prove property (D_2). Since $(\mathbf{W}(T_1), \ldots, \mathbf{W}(T_d))$ is a delayed split of $\mathbf{W}(T)$ we have:

$$\mathbf{W}(T) \ \subseteq \ \cup_{i=1}^{d} \mathbf{W}(T_i). \tag{6}$$

Since $p - q_i \in \mathbf{Sat}(T_i)$ we have $\mathbf{V}(p) \cap \mathbf{W}(T_i) = \mathbf{V}(q_i) \cap \mathbf{W}(T_i)$, leading to:

$$\mathbf{V}(p) \cap \mathbf{W}(T_i) \ \subseteq \ \cup_{j=1}^{e_i} \mathbf{Z}(F_{i,j}, T_{i,j}). \tag{7}$$

In relation (7) we can replace every term $\mathbf{Z}(F_{i,j}, T_{i,j})$ such that $F_{i,j} \neq \emptyset$ by $\mathbf{V}(p) \cap \mathbf{Z}(F_{i,j}, T_{i,j})$. Hence relations (6) and (7) lead to (D_2). Since $\mathbf{Sat}(T) \subseteq \mathbf{Sat}(T_i)$ and $\mathbf{Sat}(T_i) \subseteq \mathbf{Sat}(T_{i,j})$ hold property (D_3) is clear. Since $F_{i,j} \cdot p \neq \emptyset$ implies $p \in F_{i,j} \cdot p$ property (D_4) is clear. We prove (D_5). We assume $F_{i,j} \cdot p = \emptyset$ which implies $F_{i,j} = \emptyset$. If (A_i) holds, then we have $\mathbf{W}(T_{i,j}) \subseteq \mathbf{V}(p)$ since $T_{i,j} = T_i$ and $p \in \mathbf{Sat}(T_i)$. If (B_i) holds, then we have also $\mathbf{W}(T_{i,j}) \subseteq \mathbf{V}(q_i)$ because (D_5) holds for the delayed split $(\mathbf{Z}(F_{i,1}, T_{i,1}), \ldots, \mathbf{Z}(F_{i,e_i}, T_{i,e_i}))$ of $\mathbf{Z}(q_i, T_i)$. Moreover we have $\mathbf{W}(T_{i,j}) \subseteq \mathbf{V}(p - q_i)$ since $p - q_i \in \mathbf{Sat}(T_i)$ and $\mathbf{Sat}(T_i) \subseteq \mathbf{Sat}(T_{i,j})$ hold. Thus $\mathbf{W}(T_{i,j}) \subseteq \mathbf{V}(p)$ holds in any case, leading to (D_5). □

Theorem 23 (Lifting Property) *Let k be an integer such that $0 \leq k < n$. Let $C \subseteq \mathbf{P}_k$ and $D \subseteq \mathbf{P}_n$ such that $C \cup D$ is a regular chain, the set D is not empty and $D \cap \mathbf{P}_k = \emptyset$ holds. Let h_C the product of the initials of C. Let F be a finite subset of $\mathbf{P}_n$. We assume that the following conditions hold:*

(i) there exist regular chains $C_1, \ldots, C_d$ of $\mathbf{P}_k$ and finite subsets $F_1, \ldots, F_d$ of $\mathbf{P}_n$ such that

$$\mathbf{Z}(F, C) \longrightarrow_D (\mathbf{Z}(F_1, C_1), \ldots, \mathbf{Z}(F_d, C_d)),$$

(ii) for every $i = 1 \cdots d$ there exist regular chains $C_{i,1}, \ldots C_{i,e_i}$ such that

$$C_i \cup D \longrightarrow_D (C_{i,1}, \ldots C_{i,e_i}),$$

(iii) for every $i = 1 \cdots d$ and every $j = 1 \cdots e_i$ the polynomial h_C is regular w.r.t. $C_{i,j}$.

Then we have:

$$\mathbf{Z}(F, C \cup D) \longrightarrow_D (\mathbf{Z}(F_1, C_{1,1}), \ldots, \mathbf{Z}(F_1, C_{1,e_1}), \ldots, \mathbf{Z}(F_d, C_{d,1}), \ldots, \mathbf{Z}(F_d, C_{d,e_d})).$$

PROOF. Let i be in the range $1 \cdots d$ and j be in $1 \cdots e_i$. First we prove $[F_i, C_{i,j}] \prec [F, C \cup D]$. From (i) we have $[F_i, C_i] \prec [F, C]$. From (ii) we have either $C_{i,j} \sim C_i \cup D$ or $C_{i,j} \prec C_i \cup D$. Two cases arise. If $C_i \prec C$ holds then we have $C_i \cup D \prec C \cup D$ leading to $C_{i,j} \prec C \cup D$ and thus to the desired relation. Assume now that $C_i \sim C$ holds. This leads to $C_i \cup D \sim C \cup D$. If $C_{i,j} \prec C_i \cup D$ we are done again. Otherwise $C_{i,j} \sim C \cup D$ holds. Moreover, F_i must contain a polynomial with smaller rank than any polynomial of F. Hence $[F_i, C_{i,j}] \prec [F, C \cup D]$ holds in any case and property (D_1) is proved. Now we prove property (D_2). From (i) we have $\mathbf{Z}(F, C) \subseteq \cup_{i=1}^{d} \mathbf{Z}(F_i, C_i)$. This leads to $\mathbf{Z}(F, C \cup D) \subseteq \cup_{i=1}^{d} \mathbf{Z}(F_i, C_i \cup D)$. Indeed one can easily check that $\mathbf{W}(C \cup D) = \mathbf{W}(C) \cap \mathbf{W}(D)$ and $\mathbf{W}(C_i \cup D) = \mathbf{W}(C_i) \cap \mathbf{W}(D)$ hold for $i = 1 \cdots d$. Now, from (ii) we have $\mathbf{W}(C_i \cup D) \subseteq \cup_{j=1}^{e_i} \mathbf{W}(C_{i,j})$. Therefore we obtain

$$\mathbf{Z}(F, C \cup D) \subseteq \cup_{i=1}^{d} \cup_{j=1}^{e_i} \mathbf{Z}(F_i, C_{i,j}).$$

Now we prove property (D_3). From (i) we have $\mathbf{Sat}(C) \subseteq \mathbf{Sat}(C_i)$. Assume that we have $\mathbf{Sat}(C \cup D) \subseteq \mathbf{Sat}(C_i \cup D)$. Then we conclude with (ii) by using $\mathbf{Sat}(C_i \cup D) \subseteq \mathbf{Sat}(C_{i,j})$. Note that this shows that $\mathbf{Sat}(C_i \cup D) \neq \mathbf{P}_n$ (since $C_{i,j}$ is a regular chain). So we check now that we have $\mathbf{Sat}(C \cup D) \subseteq \mathbf{Sat}(C_i \cup D)$ (although $C_i \cup D$ is not necessarily a regular chain). Let h_D and h_{C_i} be the product of the initials of D and C_i respectively. Let f be in $\mathbf{Sat}(C \cup D)$. There exists a non-negative integer m, a polynomial $f_C \in \mathcal{I}(C)$ and a polynomial $f_D \in \mathcal{I}(D)$ such that we have: $(h_C\, h_D)^m\, f = f_C + f_D$. Since $\mathcal{I}(C) \subseteq \mathbf{Sat}(C)$ and $\mathbf{Sat}(C) \subseteq \mathbf{Sat}(C_i)$ there exists a non-negative integer m' and a polynomial $f_i \in \mathcal{I}(C_i)$ such that ${h_{C_i}}^{m'}\, f_C = f_i$. We obtain:

$${h_{C_i}}^{m'}\, (h_C\, h_D)^m\, f = f_i + {h_{C_i}}^{m'}\, f_D.$$

This shows that ${h_C}^m\, f \in \mathbf{Sat}(C_i \cup D)$ holds. Now observe that (ii) implies that the radical of $\mathbf{Sat}(C_i \cup D)$ is equal to the intersection of the radicals of the $\mathbf{Sat}(C_{i,j})$. Thus every prime ideal associated with $\mathbf{Sat}(C_i \cup D)$ is a prime ideal associated with one $\mathbf{Sat}(C_{i,j})$ (by uniqueness of a minimal primary decomposition). Hence with (iii) it follows that h_C is regular modulo $\mathbf{Sat}(C_i \cup D)$. Therefore $f \in \mathbf{Sat}(C_i \cup D)$ holds and property (D_3) is proved. We remark that property (D_4) is clearly satisfied. Finally we prove property (D_5). So we assume $F_i = \emptyset$. With (i) this implies $\mathbf{W}(C_i) \subseteq \mathbf{V}(F)$. This leads to $\mathbf{W}(C_i \cup D) \subseteq \mathbf{V}(F)$ and thus to $\overline{\mathbf{W}(C_i \cup D)} \subseteq \mathbf{V}(F)$. From (ii) we have $\mathbf{Sat}(C_i \cup D) \subseteq \mathbf{Sat}(C_{i,j})$ and thus $\overline{\mathbf{W}(C_{i,j})} \subseteq \overline{\mathbf{W}(C_i \cup D)}$ leading to $\mathbf{W}(C_{i,j}) \subseteq \mathbf{V}(F)$ and proving (D_5). □

4 Intersecting Hypersurfaces with Quasi-Components

Let $T \subseteq \mathbf{P}_n$ be a regular chain and let $p \in \mathbf{P}_n$ be a polynomial satisfying $p \notin \mathbf{k}$ and $p \notin \mathbf{Sat}(T)$. The goal of this section is to show how a regular delayed split of the intersection $\mathbf{Z}(p, T)$ of the hypersurface $\mathbf{V}(p)$ and the quasi-component $\mathbf{W}(T)$ can be obtained. If the main variable of p is not algebraic w.r.t. T, we will use the following proposition whose proof is straightforward.

Proposition 24 *Let $T \subseteq \mathbf{P}_n$ be a regular chain and let $p \in \mathbf{P}_n$ be a polynomial satisfying $p \notin \mathbf{k}$ and $p \notin \mathbf{Sat}(T)$. We assume that $\mathsf{mvar}(p)$ is not algebraic w.r.t. T and we denote by h_p the initial of p. Let $(T_1, \ldots, T_d)$ be a regular delayed split of $T \cup p$. Then we have:*

$$\mathbf{Z}(p, T) \longrightarrow_D (\mathbf{Z}(\{h_p, p\}, T), \mathbf{W}(T_1), \ldots, \mathbf{W}(T_d)).$$

The above triangular set $T \cup p$ may not be a regular chain. Hence we need a procedure for building a regular delayed split $(T_1, \ldots, T_d)$ of $T \cup p$. To do so we assume from now on that we are given an operation regular? defined in Specification 2. The validity of this assumption will be established in Section 5. The operation regular? leads to the operation extend defined in Algorithm 3.

Specification 2 *For any polynomial $f \in \mathbf{P}_n$ and any regular chain $C \subseteq \mathbf{P}_n$, the operation $\mathsf{regular?}(f, C)$ returns a regular delayed split $(C_1, \ldots, C_d)$ of C such that for every $i = 1 \cdots d$ if $\mathsf{red}(f, C_i) \neq 0$ then f is regular w.r.t. C_i.*

Algorithm 3

- **Input:** a regular chain C contained in $\mathbf{P}_k$, for an integer k in the range $0 \cdots n-1$, and a non-constant $p \in \mathbf{P}_n$ with $\mathsf{mvar}(p) = x_{k+1}$.
- **Output:** a regular delayed split of $C \cup p$.
- $\mathsf{extend}(C, p)$ ==
  $h := \mathsf{init}(p)$
  Let h_C be the product of the initials of C
  for $D \in \mathsf{regular?}(h, C)$ **repeat**
    $\mathsf{red}(h, D) = 0 \Longrightarrow$ **iterate**
    $\mathsf{dim}(D) = \mathsf{dim}(C) \Longrightarrow$ **output** $D \cup p$
    for $E \in \mathsf{regular?}(h_C\, h, D)$ **repeat**
      $\mathsf{red}(h_C\, h, E) \neq 0 \Longrightarrow$ **output** $E \cup p$
  exit

We will see that each of the algorithms presented in Sections 3, 4 and 5 may call each other directly or indirectly. In order to establish their termination and correctness we need to point out on which hypothesis each algorithm calls another. Proposition 26 states that Algorithm 3 terminates and satisfies its

specification provided that its calls to the operation regular? are valid.

Hypothesis 25 *Let C and p be as in the input specification of Algorithm 3. We denote by $\mathcal{H}_1(C,p)$ the following hypothesis: for every regular chain $R \subseteq \mathbf{P}_n$ and for every polynomial $f \in \mathbf{P}_n$ such that $[f,R] \prec [p,C]$ holds, the call* regular?(f,R) *satisfies Specification 2.*

Proposition 26 *Let C and p be as in the input of Algorithm 3. If hypothesis $\mathcal{H}_1(C,p)$ holds then Algorithm 3 terminates and satisfies its output specification. Moreover $\mathbf{Sat}(C \cup p)$ is the unit ideal iff* extend(C,p) *is empty.*

PROOF. We first show that the hypothesis of Proposition 26 applies to every call to the operation regular? in Algorithm 3. As in Algorithm 3 we denote by h the initial of p and by h_C the product of the initials of C. Since $h \prec p$ we have $[h,C] \prec [p,C]$. Thus, since $\mathcal{H}_1(C,p)$ holds, we can assume that regular?(h,C) returns a regular delayed split $(C_1,\ldots,C_d,{C'}_1,\ldots,{C'}_e)$ of C such that for every $i = 1 \cdots d$ the polynomial h is regular w.r.t. C_i, and for every $j = 1 \cdots e$ the polynomial h reduces to zero w.r.t. ${C'}_j$. We observe that every regular chain ${C'}_1,\ldots,{C'}_e$ is discarded. Then the regular chains $C_1,\ldots,C_d$ are separated in two groups: those satisfying $\mathsf{dim}(C_i) = \mathsf{dim}(C)$ and the others. We denote $D_1,\ldots,D_a$ the former ones and ${D'}_1,\ldots,{D'}_b$ the latter ones. The algorithm returns $D_1 \cup p,\ldots,D_a \cup p$ and calls regular?$(h_C\, h, D)$ for every $D = {D'}_1,\ldots,{D'}_b$. By Definition 15 we have either ${D'}_i \sim C$ or ${D'}_i \prec C$ for $i = 1 \cdots b$. Thus for every $D = {D'}_1,\ldots,{D'}_b$ the call regular?$(h_C\, h, D)$ satisfies Specification 2. Therefore every call to regular? of Algorithm 3 satisfies Specification 2 and every returned set is a regular chain. Moreover, Algorithm 3 terminates. We prove now that all these returned sets form a delayed split of the input $C \cup p$.

Checking properties (D_1), (D_4) and (D_5) is easy. We prove (D_2). By Specification 2 and Definition 15 the union of the quasi-components of ${C'}_1,\ldots,{C'}_e$, $D_1,\ldots,D_a$, ${D'}_{1,1},\ldots,{D'}_{b,c_b}$ contains $\mathbf{W}(C)$. Since either h or h_C is zero modulo the saturated ideal of every discarded regular chain, the quasi-component $\mathbf{W}(C \cup p)$ is contained in the union of the quasi-components of the returned regular chains. This proves property (D_2). Now we prove (D_3). Observe that h_C is regular w.r.t. every returned regular chain. Indeed this holds for $D_1 \cup p,\ldots,D_a \cup p$ by Proposition 5. This holds also for the regular chains ${D'}_{i,j} \cup p$ such that $\mathsf{red}(h_C\, h, {D'}_{i,j}) \neq 0$. by Specification 2. Therefore Theorem 23 applies. This proves (D_3). Observe that if Algorithm 3 does not return any regular chain, then $\overline{\mathbf{W}(C)} \subseteq \mathbf{V}(h)$ holds. Hence we have $\mathbf{W}(C \cup p) = \emptyset$, leading to $\mathbf{Sat}(C \cup p) = \mathbf{P}_n$. On the contrary, if Algorithm 3 returns at least one regular chain R, then $\mathbf{Sat}(C \cup p) \subseteq \mathbf{Sat}(R)$ holds showing that $\mathbf{Sat}(C \cup p) \neq \mathbf{P}_n$. □

Algorithm 4 generalizes the operation extend to the case where its second argument is a triangular set. This will provide us the tool for lifting computations in the context of Theorem 23. Proposition 28 states that Algorithm 4 terminates and satisfies its specification, provided that its calls to the operation regular? are valid. Proposition 28 follows from Proposition 26 and its proof is not reported here.

Algorithm 4

- **Input:** C and D as in the statement of Theorem 23 and let p be the element of D with smallest main variable.
- **Output:** a regular delayed split of $C \cup D$.
- $\mathsf{extend}(C, D)$ ==
  $D^+ := D \setminus p$
  for $E \in \mathsf{extend}(C, p)$ **repeat**
    $D^+ = \emptyset \Longrightarrow$ **output** E
    for $F \in \mathsf{extend}(E, D^+)$ **repeat output** F
  exit

Hypothesis 27 *Let $T \subseteq \mathbf{P}_n$ be a triangular set and let m be an integer in the range $1 \cdots m$. We denote by $\mathcal{H}_2(T, m)$ the following hypothesis: for every regular chain $R \subseteq \mathbf{P}_{m-1}$ and for every polynomial $f \in \mathbf{P}_{m-1}$ such that R has no higher rank than $T^-_{x_m}$, the call* regular?(f, R) *satisfies Specification 2.*

Proposition 28 *Let C and D be as in the input of Algorithm 4. Let m be the smallest index such that $C \cup D \subseteq \mathbf{P}_m$ holds. If hypothesis $\mathcal{H}_2(C \cup D, m)$ holds then Algorithm 4 terminates and satisfies its output specification. Moreover $\mathbf{Sat}(C \cup D)$ is the unit ideal of $\mathbf{P}_n$ iff* extend(C, D) *does not return any regular chain. In addition, for any regular chain R returned by* extend(C, D) *the product of the initials of $C \cup D$ is regular w.r.t. R.*

When the hypothesis of Proposition 28 is satisfied, we have an algorithm to compute a delayed split of $\mathbf{Z}(p, T)$ in the context of Proposition 24. So we consider now the case where the main variable of p is algebraic w.r.t. T. The following concept will be our main tool in this new context.

Definition 29 *Let $T \subseteq \mathbf{P}_n$ be a regular chain. Let k be an integer in the range $0 \cdots n-1$. Let p and t be two non-constant polynomials in $\mathbf{P}_n$ with main variable x_{k+1} such that the initial of p is regular w.r.t. T and $T \cup t$ is a regular chain. Let g be a polynomial in $\mathbf{P}_{k+1}$. We say that g is a* regular gcd *of the polynomials p and t* modulo *T if the following conditions hold:*

(G_1) *the leading coefficient of g as a univariate polynomial in $\mathbf{P}_k[x_{k+1}]$ is regular w.r.t. T,*
(G_2) *there exist polynomials $u, v \in \mathbf{P}_{k+1}$ such that $ut + vp - g \in \mathbf{Sat}(T) \cap \mathbf{P}_k$,*
(G_3) *if $g \notin \mathbf{k}$ and $\mathsf{mvar}(g) = x_{k+1}$ then both t and p lye in $\mathbf{Sat}((T \cap \mathbf{P}_k) \cup g)$.*

If g is not constant and has main variable x_{k+1}, then we say that g is a non-trivial regular gcd *of p and t* modulo T.

Remark 30 Let T, p, t, g be as in Definition 29 such that g is a regular gcd of t and p modulo T. We point out three obvious remarks that are used quite often in practice. First observe that condition (G_1) implies $g \neq 0$. Secondly, condition (G_2) shows that if $g \in \mathbf{k}$ holds then we have $\mathbf{Z}(\{p, t\}, T) = \emptyset$. Lastly, condition (G_3) implies: $\mathsf{rank}(g) = \mathsf{rank}(t) \implies p \in \mathbf{Sat}(T \cup t)$. We prove this statement. We define $h = \mathsf{init}(t)$ and $h_g = \mathsf{init}(g)$. There exist non-negative integers e and e', polynomials $q_t, q_f \in \mathbf{P}_{k+1}$ and polynomials $r_t, r_f \in \mathbf{Sat}(T) \cap \mathbf{P}_k$ such that ${h_g}^e t = q_t g + r_t$ and ${h_g}^{e'} p = q_p g + r_p$. This leads to $q_t {h_g}^{e'} p = q_p {h_g}^e t + q_t r_p - q_p r_t$. Since $\mathsf{mdeg}(t) = \mathsf{mdeg}(g)$ the relation ${h_g}^e t = q_t g + r_t$ shows that $q_t \in \mathbf{P}_k$ holds, which leads to ${h_g}^e h_t = q_t h_g$. Thus $q_t {h_g}^{e'}$ is regular w.r.t. $T \cap \mathbf{P}_k$. It follows with relation (3) p. 7 that p lies in $\mathbf{Sat}((T \cap \mathbf{P}_k) \cup t)$. Therefore $p \in \mathbf{Sat}(T \cup t)$. This last property shows that if $\mathsf{mvar}(g) = x_{k+1}$ and $p \notin \mathbf{Sat}(T \cup t)$ hold then we have $\mathsf{mdeg}(g) < \mathsf{mdeg}(t)$. Indeed $\mathsf{mdeg}(g) > \mathsf{mdeg}(t)$ and (G_3) implies $\mathsf{prem}(t, T) = 0$ contradicting the fact that h is regular w.r.t. T.

Lemma 31 *Let T, p, t, g be as in Definition 29. Assume that $p \notin \mathbf{Sat}(T \cup t)$ and that g is a non-trivial regular gcd of t and p modulo T. Let h_g be the initial of g and q be the pseudo-quotient of t by g. Then $T \cup q$ is a regular chain with $\mathsf{mvar}(q) = x_{k+1}$. Moreover we have:*

(1) $\mathbf{W}(T \cup g) \subseteq \mathbf{Z}(\{p, t\}, T)$,
(2) $\mathbf{Z}(p, T \cup q) \setminus \mathbf{V}(h_g) \subseteq \mathbf{W}(T \cup g)$.

PROOF. By assumption we have $\mathsf{mvar}(g) = x_{k+1}$. Since $p \notin \mathbf{Sat}(T \cup t)$ holds, Remark 30 implies $\mathsf{mdeg}(g) < \mathsf{mdeg}(t)$. Hence $\mathsf{mvar}(q) = x_{k+1}$ holds. Since the initial of t is regular w.r.t. T, conditions (G_1) and (G_3) allow us to apply Theorem 7 and we deduce that $T \cup g$ and $T \cup q$ are regular chains. Now, let us prove (1). Condition (G_3) implies $p \in \mathbf{Sat}(T \cup g)$ and $t \in \mathbf{Sat}(T \cup g)$. Thus we have $\mathbf{V}(p) \cap \mathbf{V}(t) \supseteq \overline{\mathbf{W}(T \cup g)}$, proving (1). We prove (2). From condition (G_2) there exist polynomials $u, v \in \mathbf{P}_{k+1}$ and a polynomial $r' \in \mathbf{Sat}(T)$ such that $g = ut + vp + r'$. From condition (G_3), there exists a polynomial $r'' \in \mathbf{Sat}(T)$ and a non-negative integer m such that ${h_g}^m t = qg + r''$. Thus we have:

$$ {h_g}^m g = u(qg + r'') + {h_g}^m (vp + r'). $$

Since $\mathbf{W}(T) \subseteq \mathbf{V}(\mathbf{Sat}(T))$ any point ζ of $\mathbf{Z}(p, T \cup q)$ cancels the right hand side of the above equality and thus belongs to $\mathbf{V}(h_g\, g)$, leading to (2). □

Theorem 32 *Let T, p, t, g be as in Definition 29. Assume that $p \notin \mathbf{Sat}(T \cup t)$ and that g is a non-trivial regular gcd of t and p modulo T. Let h_g be the initial*

of g. Then we have:

$$\mathbf{Z}(p, T \cup t) \longrightarrow_D (\mathbf{W}(T \cup g),\ \mathbf{Z}(\{h_g, p\}, T \cup t)). \tag{8}$$

PROOF. Let us prove that properties (D_1) to (D_5) of Definition 11 hold. Recall that our assumptions imply $\mathsf{mvar}(g) = \mathsf{mvar}(t)$ and $\mathsf{mdeg}(g) < \mathsf{mdeg}(t)$. Since $g \prec t$ and $h_g \prec p$ hold each of the processes $[\emptyset, T \cup g]$ and $[\{h_g, p\}, T \cup t]$ has smaller rank than the process $[\{p\}, T \cup t]$. This proves (D_1). As in Lemma 31 let q be the pseudo-quotient of t w.r.t. g. From Theorem 7 we know that $T \cup g$ and $T \cup q$ are regular chains. Moreover we have:

$$\mathbf{Z}(p, T \cup t) \ \subseteq\ \mathbf{W}(T \cup g) \ \cup\ \mathbf{Z}(p, T \cup q) \ \cup\ \mathbf{Z}(\{h_g, p\}, T \cup t).$$

Indeed, Relation (1) of Lemma 31 shows that $\mathbf{Z}(p, T \cup g) = \mathbf{W}(T \cup g)$. Observe that all the points of $\mathbf{Z}(p, T \cup t)$ that cancel h_g are already captured by $\mathbf{Z}(\{h_g, p\}, T \cup t)$ and we replace the term $\mathbf{Z}(p, T \cup q)$ by $\mathbf{Z}(p, T \cup q) \setminus \mathbf{V}(h_g)$ in the above union. Hence with relation (2) of Lemma 31 we obtain (D_2). Property (D_3) follows from Lemma 6 and properties (D_4) and (D_5) are clear. □

We will show in Section 5 that there exists an operation called *lastSubResultant*, denoted for short by lsr and defined in Specification 3. The operation lsr together with Theorems 23 and 32 leads us to a first algorithm for the operation $\mathsf{decompose}$ in a restricted case, given below in Algorithm 5. This specialized $\mathsf{decompose}$ takes three arguments whereas the general one (Algorithm 7) will only have two, Proposition 34 states that Algorithm 5 terminates and satisfies its specifications provided that its call to $\mathsf{regular?}$ and lsr are valid.

Specification 3 *Let k be an integer in the range $0 \cdots n-1$. Let $T \subseteq \mathbf{P}_k$ be a regular chain. Let $p_1, p_2 \in \mathbf{P}_{k+1}$ be non-constant polynomials such that $\mathsf{mvar}(p_1) = \mathsf{mvar}(p_2) = x_{k+1}$ and $\mathsf{mdeg}(p_1) \geq \mathsf{mdeg}(p_2)$ hold. Assume that $\mathsf{init}(p_1)$ and $\mathsf{init}(p_2)$ are regular w.r.t. T. The operation $\mathsf{lsr}(p_1, p_2, T)$ returns a sequence of processes $([g_1, C_1], \ldots [g_d, C_d],\ [\emptyset, D_1], \ldots [\emptyset, D_e])$ of $\mathbf{P}_n$ where $C_1, \ldots, C_d, D_1, \ldots, D_e$ are regular chains of $\mathbf{P}_k$ and $g_1, \ldots, g_d$ are polynomials of $\mathbf{P}_{k+1}$ satisfying the following conditions:*

(L_1) $T \longrightarrow_D (C_1, \ldots, C_d, D_1, \ldots, D_e)$,
(L_2) *for every $i = 1 \cdots d$ the polynomial g_i is a regular gcd of p_1 and p_2 modulo C_i and we have $\mathsf{dim}(C_i) = \mathsf{dim}(T)$,*
(L_3) *for every $j = 1 \cdots e$ we have $\mathsf{dim}(D_j) < \mathsf{dim}(T)$.*

Algorithm 5

- **Input:** T, p, t, g be as in Definition 29 such that $p \notin \mathbf{Sat}(T \cup t)$ and define $v = \mathsf{mvar}(p)$.

- **Output:** a regular delayed split of $\mathbf{Z}(p, T \cup t)$.
- $\mathsf{decompose}(p, T, t) ==$

  Let h_v^- be the product of the initials of T_v^-
  if $\mathsf{mdeg}(p) < \mathsf{mdeg}(t)$ **then** $(p_1, p_2) := (t, p)$ **else** $(p_1, p_2) := (p, t)$
  for $[G, C] \in \mathsf{lsr}(p_1, p_2, T_v^-)$ **repeat**
    $G = \emptyset \Longrightarrow$
      for $D \in \mathsf{extend}(C, T_v^+ \cup h_v^- t)$ **repeat output** $[p, D]$
    $\{g\} := G$
    $g \in \mathbf{k} \Longrightarrow$ **iterate**
    $\mathsf{mvar}(g) < v \Longrightarrow$ **output** $[\{g, p\}, C \cup t \cup T_v^+]$
    $\mathsf{mdeg}(g) = \mathsf{mdeg}(t) \Longrightarrow$ **output** $[\emptyset, C \cup t \cup T_v^+]$
    output $[\emptyset, C \cup g \cup T_v^+]$
    output $[\{\mathsf{init}(g), p\}, C \cup t \cup T_v^+]$
  exit

Hypothesis 33 *Let $T \subseteq \mathbf{P}_n$ be a triangular set and let m be an integer in the range $1 \cdots n$. We denote by $\mathcal{H}_3(T, m)$ the following hypothesis: for every regular chain $R \subseteq \mathbf{P}_{m-1}$ with no higher rank than $T_{x_m}^-$, every call to* lsr *with R as third argument satisfies Specification 3.*

Proposition 34 *Let T, p, t be as in the input of Algorithm 5. Let m be the smallest index such that $T \cup t \subseteq \mathbf{P}_m$ holds. If hypothesis $\mathcal{H}_2(T \cup t, m)$ and $\mathcal{H}_3(T \cup t, m)$ hold then Algorithm 5 terminates and satisfies its specification. Moreover for every process $[F, R]$ returned by $\mathsf{decompose}(p, T, t)$ such that $\mathsf{dim}(R) < \mathsf{dim}(T \cup t)$ holds we have $F \neq \emptyset$.*

PROOF. We assume that $\mathcal{H}_2(T \cup t, m)$ and $\mathcal{H}_3(T \cup t, m)$ hold. Let h be the initial of t. First we prove that the calls to the operations lsr and extend satisfy Specification 3 and the specifications of Algorithm 4 respectively. This implies immediately that Algorithm 5 terminates. Since $T_v^- \subseteq \mathbf{P}_{m-1}$ holds the call $\mathsf{lsr}(p_1, p_2, T_v^-)$ satisfies Specification 3. Hence every regular chain C computed by $\mathsf{lsr}(p_1, p_2, T_v^-)$ has no higher rank than T_v^-. By Proposition 28 it follows that $\mathsf{extend}(C, T_v^+ \cup h_v^- t)$ returns a regular delayed split $(D_1, \ldots, D_e)$ of $C \cup T_v^+ \cup h_v^- t$ such that $h_v^- h$ is regular modulo D_i for every $i = 1 \cdots e$. This proves our claim. Now we show that Algorithm 5 is correct. First we assume $T_v^+ = \emptyset$. We claim that in this case Algorithm 5 returns a regular delayed split of $\mathbf{Z}(p, T \cup h_v^- t)$. To see this let $(C_1, \ldots, C_d, C'_1, \ldots, C'_e)$ be the regular delayed split of T_v^- returned by $\mathsf{lsr}(p_1, p_2, T_v^-)$ such that for every $i = 1 \cdots d$ we have $\mathsf{dim}(C_i) < \mathsf{dim}(T_v^-)$ and for every $j = 1 \cdots e$ we have $\mathsf{dim}(C'_j) = \mathsf{dim}(T_v^-)$. Note that the second **for** loop computes for $i = 1 \cdots d$ a regular delayed split (possibly empty) of $C_i \cup h_v^- t$. By Proposition 5, for $j = 1 \cdots e$ the triangular set $C'_j \cup h_v^- t$ is a regular chain. Then, by applying the formulas of Remark 30 and Theorem 32 we complete easily the proof of our claim. We return now to the case where T_v^+ may not be empty. Since $h_v^- h$

is regular w.r.t. every returned set, it suffices to apply Theorem 23. □

We are now ready to state an algorithm for the operation decompose in the general case. It consists essentially in a preparation phase by means of the operation regularizeInitial defined in Algorithm 6 so that we can apply either Proposition 24 (the *non-algebraic case*) or Proposition 34 (the *algebraic case*).

Algorithm 6

- **Input:** $f \in \mathbf{P}_n$ a polynomial and $C \subseteq \mathbf{P}_n$ a regular chain.
- **Output:** a regular delayed split $(C_1, \ldots, C_d)$ of C such that the polynomial $\mathsf{red}(f, C_i)$ is either null or regular w.r.t. C_i, for every $i = 1 \cdots d$.
- $\mathsf{regularizeInitial}(f, C)$ ==
  $f := \mathsf{red}(f, C)$
  $R := [[f, C]]$
  # R is a list of processes
  while $R \neq [\,]$ **repeat**
    choose and remove a process $[f, C]$ from R
    $f \in \mathbf{k} \Longrightarrow$ **output** C
    for $D \in \mathsf{regular?}(\mathsf{init}(f), C)$ **repeat**
      $\mathsf{red}(\mathsf{init}(f), D) \neq 0 \Longrightarrow$ **output** D
      $R := \mathsf{cons}([\mathsf{red}(\mathsf{tail}(f), D), D])$
  exit

Proposition 35 *Let C and f be as in the input of Algorithm 6. If hypothesis $\mathcal{H}_1(C, f)$ holds then Algorithm 6 terminates and satisfies its specification.*

Algorithm 7

- **Input:** T regular chain of $\mathbf{P}_n$ and $p \in \mathbf{P}_n$ such that $p \notin \mathbf{k}$ and $p \notin \mathbf{Sat}(T)$.
- **Output:** a regular delayed split of $\mathbf{Z}(p, T)$.
- $\mathsf{decompose}(p, T)$ ==
  for $C \in \mathsf{regularizeInitial}(p, T)$ **repeat**
    $f := \mathsf{red}(p, C)$
    $f = 0 \Longrightarrow$ **output** $[\emptyset, C]$
    $f \in \mathbf{k} \Longrightarrow$ **iterate**
    $v := \mathsf{mvar}(f)$
    $v \notin \mathsf{alg}(C) \Longrightarrow$
      output $[\{\mathsf{init}(f), p\}, C]$
      for $D \in \mathsf{extend}(\ C_v^- \cup f,\ C_v^+)$ **repeat output** $[\emptyset, D]$
    for $[F, E] \in \mathsf{decompose}(f,\ C_v^- \cup\ C_v^+,\ C_v)$ **repeat output** $[F \cdot p, E]$
  exit

Proposition 36 *Let T and p be as in the input specification of Algorithm 7. Let m be the smallest index such that $T \subseteq \mathbf{P}_m$ holds. If hypothesis $\mathcal{H}_1(T, p)$,*

$\mathcal{H}_2(T, m)$ and $\mathcal{H}_3(T, m)$ hold then Algorithm 7 terminates and satisfies its specification. Moreover for every process $[F, R]$ returned by $\mathsf{decompose}(p, T)$ *such that* $\mathsf{dim}(R) < \mathsf{dim}(T \cup t)$ *holds we have* $F \neq \emptyset$.

PROOF. We assume that $\mathcal{H}_1(T, p)$, $\mathcal{H}_2(T, m)$ and $\mathcal{H}_3(T, m)$ hold. Since $\mathcal{H}_1(T, p)$ holds, it follows from Proposition 35 that $\mathsf{regularizeInitial}(p, T)$ returns a regular delayed split of T. We claim that for each regular chain C produced by $\mathsf{regularizeInitial}(p, T)$ the corresponding iteration of the **for** loop returns either C or a regular delayed split of $\mathbf{Z}(p, C)$. Therefore the correctness of the whole algorithm follows from Proposition 22. Let us prove this claim. So let C be a regular chain produced by $\mathsf{regularizeInitial}(p, T)$. Recall that C has not higher rank than T. As in Algorithm 7 we put $f = \mathsf{red}(p, C)$. If $f = 0$ or $f \in \mathbf{k}$, the conclusion is clear. Let $v = \mathsf{mvar}(f)$. Note that $\mathsf{init}(f)$ is regular w.r.t. C. Moreover C_v^- has not higher rank than T_v^-. First assume that $v \notin \mathsf{alg}(C)$. Since $\mathcal{H}_2(T, m)$ holds we can apply Proposition 28. Then the conclusion follows from Proposition 24. Now Assume $v \in \mathsf{alg}(C)$. Since $\mathcal{H}_2(T, m)$ and $\mathcal{H}_3(T, m)$ hold we can apply Proposition 34. This proves our above claim. Finally we observe that the last statement of Proposition 34 is clear. □

5 Computations Modulo Regular Chains

The goal of this section is to establish algorithms for the operations regular? and lastSubResultant introduced in Specifications 2 and 3 respectively. For the latter operation we will rely on subresultant theory. We start this section by a brief review of this subject and refer to (Loos, 1982; Ducos, 2000) for more details.

Let $\mathbf{A}$ be an integral domain. For $p_1 \in \mathbf{A}[X]$ with $p_1 \neq 0$ we denote by $\mathsf{lc}(p_1)$ and $\mathsf{deg}(p_1)$ the leading coefficient and the degree of p_1. If $\mathsf{deg}(p_1) > 0$, for $p_2 \in \mathbf{A}[X]$ we denote by $\mathsf{prem}(p_1, p_2)$ the pseudo-remainder of p_1 by p_2. From now on we assume that p_1 and p_2 have positive degrees d_1 and d_2 such that $d_1 \geq d_2$. Recall that for $j = 0 \cdots d_2$ the j-th subresultant S_j of p_1 and p_2 lies in $\mathbf{A}[X]$ and that it is either null or satisfies $\mathsf{deg}(S_j) \leq j$; we say that S_j is *regular* if $\mathsf{deg}(S_j) = j$. Let S be a non-zero subresultant of p_1 and p_2 with degree $d < d_2$. From the *Block Structure Theorem* of subresultants (Loos, 1982) there exist at most two subresultants with the same degree d: that with biggest index is denoted by S^+ and that with smallest index is denoted by S^-. Moreover S^- is regular. We denote by $S^<$ (resp. $S^>$) the non-zero subresultant S_k with biggest (resp. smallest) index k whose degree is strictly less (resp. greater) than d. The non-zero subresultant of p_1 and p_2 with smallest index is denoted by $\mathsf{lsr}(p_1, p_2)$ and called the *last-subresultant* of p_1 and p_2. If S_ℓ is the last-subresultant of p_1 and p_2, the subresultant S_ℓ^+

is denoted by $\mathsf{srgcd}(p_1, p_2)$ and called the *subresultant-gcd* of p_1 and p_2. If $\mathsf{lsr}(p_1, p_2)$ has degree zero, then it is the resultant of p_1 and p_2. We will use the following fact. The polynomial $\mathsf{lsr}(p_1, p_2)$ lies in the ideal generated by p_1 and p_2 in $\mathbf{A}[X]$. Moreover p_1 and p_2 reduces to zero w.r.t. $\mathsf{lsr}(p_1, p_2)$ by pseudo-division. Let S be a non-zero subresultant of p_1 and p_2 with degree $d < d_2$ and such that $S = S^+$. If $S^>$ has degree d_2 we define $s = \mathsf{lc}(p_1)^\delta$ where $\delta = d_1 - d_2$ otherwise we define $s = \mathsf{lc}(S^>)$. The Block Structure Theorem implies the following properties. First, one can compute the subresultant S^- from $S^>$, S^+ and s. Secondly, one can compute the subresultant $S^<$ from $S^>$, S^+, S^- and s. This allows us to define a *next-subresultant* operation denoted by nsr and defined by $\mathsf{nsr}(S^>, S^+, s) = S^-$ and $\mathsf{nsr}(S^>, S^+, S^-, s) = S^<$. We will use the following facts. Computing $\mathsf{nsr}(p_1, p_2, s)$ only requires multiplications by $\mathsf{lc}(p_2)$ and exact divisions by s. The only exact divisions involved in $\mathsf{nsr}(p_1, p_2, p_3, s)$ are by $\mathsf{lc}(p_1)$, $\mathsf{lc}(p_2)$ and s. Finally $\mathsf{lsr}(p_1, p_2)$ can be computed as follows:

Algorithm 8

- **Input:** $p_1, p_2 \in \mathbf{A}[X]$ with $\mathsf{deg}(p_1) \geq \mathsf{deg}(p_2) > 0$.
- **Output:** The last-subresultant of p_1 and p_2.
- $\mathsf{lsr}(p_1, p_2) ==$
  $\delta := \mathsf{deg}(p_1) - \mathsf{deg}(p_2)$
  $s := \mathsf{lc}(p_2)^\delta$
  $(p_1,\ p_2) := (p_2,\ \mathsf{prem}(p_1, -p_2))$
  repeat
    $p_2 = 0 \Longrightarrow$ **return** p_1
    $p_3 := \mathsf{nsr}(p_1, p_2, s)$
    $\mathsf{deg}(p_3) = 0 \Longrightarrow$ **return** p_3
    $(p_1,\ p_2) := (p_3,\ \mathsf{nsr}(p_1, p_2, p_3, s))$
    $s := \mathsf{lc}(p_1)$

Let T, p_1, p_2 and k be as in Specification 3. Let $a = p_1$ and $b = p_2$. Assume that $\mathcal{H}_2(T, k+1)$ holds. We shall adapt Algorithm 8 in order to compute $\mathsf{lsr}(p_1, p_2, T)$. We proceed by induction on the rank of the process $[\{p_1, p_2\}, T]$.

A first step is modify in Algorithm 8 as follows. First, we replace the test $p_2 = 0$ by $\mathsf{red}(p_2, T) = 0$. Secondly, we insert between the first and the second lines of the **repeat** loop the following: if $\mathsf{lc}(p_2, v)$ is not regular w.r.t. T then **error**. Indeed, since $\mathcal{H}_2(T, k+1)$ holds we can check whether $\mathsf{lc}(p_2, v)$ is regular or not w.r.t. T. Assume now that this modified algorithm terminates without producing an error. Then we return either $[p_1, T]$ (if $\mathsf{red}(p_2, T) = 0$ holds) or $[p_3, T]$ (if $\mathsf{deg}(p_3, v) = 0$). Observe that these returned values satisfy property (G_1) of Definition 29. Indeed, the leading coefficients w.r.t. v of the successive subresultants have been proved to be regular w.r.t. T. For the same reason every exact division performed by the algorithm is well defined modulo $\mathbf{Sat}(T)$. Now observe that the fact that $\mathsf{lsr}(p_1, p_2)$ lies in the ideal generated by p_1 and p_2

comes from a manipulation of determinants (that only requires commutativity for the coefficient ring). Therefore, even if $\mathbf{Sat}(T)$ is not prime, our returned values satisfy property (G_2). Finally property (G_3) holds because $\mathsf{lsr}(p_1, p_2)$ pseudo-divides p_1 and p_2, provided that this pseudo-division is well defined, which is the case. It follows that we have proved Algorithm 9 in the case where at each iteration either $\mathsf{lc}(p_2, v)$ is regular w.r.t. T or p_2 is null modulo $\mathbf{Sat}(T)$.

Algorithm 9

- **Input:** see Specification 3.
- **Output:** see Specification 3.
- $\mathsf{lsr}(p_1, p_2, T)$ ==

$v := \mathsf{mvar}(p_1)$
$\delta := \mathsf{mdeg}(p_1) - \mathsf{mdeg}(p_2)$
$s := \mathsf{init}(p_2)^{\delta}$
$(p_1, p_2) := (p_2, \mathsf{prem}(p_1, -p_2))$
$R := [T]$
repeat
  $R' := [\,]$
  for $C \in R$ **repeat**
    for $D \in \mathsf{regularizeInitial}(p_2\, v, C)$ **repeat**
      $\mathsf{dim}(D) < \mathsf{dim}(C) \Longrightarrow$ **output** $[\emptyset, D]$
      $\mathsf{red}(p_2, D) = 0 \Longrightarrow$ **output** $[p_1, D]$
      $\mathsf{red}(\mathsf{lc}(p_2, v), D) \neq 0 \Longrightarrow R' := \mathsf{cons}(D, R')$
      $\mathsf{deg}(\mathsf{red}(p_2, D), v) = 0 \Longrightarrow$ **output** $[\mathsf{red}(p_2, D), D]$
      for $[G, E] \in \mathsf{lsr}(p_1, \mathsf{red}(p_2, D), D)$ **repeat output** $[G, E]$
  $R' = [\,] \Longrightarrow$ **exit**
  $p_3 := \mathsf{nsr}(p_1, p_2, s)$
  $\mathsf{deg}(p_3, v) = 0 \Longrightarrow$
    for $C \in R'$ **repeat output** $[p_3, C]$
    exit
  $(p_1, p_2) := (p_3, \mathsf{nsr}(p_1, p_2, p_3, s))$
  $s := \mathsf{init}(p_1)$
  $R := R'$

The variable R of Algorithm 9 is a list of regular chains whose initial value is $[T]$. Then the call $\mathsf{regularizeInitial}(p_2\, v, C)$ may split C and lead to new *cases*. Some of these cases are treated directly inside the second **for** loop, possibly by means of a recursive call to lsr. The others are delayed by inserting the corresponding regular chains into R. Observe that the multiplication of p_2 by $v = x_{k+1}$ in the call $\mathsf{regularizeInitial}(p_2\, v, C)$ is just a trick in order to avoid the distinction between the cases $\mathsf{deg}(p_2, v) = 0$ and $\mathsf{deg}(p_2, v) > 0$. Because of the composition property of Proposition 22 we only need to check the second **for** loop. Once this is done we will have proved Proposition 37. For every regular chain D returned by $\mathsf{regularizeInitial}(p_2\, v, C)$ five cases arise.

(1) $\mathsf{dim}(D) < \mathsf{dim}(C)$. We return $[\emptyset, D]$ as expected in Specification 3.
(2) $\mathsf{red}(p_2, D) = 0$. We return $[p_1, D]$. Indeed, since $\mathsf{dim}(D) = \mathsf{dim}(C)$, all our arguments for the Algorithm 8 with **error** apply.
(3) $\mathsf{red}(\mathsf{lc}(p_2, v), D) \neq 0$. Since $\mathsf{dim}(D) = \mathsf{dim}(C)$, we can continue the computations as if they were started w.r.t. D.
(4) $\mathsf{deg}(\mathsf{red}(p_2, D), v) = 0$. Since $\mathsf{dim}(D) = \mathsf{dim}(C)$, we could have followed the computations as if they were started w.r.t. D. Since $\mathsf{red}(p_2, D)$ is regular w.r.t. D and has degree 0 w.r.t. v, we return $[\mathsf{red}(p_2, D), D]$.
(5) $\mathsf{red}(\mathsf{lc}(p_2, v), D) = 0$ but $\mathsf{deg}(\mathsf{red}(p_2, D), v) > 0$. This implies that $\mathsf{init}(p_2)$ reduces to zero w.r.t. D. Therefore our induction hypothesis applies. Since $\mathsf{init}(p_1)$ is regular w.r.t. D, the recursive call $\mathsf{lsr}(p_1, \mathsf{red}(p_2, D), D)$ is valid. Since each regular subresultant S of the input polynomials a and b lies in the ideal they generate, each value returned by $\mathsf{lsr}(p_1, \mathsf{red}(p_2, D), D)$ is a correct result for the starting call $\mathsf{lsr}(a, b, T)$.

Proposition 37 *Let T, p_1, p_2 be as above. If $\mathcal{H}_2(T, k+1)$ holds then Algorithm 9 terminates and satisfies its specification. Moreover $\mathcal{H}_3(T, k+1)$ holds.*

Let $p \in \mathbf{P}_n$ be a polynomial and $T \subseteq \mathbf{P}_n$ be a regular chain. Let m be the smallest integer in the range $1 \cdots n$ such that $T \subseteq \mathbf{P}_m$ holds. We assume that $\mathcal{H}_1(T, p)$ and $\mathcal{H}_2(T, m)$ hold. We shall prove that $\mathsf{regular?}(p, T)$ defined in Algorithm 10 below satisfies its output specification.

Since $\mathcal{H}_1(T, p)$ holds Proposition 35 applies. Hence $\mathsf{regularizeInitial}(p, T)$ returns a regular delayed split $(C_1, \ldots, C_d)$ of T such that either $\mathsf{red}(p, C_i)$ lies in $\mathbf{k}$ or its initial is regular w.r.t. C_i. By Proposition 22 it suffices to prove that for every $C = C_1, \ldots, C_d$ the corresponding iteration of the **for** loop satisfies the output specification of $\mathsf{regular?}(p, C)$. The case where $\mathsf{red}(p, C)$ lies in $\mathbf{k}$ or its main variable variable v is not algebraic w.r.t. C is easy to check. So we assume that $v \in \mathsf{alg}(C)$ holds. Since $\mathcal{H}_2(T, m)$ holds Proposition 37 applies. In particular the call $\mathsf{lsr}(p_1, p_2, C_v^-)$ generates a regular delayed split of C_v^-.

For each of the above cases, if there is a recursive call to $\mathsf{regular?}$ we show that it satisfies its output specification (i.e. using hypothesis $\mathcal{H}_1(T, p)$) otherwise we show that the desired result is obtained directly without any recursive call.

(1) $G = \emptyset$. Since $\mathcal{H}_2(T, m)$ holds, $\mathsf{extend}(D, ht \cup C_v^+)$ returns a regular delayed split of $D \cup ht \cup C_v^+$, by Proposition 28. Each regular chain E of this split has no higher rank than $D \cup ht \cup C_v^+$. Moreover we have either $D \prec C_v^-$ or $D \sim C_v^-$. But the latter would imply $\mathsf{dim}(D) = \mathsf{dim}(C_v^-)$, a contradiction with $G = \emptyset$. Since $C = C_v^- \cup t \cup C_v^+$, we obtain $E \prec C$. With hypothesis $\mathcal{H}_1(T, p)$, since C has no higher rank than T, it follows that the recursive call $\mathsf{regular?}(q, E)$ is valid. Note that in order to apply Theorem 23 the multiplication by h in $\mathsf{extend}(D, ht \cup C_v^+)$ is needed.
(2) $(g \in \mathbf{k})$ **or** $(\mathsf{mvar}(g) < v)$. Recall that we have $\mathbf{Sat}(D) \subseteq \mathbf{Sat}(C_v^-)$. Since

$\mathsf{dim}(D) = \mathsf{dim}(\ C_v^-)$ holds, $D \cup t \cup\ C_v^+$ is a regular chain. Since p equals $\mathsf{red}(p, C)$ modulo $\mathbf{Sat}(C)$, this equality holds also modulo $\mathbf{Sat}(D \cup t)$. Then, by condition (G_2) of Definition 29, there exist polynomials $u, u \in \mathbf{P}_{k+1}$ such that $up\ = vt + g$ holds modulo $\mathbf{Sat}(D)$. Now, by condition (G_1), the polynomial g is regular w.r.t. D and thus w.r.t. $D \cup t$. It follows that p is regular w.r.t. $D \cup t \cup\ C_v^+$ and that this regular chain must be returned. Observe that h is regular w.r.t the output regular chain.

(3) $\mathsf{mvar}(g) = x_{k+1}$ and $\mathsf{mdeg}(g) = \mathsf{mdeg}(t)$. By Remark 30 we have $p \in \mathbf{Sat}(D \cup t)$. Thus $D \cup t \cup\ C_v^+$ must be returned by the algorithm.

(4) $\mathsf{mvar}(g) = x_{k+1}$ and $\mathsf{mdeg}(g) \neq \mathsf{mdeg}(t)$. We have $\mathsf{mdeg}(g) < \mathsf{mdeg}(t)$, by Remark 30. Hence we apply Theorem 7 and its Corollary 8. We use the operation intersect specified in Remark 20 and consequently the operation decompose from Specification 1. By Theorem 7 the sets $D \cup g \cup\ C_v^+$ and $D \cup\ f\ \cup\ C_v^+$ are regular chains. Moreover, the polynomial h is regular w.r.t. them. By condition (G_3), the polynomial p reduces to zero w.r.t. $D \cup g \cup\ C_v^+$ and this set must be returned by the algorithm. Since $D\ \cup\ f\ \cup\ C_v^+ \prec C$. holds, the recursive call $\mathsf{regular?}(q, D \cup f \cup\ C_v^+)$ is valid. We put $h_g = \mathsf{init}(g)$. Since $\mathcal{H}_2(T, k+1)$ holds, Proposition 36 applies. Hence $\mathsf{intersect}(h_g, D \cup t \cup\ C_v^+)$ returns a delayed split of $\mathbf{Z}(h_g, D \cup t \cup\ C_v^+)$ consisting of regular chains. Let E be one of them. By Proposition 16, since h_g is regular w.r.t. $D \cup t \cup\ C_v^+$, we have $\mathsf{dim}(E) < \mathsf{dim}(C)$. Hence $E \prec C$ holds and the last recursive call is valid too.

Algorithm 10

- **Input:** T regular chain of $\mathbf{P}_n$ and $p \in \mathbf{P}_n$.
- **Output:** a regular delayed split $(T_1, \ldots T_d)$ of T such that if $\mathsf{red}(p, T_i) \neq 0$ then p is regular w.r.t. T_i, for every $i = 1 \cdots d$.
- $\mathsf{regular?}(p, T)$ ==

```
for C ∈ regularizeInitial(p, T) repeat
  q := red(p, C)
  q ∈ k ⟹ output C
  v := mvar(q)
  v ∉ alg(C) ⟹ output C
  t := C_v
  if mdeg(t) ≤ mdeg(q) then (p1, p2) := (q, t) else (p1, p2) := (t, q)
  Let h be the product of the initials of C_v^-
  for [G, D] ∈ lsr(p1, p2, C_v^-) repeat
    G = ∅ ⟹
      for E ∈ extend(D, ht ∪ C_v^+) repeat
        for H ∈ regular?(q, E) repeat output H
    {g} := G
    (g ∈ k) or (mvar(g) < v) ⟹ output D ∪ t ∪ C_v^+
    mdeg(g) = mdeg(t) ⟹ output D ∪ t ∪ C_v^+
    f := pquo(t, g)
```

output $D \cup g \cup\ C_v^+$
for $E \in$ regular?$(q, D \cup f \cup\ C_v^+)$ **repeat output** E
for $E \in$ intersect(init$(g), D \cup t \cup\ C_v^+)$ **repeat**
for $H \in$ regular?(q, E) **repeat output** H
exit

Theorem 38 *Algorithm 10 terminates and satisfies its specification.*

PROOF. Following the above discussion, the only remaining point to prove is that Algorithm 10 terminates. Let us consider the ordered tree $\mathcal{T}$ of the recursive calls to regular? (possibly via the other operations) required to compute regular?(p, T). We know if $\mathcal{H}_1(T, p)$ and $\mathcal{H}_2(T, m)$ hold then regular?(p, T) can be computed. Hence for each recursive call regular?(p', T') we have either $[p', T'] \prec [p, T]$ or the greatest variable occurring in T' is strictly smaller than that of T. Hence $\mathcal{T}$ is finite. Finally it is easy to check than when no call to regular? is needed, each of our algorithms satisfies its specification. □

Corollary 39 *Algorithms 2, 3, 4, 6, 5, 7, and 9 terminate and satisfy their specifications.*

6 Implementation and Improvements

We give several improvements for the algorithms of the previous sections. Then we show how to use them for producing decompositions into normalized and square-free regular chains, and for removing redundant quasi-components. In particular we explain how to compute the decompositions of Lazard's algorithm Lazard (1991). Finally we adapt our algorithms in order to compute efficiently the decompositions of Kalkbrener (1991).

Clearly, the operation regular? is called intensively by our algorithms. Let $p \in \mathbf{P}_n$ be a polynomial and $T \subseteq \mathbf{P}_n$ be a regular chain. Here are some tricks in order to speed up the computation of regular?(p, T) and to reduce the number of recursive calls. First, if p is normalized w.r.t. T or if $\mathbf{Sat}(T)$ is a prime ideal then regular?(p, T) is simply (T). Consider now the call to intersect in our Algorithm 10. Let h_g be the initial of g. Assume that T is purely algebraic. Assume also that every variable occurring in h_g is algebraic w.r.t. $D \cup t \cup\ C_v^+$. Then $\mathbf{Z}(h_g, D \cup t \cup\ C_v^+)$ is necessarily empty, by virtue of Proposition 16. This trick applies very often in practice and provides a great speed up. Consider now the case where the previous trick does not work. Let h_t and h_q be the initials of t and q. Let γ be the gcd of h_t and h_q as multivariate polynomials of $\mathbf{P}_k$. Assume that g is the subresultant-gcd of t

and q regarded as univariate polynomials in x_{k+1}. Then h_g divides γg and again $\mathbf{Z}(h_g, D \cup t \cup C_v^+)$ is necessarily empty. This is a very powerful trick.

We give now some tricks in order to improve the operation lsr. Let p_1, p_2, and T be as Algorithm 9. If T is purely algebraic and if the coefficients of p_1 and p_2 (regarded as univariate polynomials) are algebraic w.r.t. T, then we can use the algorithm given in (Moreno Maza and Rioboo, 1995). Furthermore, if $\mathbf{Sat}(T)$ is prime, we can use a modular algorithm for computing polynomial gcds (Hoeij and Monagan, 2002, 2004; Boulier et al., 2004).

Several of our algorithms are simpler if their input polynomials or regular chains are normalized. Let $T \subseteq \mathbf{P}_n$ be a regular chain. If the polynomial $p \in \mathbf{P}_n$ is normalized w.r.t. T then $\mathsf{regularizeInitial}(p, T)$ returns simply (T). In the same way, if $p \in \mathbf{P}_n$ is normalized w.r.t. T (and $\mathsf{mvar}(p)$ is greater than any variable of T) then $\mathsf{extend}(p, T)$ returns simply $(T \cup p)$.

We explain now how we compute a delayed split of T by means of normalized regular chains. We proceed by induction on the rank of the greatest variable occurring in T. Assume that T writes $N \cup t$ where N is a normalized regular chain and t is a polynomial regular w.r.t. N with main variable greater than any variable of N. By repeated calls to the operation lsr, computing the Bézout coefficients given by property (G_2) of Definition 29, we can compute a sequence of processes $([q_1, M_1], \ldots, [q_e, M_e])$ with the following properties. First $(M_1 \cup t, \ldots, M_e \cup t)$ is a regular delayed split of $N \cup t$ where each M_i is normalized. Secondly, each polynomial $\mathsf{red}(q_i\, t, M_i)$ is not constant, normalized w.r.t. M_i and has the same rank as t. Clearly every set $M_i \cup \mathsf{red}(q_i\, t, M_i)$ is a normalized regular chain and is part of the desired delayed split of $N \cup t$. In order to apply formula (c) of Lemma 4 we need to compute $\mathsf{intersect}(q_i, M_i)$ for every $i = 1 \cdots e$, obtaining regular chains $M_{i,j}$. Then we call $\mathsf{extend}(M_{i,j}, t)$ since t may not be regular w.r.t. $M_{i,j}$ and obtain regular chains $M_{i,j,l}$ to which we apply recursively our normalization procedure. The whole normalization procedure terminates since the call $\mathsf{intersect}(q_i, M_i)$ returns regular chains $M_{i,j}$ such that $\mathsf{dim}(M_{i,j}) < \mathsf{dim}(M_i)$ holds. Indeed q_i is regular w.r.t. M_i.

Several of our algorithms are also simpler if the input regular chain T is square-free. In this case, $\mathbf{Sat}(T)$ is radical. See Proposition V.1 in Moreno Maza (1997). This has several advantages. First, this is crucial for deciding whether a quasi-component is contained in another as we shall below. Secondly, this clearly reduces the size of the computed objects and increases the legibility of the output. Lastly, this allows an improvement of Algorithm 10 as follows. Assume that we are considering the case $\mathsf{mvar}(g) = \mathsf{mvar}(t)$ and $\mathsf{mdeg}(g) < \mathsf{mdeg}(t)$ in Algorithm 10. If t is known to be square-free w.r.t. D, then the polynomials g and f are relatively prime modulo $\mathbf{Sat}(D)$. Recall that p reduces to zero modulo $D \cup g$. Then p must be regular w.r.t. $D \cup f$. Hence we can avoid the second recursive call in Algorithm 10 and return directly $D \cup f \cup C_v^+$.

The production of square-free regular chains is similar to that of normalized triangular sets and we do not detail this point.

Let T_1 and T_2 be two regular chains of $\mathbf{P}_n$. We want to decide whether $\mathbf{W}(T_1) \subseteq \mathbf{W}(T_2)$ holds. If T_1 and T_2 are square-free normalized triangular sets, one can use the procedure subQuasiComponent? of Moreno Maza (1997). We discuss here the general case, for which we provide several criteria. Indeed we do not know a decision algorithm in the general case. However these criteria are powerful enough in practice, especially if T_1 and T_2 are square-free. Let h_2 be the product of the initials of T_2. If $h_2 \in \mathbf{k}$ then $\mathbf{W}(T_2) = \mathbf{V}(T_2)$. Thus $\mathbf{W}(T_1) \subseteq \mathbf{W}(T_2)$ and $\overline{\mathbf{W}(T_1)} \subseteq \overline{\mathbf{W}(T_2)}$ are equivalent. Thus if $T_2 \subseteq \mathbf{Sat}(T_1)$ holds, we can answer *yes* otherwise we cannot conclude. Clearly, the chain T_1 should better be square-free. If $h_2 \notin \mathbf{k}$ then we need to add another condition. First we can test $\mathsf{intersect}(h_2, T_1) = \emptyset$ (together with $T_2 \subseteq \mathbf{Sat}(T_1)$) and answer *yes* if this holds. Secondly we can refine this idea by computing $\mathsf{regular?}(h_2, T_1) = (R_1, \ldots, R_d)$. Then, for every R_i such that $\mathsf{red}(h_2, R_i) \neq 0$ and such that $\mathsf{intersect}(h_2, R_i) \neq \emptyset$ together with $T_2 \subseteq \mathbf{Sat}(R_i)$ we answer *yes*.

A direct application of our procedures to decide whether a quasi-component is contained in another is the management of the processes. This task has to be realized by Algorithm 2. We use its notations. Recall that the idea is to produce the regular chains in the output of $\mathsf{solve}(F)$ in order of decreasing dimension such that the redundant quasi-components can be removed as soon as possible. Hence, among all the processes in R one must chose a process $[F_1, U_1]$ such that $\mathsf{dim}(U_1)$ is maximal. Then we must refine this choice by avoiding the process $[F_1, U_1]$ such that F_1 contains polynomials which have been proved to be regular w.r.t. U_1. When $\mathsf{decompose}(p, U_1)$ has returned a delayed split of $\mathbf{Z}(p, U_1)$ we must perform several cleanings. First, for a given process $[F_1, U_1]$, we can remove from F_1 any polynomial f such that $\mathsf{red}(f, U_1) = 0$ holds. Indeed any regular chain C generated by this process satisfies $\mathbf{Sat}(U_1) \subseteq \mathbf{Sat}(C)$ and thus $\mathsf{red}(f, C) = 0$. Secondly, we can remove from R every process $[F_1, U_1]$ such that there exists a computed regular chain R (part of the output) satisfying $\mathbf{Z}(F_1, U_1) \subseteq \mathbf{W}(R)$. Lastly, we can compare the processes of R between them. All these cleanings leave the loop invariants of the proof of Proposition 19 unchanged.

It follows that we have provided all the tools in order to compute a Lazard split of $\mathbf{V}(F)$ by means of Lazard's sets (i.e. square-free normalized triangular sets) such no output quasi-component is contained in another one.

If Algorithm 2 is used for computing a Kalkbrener split of a given algebraic variety $\mathbf{V}(F)$, redundant components may appear. More precisely this split may contain two regular chains T and T' such that $\overline{\mathbf{W}(T)} \subseteq \overline{\mathbf{W}(T')}$ holds, even if $\mathbf{W}(T) \subseteq \mathbf{W}(T')$ does not hold. This is essentially due to Theorem 7. Here are some tricks to reduce these redundancies in a significant manner.

Since the height of the saturated ideal of a regular chain equals the number of its elements, we can delete any process $[F_1, T_1]$ of Algorithm 2 such that T_1 contains more elements than the input system F. In fact we can do more as follows. Let T be a regular chain containing as many elements as F. Let p be in $\mathbf{P}_n$ and $[F_i, T_i]$ be a process returned by $\mathsf{decompose}(p, T)$ such that $F_i \neq \emptyset$ holds. Then we can discard $[F_i, T_i]$ from the output of $\mathsf{decompose}(p, T)$. Indeed there exists $f_i \in F_i$ such that f_i is regular w.r.t. T_i. Observe also that during the computation of $\mathsf{decompose}(p, T)$ we should never perform the case $G = \emptyset$ in Algorithm 5 and the case $v \notin \mathsf{alg}(C)$ in Algorithm 7, since they will generate regular chains D such that $\mathsf{dim}(D) < \mathsf{dim}(T)$. See Table 6 in Aubry and Moreno Maza (1999) for experimental results.

References

Aubry, P., 1999. Ensembles triangulaires de polynômes et résolution des systèmes d'équations algébriques. Ph.D. thesis, Université Paris 6.

Aubry, P., Lazard, D., Moreno Maza, M., 1999. On the theories of triangular sets. J. Symb. Comp. 28 (1-2), 105–124.

Aubry, P., Moreno Maza, M., 1999. Triangular sets for solving polynomial systems: A comparative implementation of four methods. J. Symb. Comp. 28 (1-2), 125–154.

Boulier, F., Denis-Vidal, L., Henin, T., Lemaire, F., 2004. Lépisme. In: International Conference on Poynomial System Solving. University of Paris 6, France.

Boulier, F., Lazard, D., Ollivier, F., Petitot, M., 1995. Representation for the radical of a finitely generated differential ideal. In: proceedings of ISSAC'95. Montréal, Canada, pp. 158–166.

Boulier, F., Lemaire, F., 2000. Computing canonical representatives of regular differential ideals. In: proceedings of ISSAC 2000. ACM Press, St Andrews, Scotland, pp. 37–46.

Boulier, F., Lemaire, F., Moreno Maza, M., 2001. Well known theorems on triangular systems. Tech. Rep. LIFL 2001–09, Université Lille I, LIFL.

Boulier, F., Moreno Maza, M., C.Oancea, 2004. A new henselian construction and its application to polynomial gcds over direct products of fields. In: proceedings of EACA'04. Universidad de Santander, Spain.

Broadbery, P. A., Dooley, S. S., Iglio, P., Morisson, S. C., Steinbach, J. M., Sutor, R. S., Watt, S. M., November 1994. AXIOM Library Compiler User Guide. NAG, The Numerical Algorithms Group Limited, Oxford, United Kingdom, 1st Edition, AXIOM is a registred trade mark of NAG.

Dahan, X., Moreno Maza, M., Schost, É., Wu, W., Xie, Y., 2004. Equiprojectable decompositions of zero-dimensional varieties. In: Valibouze, A. (Ed.), International Conference on Poynomial System Solving. University of Paris 6, France.

Dahan, X., Moreno Maza, M., Schost, É., Wu, W., Xie, Y., 2005. Lifting techniques for triangular decompositions. In: ISSAC'05. ACM Press, pp. 108–115.

Dahan, X., Schost, É., 2004. Sharp estimates for triangular sets. In: ISSAC 04. ACM, pp. 103–110.

Dellière, S., 1999. Triangularisation de systèmes constructibles. Application à l'évaluation dynamique. Ph.D. thesis, Université de Limoges.

Ducos, L., 2000. Optimizations of the subresultant algorithm. Journal of Pure and Applied Algebra 145, 149–163.

Emiris, I., Mourrain, B., 1999. Matrices in elimination theory. J. Symb. Comp. 28 (1-2), 3–44.

Foursov, M., Moreno Maza, M., 2002. On computer-assisted classification of coupled integrable equations. J. Symb. Comp. 33, 647–660.

Gallo, G., Mishra, B., 1990. Efficient algorithms and bounds for Wu-Ritt characteristic sets. In: Proc. MEGA'90. pp. 119–142.

Giusti, M., Heintz, J., Morais, J. E., Pardo, L. M., 1995. When polynomial equation systems can be solved fast? In: AAECC-11. Vol. 948 of LNCS. Springer, pp. 205–231.

Gómez Díaz, T., 1994. Quelques applications de l'évaluation dynamique. Ph.D. thesis, Université de Limoges.

Hoeij, M. v., Monagan, M., Jul. 2002. A modular gcd algorithm over number fields presented with multiple extensions. In: Mora, T. (Ed.), Proc. ISSAC 2002. ACM Press, pp. 109–116.

Hoeij, M. v., Monagan, M., 2004. Algorithm for polynomial gcd computation over algebraic function fields, (accepted for ISSAC 2004).

Hubert, É., 2003. Notes on triangular sets and triangulation-decomposition algorithms. II. Differential systems. In: Symbolic and numerical scientific computation (Hagenberg, 2001). Vol. 2630 of LNCS. Springer.

Jenks, R. D., Sutor, R. S., 1992. AXIOM, The Scientific Computation System. Springer-Verlag, AXIOM is a trade mark of NAG Ltd, Oxford UK.

Kalkbrener, M., 1991. Three contributions to elimination theory. Ph.D. thesis, Johannes Kepler University, Linz.

Kalkbrener, M., 1993. A generalized euclidean algorithm for computing triangular representations of algebraic varieties. J. Symb. Comp. 15, 143–167.

Kalkbrener, M., 1998. Algorithmic properties of polynomial rings. J. Symb. Comp. 26 (5), 525–581.

Kogan, I. A., Moreno Maza, M., Jul. 2002. Computation of canonical forms for ternary cubics. In: Mora, T. (Ed.), Proc. ISSAC 2002. ACM Press, pp. 151–160.

Lazard, D., 1991. A new method for solving algebraic systems of positive dimension. Discr. App. Math 33, 147–160.

Lecerf, G., 2000. Computing an equidimensional decomposition of an algebraic variety by means of geometric resolutions. In: Proceedings of ISSAC'2000 (ACM).

Lecerf, G., 2002. Quadratic Newton iteration for systems with multiplicity. Foundations of Computational Mathematics 2 (3), 247–293.

Lecerf, G., 2003. Computing the equidimensional decomposition of an algebraic closed set by means of lifting fibers. Journal of Complexity 19 (4), 564–596.

Lemaire, F., Moreno Maza, M., Xie, Y., 2005. The `RegularChains` library. In: Kotsireas, I. S. (Ed.), *Maple Conference 2005*. pp. 355–368.

Li, Z., 1995. An implementation of the characteristic set method for solving algebraic equations. In: Proc. POSSO Workshop. pp. 107–122.

Loos, R., 1982. Generalized polynomial remainder sequences. In: Symbolic and Algebraic Computation. Spinger-Verlag, pp. 115–137.

Moreno Maza, M., 1997. Calculs de Pgcd au-dessus des Tours d'Extensions Simples et Résolution des Systèmes d'Équations Algébriques. Ph.D. thesis, Université Paris 6.

Moreno Maza, M., 1999. On triangular decompositions of algebraic varieties. Tech. Rep. TR 4/99, NAG Ltd, Oxford, UK, presented at the MEGA-2000 Conference, Bath, England.

Moreno Maza, M., Rioboo, R., 1995. Polynomial gcd computations over towers of algebraic extensions. In: Proc. AAECC-11. Springer, pp. 365–382.

Ollivier, F., 1990. Le problème de l'identifiabilité structurelle globale : approche théorique, méthodes effectives et bornes de complexité. Ph.D. thesis, École Polytechnique.

Pardo, L. M., 1995. How lower and upper complexity bounds meet in elimination theory. In: Proceedings of AAECC 11. Vol. 948 of Lecture Notes in Computer Science. Springer-Verlag, pp. 33–69.

Ritt, J. F., 1932. Differential Equations from an Algebraic Standpoint. Vol. 14. American Mathematical Society, New York.

Ritt, J. F., 1950. Differential Algebra. Amer. Math. Soc, New York.

Rosenfeld, A., 1959. Specializations in differential algebra. Trans. Amer. Math. Soc. 90, 394–407.

Samuel, P., Zariski, O., 1967. Commutative algebra. D. Van Nostrand Company, INC.

Schost, É., 2003. Complexity results for triangular sets. J. Symb. Comp. 36 (3-4), 555–594.

Seidenberg, A., 1956. Some remarks on Hilbert's Nullstellensatz. Archiv der Mathematik 7, 235–240.

Szántó, Á., 1999. Computation with polynomial systems. Ph.D. thesis, Cornell University.

van der Waerden, B., 1991. Algebra. Springer-Verlag, seventh edition.

Wang, D. M., 1993. An elimination method for polynomial systems. J. Symb. Comp. 16, 83–114.

Wang, D. M., 1998. Decomposing polynomial systems into simple systems. J. Symb. Comp. 25 (3), 295–314.

Wu, W. T., 1986. On zeros of algebraic equations – an application of Ritt principle. Kexue Tongbao 31 (1), 1–5.