14

Quantifier Elimination

The principal problem we consider in this chapter is the quantifier elimination problem. This problem was already studied in Chapter 11, where we obtained doubly exponential complexity in the number of variables. On the other hand, we have seen in Chapter 13 an algorithm for the existential theory of the reals (which is to decide the truth or the falsity of a sentence with a single block of existential quantifiers) with complexity singly exponential in the number of variables (see Theorem 13.13). In this chapter, we pay special attention to the structure of the blocks of variables in a formula in order to take into account this block structure in the complexity estimates and improve the results obtained in Chapter 11.

If $Z = (Z_1, ..., Z_\ell)$, Φ is a formula, and $\mathrm{Qu} \in \{\forall, \exists\}$, we denote the formula $(\mathrm{Qu}\ Z_1)...(\mathrm{Qu}\ Z_\ell)\ \Phi$ by the abbreviation $(\mathrm{Qu}\ Z)\ \Phi$.

Let $\mathcal{P} \subset \mathrm{R}[X_1, ..., X_k, Y_1, ..., Y_\ell]$ be finite, and let Π denote a partition of the list of variables $X = (X_1, ..., X_k)$ into blocks, $X_{[1]}, ..., X_{[\omega]}$, where the block $X_{[i]}$ is of size $k_i, 1 \leq i \leq \omega$, $\sum_{1 \leq i \leq \omega} k_i = k$.

A $(\mathcal{P}, \Pi)$-formula $\Phi(Y)$ is a formula of the form

$$\Phi(Y) = (\mathrm{Qu}_1 X_{[1]})...(\mathrm{Qu}_\omega X_{[\omega]}) F(X, Y),$$

where $\mathrm{Qu}_i \in \{\forall, \exists\}$, $Y = (Y_1, ..., Y_\ell)$, and $F(X, Y)$ is a quantifier free $\mathcal{P}$-formula.

In Section 14.1, we describe an algorithm for solving the general decision problem, that is a procedure to decide the truth or falsity of a $(\mathcal{P}, \Pi)$-sentence. The key notion here is the tree of realizable sign conditions of a family of polynomials with respect to a block structure Π on the set of variables. This is a generalization of the set of realizable sign conditions, seen in Chapter 7, which corresponds to one single block of variables. It is also a generalization of the tree of cylindrical realizable sign conditions, seen in Chapter 11, which correspond to k blocks of one variable each. The basic idea of this algorithm is to perform parametrically the algorithm in Chapter 13, using the critical point method.

Section 14.2 is devoted to the more general problem of quantifier elimination for a $(\mathcal{P}, \Pi)$-formula.

Section 14.3 is devoted to a variant of Quantifier Elimination exploiting better the logical structure of the formula.

Finally, the block elimination technique is used to perform global optimization and compute the dimension of a semi-algebraic set in Section 14.4 and Section 14.5.

14.1 Algorithm for the General Decision Problem

We first study the general decision problem, which is to decide the truth or falsity of a $(\mathcal{P}, \Pi)$-sentence (which is a $(\mathcal{P}, \Pi)$-formula without free variables). In order to decide the truth or falsity of a sentence, we construct a certain tree of sign conditions adapted to the block structure Π of the sentence, which we define below.

The following definition generalizes the definition of the tree of cylindrical realizable sign conditions (Notation 11.7). The importance of this notion is that the truth or falsity of any $(\mathcal{P}, \Pi)$-sentence can be decided from $\mathrm{SIGN}_{\Pi}(\mathcal{P})$.

Notation 14.1. [Block realizable sign conditions] Let $\mathcal{P}$ be a set of s polynomials in k variables $X_1, ..., X_k$, and let Π denote a partition of the list of variables $X_1, ..., X_k$ into blocks, $X_{[1]}, ..., X_{[\omega]}$, where the block $X_{[i]}$ is of size k_i, for $1 \leq i \leq \omega$, $\sum_{1 \leq i \leq \omega} k_i = k$. Let $\mathrm{R}^{[i]} = \mathrm{R}^{k_1 + \cdots + k_i}$, and let $\pi_{[i]}$ be the projection from $\mathrm{R}^{[i+1]}$ to $\mathrm{R}^{[i]}$ forgetting the last k_{i+1}-coordinates. Note that $\mathrm{R}^{[\omega]} = \mathrm{R}^k$. By convention, $\mathrm{R}^{[0]} = \{0\}$.

We are going to define inductively the tree of realizable sign conditions of $\mathcal{P}$ with respect to Π.

For $z \in \mathrm{R}^{[\omega]}$, let $\mathrm{SIGN}_{\Pi,\omega}(\mathcal{P})(z) = \mathrm{sign}(\mathcal{P})(z)$, where $\mathrm{sign}(\mathcal{P})(z)$ is the sign condition on $\mathcal{P}$ mapping $P \in \mathcal{P}$ to $\mathrm{sign}(P)(z) \in \{0, 1, -1\}$ (Notation 11.7).

For all i, $0 \leq i < \omega$, and $y \in \mathrm{R}^{[i]}$, we inductively define,

$$\mathrm{SIGN}_{\Pi,i}(\mathcal{P})(y) = \{\mathrm{SIGN}_{\Pi,i+1}(\mathcal{P})(z) | z \in \mathrm{R}^{[i+1]}, \pi_{[i]}(z) = y\}.$$

Finally, we define

$$\mathrm{SIGN}_{\Pi}(\mathcal{P}) = \mathrm{SIGN}_{\Pi,0}(\mathcal{P})(0).$$

Note that $\mathrm{SIGN}_{\Pi}(\mathcal{P})$ is naturally equipped with a tree structure. We call $\mathrm{SIGN}_{\Pi}(\mathcal{P})$ the **tree of realizable sign conditions of $\mathcal{P}$ with respect to Π**. □

When there is only one block of variables, we recover $\mathrm{SIGN}(\mathcal{P})$ (Notation 7.29). When $\Pi = \{X_1\}, ..., \{X_k\}$, we recover $\mathrm{CSIGN}(\mathcal{P})$ (Notation 11.7).

We will see that the truth or falsity of a $(\mathcal{P}, \Pi)$-sentence can be decided from the set $\mathrm{SIGN}_{\Pi}(\mathcal{P})$. We first consider an example.

Example 14.2. Let $P = X_1^2 + X_2^2 + X_3^2 - 1$, $\mathcal{P} = \{P\}$. Let Π consist of two blocks of variables, defined by $X_{[1]} = X_1$ and $X_{[2]} = \{X_2, X_3\}$. Note that $\pi_{[1]}$ projects $\mathrm{R}^{[2]} = \mathrm{R}^3$ to $\mathrm{R}^{[1]} = \mathrm{R}$ by forgetting the last two coordinates. We now determine $\mathrm{SIGN}_\Pi(\mathcal{P})$.

For $x \in \mathrm{R} = \mathrm{R}^{[1]}$,

$$\mathrm{SIGN}_{\Pi,1}(\mathcal{P})(x) = \{\mathrm{sign}(P(z)) \mid z \in \mathrm{R}^{[2]}, \pi_{[1]}(z) = x\}.$$

Thus

$$\mathrm{SIGN}_{\Pi,1}(\mathcal{P})(x) = \begin{cases} \{0, 1, -1\} & \text{if } x \in (-1, 1) \\ \{0, 1\} & \text{if } x \in \{-1, 1\} \\ \{1\} & \text{otherwise.} \end{cases}$$

Finally,

$$\mathrm{SIGN}_\Pi(\mathcal{P}) = \{\mathrm{SIGN}_{\Pi,1}(\mathcal{P})(x) \mid x \in \mathrm{R}\}.$$

Thus

$$\mathrm{SIGN}_\Pi(\mathcal{P}) = \{\{1\}, \{0, 1\}, \{0, 1, -1\}\}.$$

This means that there are three cases:

- there are values of x_1 for which the only sign taken by $P(x_1, x_2, x_3)$ when (x_2, x_3) varies in R^2 is 1,
- there are values of x_1 for which the only sign taken by $P(x_1, x_2, x_3)$ when (x_2, x_3) varies in R^2 are 0 and 1,
- there are values of x_1 for which the signs taken by $P(x_1, x_2, x_3)$ when (x_2, x_3) varies in R^2 are 0, 1 and -1,
- and these exhaust all choices of $x_1 \in \mathrm{R}$.

So, the sentence $(\forall X_1)\ (\exists (X_2, X_3))\ X_1^2 + X_2^2 + X_3^2 - 1 > 0$ is certainly true.

Since there are values of x_1 for which the only sign taken by $P(x_1, x_2, x_3)$ for every $(x_2, x_3) \in \mathrm{R}^2$ is 1 it is equally clear that the sentence $(\exists X_1)\ (\forall (X_2, X_3))\ X_1^2 + X_2^2 + X_3^2 - 1 > 0$ is true.

On the other hand, the sentence $(\forall X_1)\ (\exists (X_2, X_3))\ X_1^2 + X_2^2 + X_3^2 - 1 = 0$ is false: there are values of x_1 for which the only sign taken by $P(x_1, x_2, x_3)$ is 1.

This differs from what was done in Example 11.10 in that here we do not decompose the (X_2, X_3) space: this is because the variables $\{X_2, X_3\}$ belong to the same block of quantifiers. So the information provided by $\mathrm{SIGN}_\Pi(\mathcal{P})$ is weaker than the information provided by $\mathrm{CSIGN}(\mathcal{P})$ (Notation 11.7). Note that $\mathrm{SIGN}_\Pi(\mathcal{P})$ does not provide the information necessary to decide the truth or falsity of the sentence

$$\Phi = (\exists X_1)\ (\forall X_2)\ (\exists X_3)\ X_1^2 + X_2^2 + X_3^2 - 1 = 0$$

since we do not have information for the corresponding block structure, while we have able to decide that Φ is false using

$\mathrm{CSIGN}(\mathcal{P}) = \{\{\{\{1\}, \{0, 1\}, \{0, 1, -1\}\}, \{\{1\}, \{0, 1\}\}, \{\{1\}\}\}$ in Example 11.16.

If we take $\mathcal{Q} = \{X_1 - X_3^2\}$, it is easy to check that

$$\mathrm{SIGN}_\Pi(\mathcal{Q}) = \{\{1\}, \{0,1\}, \{0,1,-1\}\} = \mathrm{SIGN}_\Pi(\mathcal{P}).$$

On the other hand we can determine

$$\mathrm{CSIGN}(\mathcal{Q}) = \{\{\{1\}\}, \{\{0,1\}\}, \{\{\{0,1,-1\}\}\}$$

and notice that

$$\mathrm{CSIGN}(\mathcal{Q}) \neq \mathrm{CSIGN}(\mathcal{P}).$$

Using $\mathrm{CSIGN}(\mathcal{Q})$, we can check that the sentence

$$\Phi' = (\exists X_1)\ (\forall X_2)\ (\exists X_3)\ X_1 - X_3^2 = 0$$

while the corresponding Φ, discussed above, is false. □

We use again Notation 11.12.

Proposition 14.3. *The* $(\mathcal{P}, \Pi)$*-sentence*

$$(\mathrm{Qu}_1 X_{[1]})\ (\mathrm{Qu}_2 X_{[2]})\ ...(\mathrm{Qu}_\omega X_{[\omega]})\ F(X),$$

is true if and only if

$$\mathrm{Qu}_1\sigma_1 \in \mathrm{SIGN}_\Pi(\mathcal{P}) \quad \mathrm{Qu}_2\sigma_2 \in \sigma_1 ... \mathrm{Qu}_\omega\sigma_\omega \in \sigma_{\omega-1} \quad F^\star(\sigma_\omega).$$

Proof: The proof is by induction on the number ω of blocks of quantifiers, starting from the one outside.

Since $(\forall\, X)\ \Phi$ is equivalent to $\neg\ (\exists X)\ \neg\Phi$, we can suppose without loss of generality that Qu_1 is $\exists$.

The claim is certainly true when there is one block of existential quantifiers, by definition of $\mathrm{sign}(\mathcal{P})$.

Suppose that

$$(\exists X_{[1]})\ (\mathrm{Qu}_2 X_{[2]}) ...\ (\mathrm{Qu}_\omega X_{[\omega]}) F(X)$$

is true, and choose $a_{[1]} \in \mathrm{R}^{k_1}$ such that

$$(\mathrm{Qu}_2 X_{[2]}) ...\ (\mathrm{Qu}_\omega X_{[\omega]}) F(a_{[1]}, X_{[2]}, ..., X_{[\omega]})$$

is true. Note that if $\mathcal{P}_{a_{[1]}}$ is the set of polynomials obtained by substituting $a_{[1]} \in \mathrm{R}^{k_1}$ for $X_{[1]}$ in $\mathcal{P}$ and $\Pi' = X_{[2]}, ..., X_{[\omega]}$,

$$\mathrm{SIGN}_{\Pi,1}(\mathcal{P})(a_{[1]}) = \mathrm{SIGN}_{\Pi'}(\mathcal{P}_{a_{[1]}}).$$

By induction hypothesis,

$$\mathrm{Qu}_2\sigma_2 \in \mathrm{SIGN}_{\Pi'}(\mathcal{P}_{a_{[1]}}) ... \mathrm{Qu}_\omega\sigma_\omega \in \sigma_{\omega-1} \quad F^\star(\sigma_\omega)$$

is true. So taking $\sigma_1 = \mathrm{SIGN}_{\Pi,1}(\mathcal{P})(a_{[1]}) = \mathrm{SIGN}_{\Pi'}(\mathcal{P}_{a_{[1]}}) \in \mathrm{SIGN}_{\Pi}(\mathcal{P})$,

$$\exists \sigma_1 \in \mathrm{SIGN}_{\Pi}(\mathcal{P}) \quad \mathrm{Qu}_2\sigma_2 \in \sigma_1 ... \mathrm{Qu}_\omega\sigma_\omega \in \sigma_{\omega-1} \quad F^\star(\sigma_\omega)$$

is true.

Conversely, suppose

$$\exists \sigma_1 \in \mathrm{SIGN}_{\Pi}(\mathcal{P}) \quad \mathrm{Qu}_2\sigma_2 \in \sigma_1 ... \mathrm{Qu}_\omega\sigma_\omega \in \sigma_{\omega-1} \quad F^\star(\sigma_\omega)$$

is true and choose $\sigma_1 \in \mathrm{SIGN}_{\Pi}(\mathcal{P})$ such that

$$\mathrm{Qu}_2\sigma_2 \in \sigma_1 ... \mathrm{Qu}_\omega\sigma_\omega \in \sigma_{\omega-1} \quad F^\star(\sigma_\omega)$$

is true. By definition of $\mathrm{SIGN}_{\Pi}(\mathcal{P})$, $\sigma_1 = \mathrm{SIGN}_{\Pi'}(\mathcal{P})(a_{[1]})$ for some $a_{[1]} \in \mathrm{R}^{k_1}$, and hence

$$\mathrm{Qu}_2\sigma_2 \in \mathrm{SIGN}_{\Pi'}(\mathcal{P}_{a_{[1]}}) ... \mathrm{Qu}_\omega\sigma_\omega \in \sigma_{\omega-1} \quad F^\star(\sigma_\omega)$$

is true. By induction hypothesis,

$$(\mathrm{Qu}_2 X_{[2]}) ... \ (\mathrm{Qu}_\omega X_\omega)\ F(a_{[1]}, X_{[2]}, ..., X_{[\omega]})$$

is true. Thus

$$(\exists X_{[1]})\ (\mathrm{Qu}_2 X_{[2]}) ... \ (\mathrm{Qu}_\omega X_{[\omega]}) F(X)$$

is true. □

In the cylindrical situation studied in Chapter 11, $\mathrm{CSIGN}(\mathcal{P})$ was obtained from a cylindrical set of sample points of a cylindrical decomposition adapted to $\mathcal{P}$. We generalize this approach to a general block structure.

A **Π-set** $\mathcal{A} = \mathcal{A}_1, ..., \mathcal{A}_\omega$ is such that $\mathcal{A}_i$ is a finite set contained in $\mathrm{R}^{[i]}$ and $\pi_{[i]}(\mathcal{A}_{i+1}) = \mathcal{A}_i$.

We define inductively the **tree of realizable sign conditions of $\mathcal{P}$ for $\mathcal{A}$ with respect to** Π, $\mathrm{SIGN}_{\Pi}(\mathcal{P}, \mathcal{A})$, as follows:

- For $z \in \mathcal{A}_\omega$, let $\mathrm{SIGN}_{\Pi,\omega}(\mathcal{P})(z) = \mathrm{sign}(\mathcal{P})(z)$, where $\mathrm{sign}(\mathcal{P})(z)$ is the sign condition on $\mathcal{P}$ mapping $P \in \mathcal{P}$ to $\mathrm{sign}(P)(z) \in \{0, 1, -1\}$ (Notation 11.7).
- For all i, $1 \leq i < \omega$, and all $y \in \mathcal{A}_i$, we inductively define,

$$\mathrm{SIGN}_{\Pi,i}(\mathcal{P}, \mathcal{A})(y) = \{\mathrm{SIGN}_{\Pi,i+1}(\mathcal{P}, \mathcal{A})(z) | z \in \mathcal{A}_{i+1}, \pi_{[i]}(z) = y\}.$$

Finally, we define

$$\mathrm{SIGN}_{\Pi}(\mathcal{P}, \mathcal{A}) = \mathrm{SIGN}_{\Pi,0}(\mathcal{P}, \mathcal{A})(0).$$

Note that $\mathrm{SIGN}_{\Pi}(\mathcal{P}) = \mathrm{SIGN}_{\Pi}(\mathcal{P}, \mathrm{R}^k)$. Note also that $\mathrm{SIGN}_{\Pi}(\mathcal{P}, \mathcal{A})$ is a subtree of $\mathrm{SIGN}_{\Pi}(\mathcal{P})$.

A **Π-set of sample points** for $\mathcal{P}$ is a Π-set $\mathcal{A} = \mathcal{A}_1, ..., \mathcal{A}_\omega$ such that

$$\mathrm{SIGN}_{\Pi}(\mathcal{P}, \mathcal{A}) = \mathrm{SIGN}_{\Pi}(\mathcal{P}).$$

A **Π-partition adapted to** $\mathcal{P}$ is given by $\mathcal{S} = \mathcal{S}_1, ..., \mathcal{S}_\omega$, where $\mathcal{S}_i$ is a partition of $\mathrm{R}^{[i]}$ into a finite number of semi-algebraically connected semi-algebraic sets such that for every $S \in \mathcal{S}_{i+1}$, $\pi_{[i]}(S) \in \mathcal{S}_i$, and such that every $S \in \mathcal{S}_\omega$ is $\mathcal{P}$- invariant. A **Π-set of sample points** for a Π-partition $\mathcal{S}$ is a Π-set $\mathcal{A} = \mathcal{A}_1, ..., \mathcal{A}_\omega$ such that

- for every i, $1 \leq i \leq \omega$, $\mathcal{A}_i$ intersects every $S \in \mathcal{S}_i$,
- for every i, $1 \leq i \leq \omega - 1$, $\pi_{[i]}(\mathcal{A}_{i+1}) = \mathcal{A}_i$.

Note that the partition of R^k by the semi-algebraically connected components of realizable sign conditions of $\mathcal{P}$ is a Π-partition with the block structure $\Pi = \{X_1..., X_k\}$ (i.e. with a single block), and a set of sample points for $\mathcal{P}$ is a Π-set of sample points for this block structure. Note also that a cylindrical decomposition $\mathcal{S}$ adapted to $\mathcal{P}$ is a Π-partition for the block structure $X_1, ..., X_k$ (k-blocks of one variable) and a cylindrical set of sample points for $\mathcal{S}$ is a Π-set of sample points for $\mathcal{S}$ for this block structure.

We are going to prove a result generalizing Proposition 11.9 to the case of a general block structure.

Proposition 14.4. *Let $\mathcal{S} = \mathcal{S}_1, ..., \mathcal{S}_\omega$ be a Π-partition of R^k adapted to $\mathcal{P}$ and $\mathcal{A} = \mathcal{A}_1, ..., \mathcal{A}_\omega$ be a Π-set of sample points for $\mathcal{S}$. Then $\mathcal{A}$ is a Π-set of sample points for $\mathcal{P}$.*

The proof is similar to the proof of Proposition 11.9 and uses the following generalization of Proposition 11.11.

Proposition 14.5. *Let $\mathcal{S} = \mathcal{S}_1, ..., \mathcal{S}_\omega$ be a Π-partition of R^k adapted to $\mathcal{P}$. For every $1 \leq i \leq \omega$ and every $S \in \mathcal{S}_i$, $\mathrm{SIGN}_{\Pi,i}(y)$ is constant as y varies in S.*

Proof: The proof is by induction on $\omega - i$.

If $i = \omega$, the claim is true since the sign of every $P \in \mathcal{P}$ is fixed on $S \in \mathcal{S}_\omega$.

Suppose that the claim is true for $i+1$ and consider $S \in \mathcal{S}_i$. Let $T_1, ..., T_\ell$ be the elements of $\mathcal{S}_{i+1}$ such that $\pi_{[i]}(T_j) = S$. By induction hypothesis, $\mathrm{SIGN}_{\Pi,i+1}(\mathcal{P})(z)$ is constant as z varies in T_j. Since $\mathcal{S}$ is a Π-partition, $\bigcup_{j=1}^{\ell} T_j = S \times \mathrm{R}^{k_{i+1}}$. Thus

$$\mathrm{SIGN}_{\Pi,i}(\mathcal{P})(y) = \{\mathrm{SIGN}_{\Pi,i+1}(\mathcal{P})(z) | z \in \mathrm{R}^{[i+1]}, \pi_{[i]}(z) = y\}$$

is constant as y varies in S. □

Proof of Proposition 14.4: Let $\mathcal{A}_0 = \{0\}$. We are going to prove that for every $y \in \mathcal{A}_i$,

$$\mathrm{SIGN}_{\Pi,i}(\mathcal{P})(y) = \mathrm{SIGN}_{\Pi,i}(\mathcal{P}, \mathcal{A})(y).$$

The proof is by induction on $\omega - i$.

If $i = \omega$, the claim is true since $\mathcal{A}_\omega$ meets every element of $\mathcal{S}_\omega$.

Suppose that the claim is true for $i+1$ and consider $y \in \mathcal{A}_i$. Let S be the element of $\mathcal{S}_i$ containing y, and let T_1, ..., T_ℓ be the elements of $\mathcal{S}_{i+1}$ such that $\pi_{[i]}(T_j) = S$. Denote by z_j a point of $T_j \cap \mathcal{A}_{i+1}$ such that $\pi_{[i]}(z_j) = y$. By induction hypothesis,

$$\mathrm{SIGN}_{\Pi,i+1}(\mathcal{P})(z_j) = \mathrm{SIGN}_{\Pi,i+1}(\mathcal{P},\mathcal{A})(z_j).$$

Since $T_1 \cup ... \cup T_\ell = S \times \mathrm{R}^{k_{i+1}}$ and $\mathrm{SIGN}_{\Pi,i+1}(\mathcal{P})(z)$ does not change as z varies over T_j,

$$\begin{aligned}\mathrm{SIGN}_{\Pi,i}(\mathcal{P})(y) &= \{\mathrm{SIGN}_{\Pi,i+1}(\mathcal{P})(z) | z \in \mathrm{R}^{[i+1]}, \pi_{[i]}(z) = y\} \\ &= \{\mathrm{SIGN}_{\Pi,i+1}(\mathcal{P},\mathcal{A})(z) | z \in \mathcal{A}_{i+1}, \pi_{[i]}(z) = y\} \\ &= \mathrm{SIGN}_{\pi,i}(\mathcal{P},\mathcal{A})(y).\end{aligned}$$

□

We now construct a Π-partition of R^k adapted to $\mathcal{P}$, generalizing Theorem 5.6. Note that a cylindrical decomposition adapted to $\mathcal{P}$ gives a Π-partition of R^k adapted to $\mathcal{P}$, so the issue here is not an existence theorem similar to Theorem 5.6 but rather a complexity result taking into account the block structure Π. The construction of a cylindrical decomposition adapted to $\mathcal{P}$ in Chapter 5 and Chapter 11 was based on a recursive call to an Elimination procedure eliminating one variable (see Algorithm 11.1 (Subresultant Elimination)). In the general block structure context, we define a Block Elimination procedure which replaces a block of variables by one single variable and computes parametrized univariate representations, giving in a parametric way sample points for every non-empty sign condition. Finally we eliminate this variable.

Algorithm 14.1. **[Block Elimination]**

- **Structure:** an ordered domain D contained in a real closed field R.
- **Input:** a block of variables $X = (X_1, ..., X_k)$ and a set of polynomials

$$\mathcal{P}(Y) \subset \mathrm{D}[Y_1, ..., Y_\ell, X_1, ..., X_k].$$

- **Output:**
  - a set $\mathrm{BElim}_X(\mathcal{P}) \subset \mathrm{D}[Y]$ such that $\mathrm{SIGN}(\mathcal{P}(y, X_1, ..., X_k))$ (Notation 7.29) is fixed as y varies over a semi-algebraically connected component of a realizable sign condition of $\mathrm{BElim}_X(\mathcal{P})$,
  - a set $\mathrm{UR}_X(\mathcal{P})$ of parametrized univariate representations of the form

$$u(Y, \varepsilon, \delta) = (f, g_0, ..., g_k),$$

where $f, g_i \in \mathrm{D}[Y, \varepsilon, \delta][T]$. For any point $y \in \mathrm{R}^\ell$, denoting by $\mathrm{UR}_X(\mathcal{P})(y)$ the subset of $\mathrm{UR}_X(\mathcal{P})$ such that $f(y, T)$ and $g_0(y, T)$ are coprime, the points associated to the univariate representations in $\mathrm{UR}_X(\mathcal{P})(y)$ intersect every semi-algebraically connected component of every realizable sign condition of the set $\mathcal{P}(y)$ in $\mathrm{R}\langle\varepsilon,\delta\rangle^k$.

- **Complexity:** $s^{k+1}\, d^{O(\ell k)}$, where s is a bound on the number of elements of $\mathcal{P}$ and d is a bound on their degree.
- **Procedure:**
  - Initialize $\mathrm{UR}_X(\mathcal{P})$ to the empty set.
  - Take as d' the smallest even natural number $> d$.
  - Define

$$\begin{aligned} P_i^\star &= \{(1-\delta)\, P_i + \delta\, H_k(d', i), (1-\delta)\, P_i - \delta\, H_k(d', i), \\ &\quad (1-\delta)\, P_i + \delta\, \gamma\, H_k(d', i), (1-\delta)\, P_i - \delta\, \gamma\, H_k(d', i)\} \\ \mathcal{P}^\star &= \{P_1^\star, ..., P_s^\star\} \end{aligned}$$

    for $0 \leq i \leq s$ using Notation 13.4.
  - For every subset $\mathcal{Q}$ of $j \leq k$ polynomials $Q_{i_1} \in P_{i_1}^\star, ..., Q_{i_j} \in P_{i_j}^\star$,
    - let $Q = Q_{i_1}^2 + \cdots + Q_{i_j}^2 + (\varepsilon^2\, (X_1^2 + \cdots + X_k^2 + X_{k+1}^2) - 1)^2$.
    - Take for $i = 1, ..., k$, $\bar{d}_i$ the smallest even natural number $> \deg(Q)$, $\bar{d}_{k+1} = 6$, $\bar{d} = (\bar{d}_1, ..., \bar{d}_k, \bar{d}_{k+1})$, and $c = \varepsilon$.
    - Perform Algorithm 12.10 (Parametrized Multiplication Table) with input $\overline{\mathrm{Cr}}(Q, \zeta)$ (using Notation 12.46). Output $\mathcal{M}$.
    - Perform Algorithm 12.15 (Parametrized Limit of Bounded Points) with input $\gamma, \zeta, \overline{\mathrm{Cr}}(Q, \zeta)$, and $\mathcal{M}$. Add the parametrized univariate representations (belonging to $\mathrm{D}[Y, \varepsilon, \delta][T]^{k+2}$) output to $\mathrm{UR}_X(\mathcal{P})$.
  - For every $v = (f, g_0, ..., g_k) \in \mathrm{UR}_X(\mathcal{P})$, consider the family of univariate polynomials $\mathcal{F}_v$ consisting of f, its derivatives with respect to T, and P_v (see Notation 13.8), for every $P \in \mathcal{P}$. Compute $\mathrm{RElim}_T(f, \mathcal{F}_v)$ using Algorithm 11.19 (Restricted Elimination). Denote by $\mathcal{B}_v$ the family of polynomials in Y that are the coefficients of the polynomials in

$$\mathrm{RElim}_T(\mathcal{F}_v) \subset \mathrm{D}[Y, \varepsilon, \delta].$$

  - Define $\mathrm{BElim}_X(\mathcal{P})$ to be the union of the sets $\mathcal{B}_v \subset \mathrm{D}[Y]$ for every $v \in \mathrm{UR}_X(\mathcal{P})$.
  - Output $\mathrm{BElim}_X(\mathcal{P})$ and $\mathrm{UR}_X(\mathcal{P})$.

The proof of correctness of Algorithm 14.1 (Block Elimination) uses the following results, which describe how to get rid of infinitesimal quantities.

Notation 14.6. Let $\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_m$ be variables and consider the real closed field $\mathrm{R}\langle \varepsilon_1, \varepsilon_2, \cdots, \varepsilon_m \rangle$. Let $S \subset \mathrm{R}\langle \epsilon_1, ..., \epsilon_m \rangle^k$ be a semi-algebraic set defined by a quantifier-free $\mathcal{P}$-formula Φ with $\mathcal{P} \subset \mathrm{D}[\epsilon_1, ..., \epsilon_m, X_1, ..., X_k]$Let $P \in \mathcal{P}$. We write P as a polynomial in $\epsilon = (\epsilon_1, ..., \epsilon_m)$ and order the monomials with the order induced by the order $<_\varepsilon$ on $\mathrm{R}\langle \varepsilon_1, \varepsilon_2, \cdots, \varepsilon_m \rangle$ with $\epsilon_1 >_\varepsilon ... >_\varepsilon \epsilon_m$. Let

$$P = P_0 \epsilon^{\alpha_0} + P_1 \epsilon^{\alpha_1} + \cdots + P_m \epsilon^{\alpha_m},$$

where, $P_i \in \mathrm{D}[X_1, ..., X_k]$, $\alpha_i \in \mathbb{N}^\ell$, and $\epsilon^{\alpha_i} >_\varepsilon \epsilon^{\alpha_{i+1}}$, for $0 \leq i \leq m$.

Define

$$\begin{aligned}
\mathrm{Remo}_\varepsilon(P=0) &= \bigwedge_{i=0}^{m} (P_i=0),\\
\mathrm{Remo}_\varepsilon(P>0) &= (P_0>0) \vee (P_0=0 \wedge P_1>0) \vee \cdots \vee (\bigwedge_{i=0}^{m-1} P_i=0 \wedge P_m>0),\\
\mathrm{Remo}_\varepsilon(P<0) &= (P_0<0) \vee (P_0=0 \wedge P_1<0) \vee \cdots \vee (\bigwedge_{i=0}^{m-1} P_i=0 \wedge P_m<0).
\end{aligned}$$

Let $\mathrm{Remo}_\varepsilon(\Phi)$ be the formula obtained from Φ by replacing every atom, $P=0$, $P>0$, or $P<0$ in Φ by the corresponding formula

$$\mathrm{Remo}_\varepsilon(P=0), \mathrm{Remo}_\varepsilon(P>0), \mathrm{Remo}_\varepsilon(P<0). \qquad \square$$

Proposition 14.7. *Let $S \subset \mathrm{R}\langle\epsilon_1, ..., \epsilon_m\rangle^k$ be a semi-algebraic set defined by a quantifier-free $\mathcal{P}$-formula Φ with $\mathcal{P} \subset \mathrm{D}[\epsilon_1, ..., \epsilon_m, X_1, ..., X_k]$. Let $S' \subset \mathrm{R}^k$ be the semi-algebraic set defined by* $\mathrm{Remo}_\varepsilon(\Phi)$. *Then, $S' = S \cap \mathrm{R}^k$.*

Proof: Let $x \in \mathrm{R}^k$ satisfy $\mathrm{Remo}_\varepsilon(\Phi)$. It is clear by construction that x also satisfies Φ. Conversely, if $x \in S \cap \mathrm{R}^k$, then for any polynomial

$$P \in \mathrm{D}[\epsilon_1, ..., \epsilon_m, X_1, ..., X_k],$$

the sign of $P(x)$ is determined by the sign of the coefficient of the biggest monomial in the lexicographical ordering, when $P(x)$ is expressed as a polynomial in ϵ_1, ..., ϵ_m. This immediately implies that x satisfies the formula $\mathrm{Remo}_\varepsilon(\Phi)$. $\square$

Proof of correctness of Algorithm 14.1: The result follows from the correctness of Algorithm 13.1 (Computing Realizable Sign Conditions) and Algorithm 11.19 (Restricted Elimination). Consider a semi-algebraically connected component S of a realizable sign condition on $\mathrm{BElim}_X(\mathcal{P})$. Then, the following remain invariant as y varies over S: the set $\mathrm{UR}_X(\mathcal{P})(y)$, for every

$$u(Y, \varepsilon, \delta) = (f(Y, \varepsilon, \delta, T), g_0(Y, \varepsilon, \delta, T), ..., g_k(Y, \varepsilon, \delta, T)) \in \mathrm{UR}_X(\mathcal{P})(y),$$

the number of roots of $f(y, \varepsilon, \delta, T)$ in $\mathrm{R}\langle\varepsilon, \delta\rangle$ and their Thom encodings, as well as the number of roots in $\mathrm{R}\langle\varepsilon, \delta\rangle$ that are common to $f(y, \varepsilon, \delta, T)$ and $P_u(y, \varepsilon, \delta, T)$ for all $P \in \mathcal{P}$. These are consequences of the properties of RElim_T (see Algorithm 11.19 (Restricted Elimination)). It is finally clear that $\mathrm{SIGN}(\mathcal{P}(y, X))$ is constant as y varies in a semi-algebraically connected component S of a realizable sign condition on $\mathrm{BElim}_X(\mathcal{P})$, using Proposition 14.7. $\square$

Complexity analysis of Algorithm 14.1: The number of arithmetic operations in $\mathrm{D}[Y, \varepsilon, \delta, \gamma, \zeta]$ for computing

$$\mathrm{UR}_X(\mathcal{P}) \subset \mathrm{D}[Y, \varepsilon, \delta][T]$$

is $\sum_{j\leq k} 4^j \binom{s}{j} d^{O(k)}$, using the complexity analysis of Algorithm 13.1 (Computing Realizable Sign Conditions). The degrees of the polynomials in T generated in this process are bounded by $O(d)^k$ (independent of ℓ), and the degree in the variables Y as well as in the variables ε and δ is $d^{O(k)}$, using the complexity analysis of Algorithm 12.10 (Parametrized Multiplication Table) and Algorithm 12.15 (Parametrized Limit of Bounded Points).

The complexity in D for computing $\mathrm{UR}_X(\mathcal{P})$ is $\sum_{j\leq k} 4^j \binom{s}{j} d^{O(\ell k)}$, using the complexity analysis of Algorithm 8.4 (Addition of multivariate polynomials) and Algorithm 8.5 (Multiplication of multivariate polynomials).

Using the complexity of Algorithm 11.19 (Restricted Elimination), the size of the set $\mathrm{BElim}_X(\mathcal{P})$ is $s\sum_{j\leq k} 4^j \binom{s}{j} d^{O(k)} = s^{k+1} d^{O(k)}$, and the degrees of the elements of $\mathrm{BElim}_X(\mathcal{P})$ is $d^{O(k)}$.

The complexity in D is finally $s\sum_{j\leq k} 4^j \binom{s}{j} d^{O(\ell k)} = s^{k+1} d^{O(\ell k)}$.

If $\mathrm{D} = \mathbb{Z}$, and the bitsizes of the coefficients of the polynomials are bounded by τ, then the bitsizes of the integers appearing in the intermediate computations and the output are bounded by $\tau d^{O(k)}$. □

We construct a Π-partition adapted to $\mathcal{P}$ using recursive calls to Algorithm 14.1 (Block Elimination).

Notation 14.8. Defining $\mathrm{B}_{\Pi,\omega}(\mathcal{P}) = \mathcal{P}$, we denote, for $1 \leq i \leq \omega - 1$,

$$\mathrm{B}_{\Pi,i}(\mathcal{P}) = \mathrm{BElim}_{X_{[i+1]}}(\mathrm{B}_{\Pi,i+1}(\mathcal{P})),$$

so that $\mathrm{B}_{\Pi,i}(\mathcal{P}) \subset \mathrm{R}[X_{[1]}, ..., X_{[i]}]$. □

For every i, $1 \leq i \leq \omega$, let $\mathcal{S}_i$ be the set of semi-algebraically connected components of non-empty realizations of sign conditions on $\bigcup_{j=1}^{i} \mathrm{B}_{\Pi,i}(\mathcal{P})$.

The following proposition follows clearly from the correctness of Algorithm 14.1 (Block Elimination).

Proposition 14.9. *The list $\mathcal{S} = \mathcal{S}_1, ..., \mathcal{S}_\omega$ is a Π-partition adapted to $\mathcal{P}$.*

In order to describe a Π-set of sample points for $\mathcal{S}$, we are going to use the parametrized univariate representations computed in Algorithm 14.1 (Block Elimination).

Notation 14.10. *Note that for every $i = \omega - 1, ..., 0$,*

$$\mathrm{UR}_{\Pi,i}(\mathcal{P}) = \mathrm{UR}_{X_{[i+1]}}\mathrm{B}_{\Pi,i+1}(\mathcal{P})).$$

The elements of $\mathrm{UR}_{\Pi,i}(\mathcal{P})$ are parametrized univariate representations in the variable T_{i+1}, contained in $\mathrm{D}[X_{[1]}, ..., X_{[i]}, \varepsilon_{i+1}, \delta_{i+1}][T_{i+1}]^{k_{i+1}+2}$. Let

$$u = (u_0, ..., u_{\omega-1}) \in \mathcal{U} = \prod_{i=0}^{\omega-1} \mathrm{UR}_{\Pi,i}(\mathcal{P}),$$

with

$$u_{i-1} = (f^{[i]}, g_0^{[i]}, g_1^{[i]}, ..., g_{k_i}^{[i]}).$$

For a polynomial $P(X_{[1]},...,X_{[i]})$, let $P_{u,i..j}(X_{[1]},...,X_{[j-1]},T_j,...,T_i)$ denote the polynomial obtained by successively replacing the blocks of variables $X_{[\ell]}$, with the rational fractions associated with the tuple $u_{\ell-1}$ (using Notation ?), for ℓ from i to j. Denoting $P_{u,i}(T_1,...,T_i) = P_{u,i..1}(T_1,...,T_i)$, define

$$\begin{aligned}
\mathcal{T}_{u,i} &= (f^{[1]}(T_1), f^{[2]}_{u,1}(T_1,T_2),..., f^{[i]}_{u,i-1}(T_1,T_2,...,T_i)),\\
\mathcal{T}_{u} &= (f^{[1]}(T_1), f^{[2]}_{u,1}(T_1,T_2),..., f^{[\omega]}_{u,\omega-1}(T_1,T_2,...,T_\omega)),\\
\bar{u}_{i-1} &= (f^{[i]}_{u,i-1}, {g_0^{[i]}}_{u,i-1}, {g_1^{[i]}}_{u,i-1},..., {g_{k_i}^{[i]}}_{u,i-1})
\end{aligned}$$

Note that $\bar{u}_{i-1}$ are univariate representations contained in

$$\mathrm{D}[T_1,...,T_{i-1},\varepsilon_1,\delta_1,...,\varepsilon_i,\ \delta_i][T_i]^{k_i+2}.$$

For $u \in \mathcal{U}$ and $t_\sigma \in \mathrm{Zer}(\mathcal{T}_u, \mathrm{R}\langle\varepsilon_1,\ \delta_1,\ ...,\ \varepsilon_\omega,\ \delta_\omega\rangle)$, with Thom encoding σ let $x_{u,\sigma,i} \in \mathrm{R}\langle\varepsilon_1,\ \delta_1,\ ...,\ \varepsilon_\omega,\ \delta_\omega\rangle^{[i]}$ be the point obtained by substituting t_σ in the rational functions associated to $\bar{u}_{j-1}$, $j \leq i$. Let $\mathcal{A}_i$ be the set of points $x_{u,\sigma,i} \in \mathrm{R}\langle\varepsilon_1,\ \delta_1,\ ...,\ \varepsilon_\omega,\ \delta_\omega\rangle^{[i]}$ obtained by considering all $u \in \mathcal{U}$ and $t_\sigma \in \mathrm{Zer}(\mathcal{T}_u, \mathrm{R}\langle\varepsilon_1,\delta_1,...,\varepsilon_\omega,\delta_\omega\rangle)$. Then $\mathcal{A} = \mathcal{A}_1,...,\mathcal{A}_\omega$ is a Π-set, specified by $\mathcal{V}$ where the elements of $\mathcal{V}$ are pairs of an element $u \in \mathcal{U}$ and a Thom encoding σ of an element of $\mathrm{Zer}(\mathcal{T}_u, \mathrm{R}\langle\varepsilon_1,\delta_1,...,\varepsilon_\omega,\delta_\omega\rangle)$. □

The correctness of Algorithm 14.1 (Block Elimination) implies the following proposition.

Proposition 14.11. *The set $\mathcal{A}$ is a Π-set of sample points for $\mathcal{P}$.*

Thus, in order to construct the set $\mathrm{SIGN}_\Pi(\mathcal{P})$, it suffices to compute the signs of $\mathcal{P}_{u,\omega}$ at the zeros of $\mathcal{T}_u$, $u \in \mathcal{U}$.

The algorithm is as follows, using the notation of Algorithm 14.1 (Block Elimination):

Algorithm 14.2. **[Block Structured Signs]**

- **Structure:** an ordered domain D contained in a real closed field R.
- **Input:** a set $\mathcal{P} \subset \mathrm{R}[X_1,...,X_k]$, and a partition, Π, of the variables $X_1,...,X_k$ into blocks, $X_{[1]},...,X_{[\omega]}$.
- **Output:** the tree $\mathrm{SIGN}_\Pi(\mathcal{P})$ of realizable sign conditions of $\mathcal{P}$ with respect to Π.
- **Complexity:** $s^{(k_\omega+1)\cdots(k_1+1)}\ d^{O(k_\omega)\cdots O(k_1)}$, where s is bound on the number of elements of $\mathcal{P}$, d is a bound on their degree, and $k_{[i)}$ is the number of elements of $X_{[i]}$.
- **Procedure:**
  - Initialize $\mathrm{B}_{\Pi,\omega}(\mathcal{P}) := \mathcal{P}$.
  - Block Elimination Phase: Compute

$$\mathrm{B}_{\Pi,i}(\mathcal{P}) = \mathrm{BElim}_{X_{[i+1]}}(\mathrm{B}_{\Pi,i+1}(\mathcal{P})),$$

for $1 \leq i \leq \omega - 1$, applying repeatedly $\mathrm{BElim}_{X_{[i+1]}}$, using Algorithm 14.1 (Block Elimination). Define $\mathrm{B}_{\Pi,0}(\mathcal{P}) = \{1\}$. Compute $\mathrm{UR}_{\Pi,i}(\mathcal{P})$, for every $i = \omega - 1, ..., 0$, using Algorithm 14.1 (Block Elimination). The elements of $\mathrm{UR}_{\Pi,i}(\mathcal{P})$ are parametrized univariate representations in the variable T_{i+1}, contained in

$$\mathrm{D}[X_{[1]}, ..., X_{[i]}, \varepsilon_{i+1}, \delta_{i+1}][T_{i+1}]^{k_{i+1}+2}.$$

- Substitution Phase: Compute the set of pairs $\{(\mathcal{T}_u, \mathcal{P}_{u,\omega}) \mid u \in \mathcal{U}\}$, using their definition in Notation 14.10.
- Sign Determination Phase: Compute the signs of the set of the polynomials in $\mathcal{P}_{u,\omega}$ on $\mathrm{Zer}(\mathcal{T}_u, \mathrm{R}\langle\varepsilon_1, \delta_1, ..., \varepsilon_\omega, \delta_\omega\rangle^\omega)$ using Algorithm 12.19 (Zero-dimensional Sign Determination).
- Construct the set $\mathrm{SIGN}_\Pi(\mathcal{P})$ from these signs.

Proof of correctness: The correctness of the algorithm follows from Proposition 14.11. □

Complexity analysis: Using the complexity of Algorithm 14.1 (Block Elimination), the degrees and number of the parametrized univariate representations in $\mathrm{UR}_{\Pi,\omega-1}(\mathcal{P})$ produced after eliminating the first block of variables $X_{[\omega]}$ are bounded respectively by $O(d)^{k_\omega}$ and $s^{k_\omega}O(d)^{k_\omega}$. The number of arithmetic operations in this step is bounded by $s^{k_\omega}d^{O((k-k_\omega)k_\omega)}$, and the size of the set $\mathrm{B}_{\Pi,\omega-1}(\mathcal{P})$ is $s^{k_\omega+1}d^{O(k_\omega)}$. Since the cardinality of $\mathrm{SIGN}_{\Pi,\omega-1}(\mathcal{P})(z)$ is, for every $z \in \mathrm{R}^{[\omega-1]}$, bounded by the number of points associated to the univariate representations obtained by substituting z to the parameters in the elements of $\mathrm{UR}_{\Pi,\omega-1}(\mathcal{P})$, $\#(\mathrm{SIGN}_{\Pi,\omega-1}(\mathcal{P})(z))$ is $s^{k_\omega}O(d)^{k_\omega}$.

An easy inductive argument shows that the number of univariate representations in $\mathrm{UR}_{\Pi,i}(\mathcal{P})$ produced after eliminating the $(i+1)$-th block of variables is bounded by

$$s^{(k_\omega+1)\cdots(k_{i+2}+1)k_{i+1}}d^{O(k_\omega)\cdots O(k_{i+1})}.$$

By a similar argument, one can show that the degrees of the parametrized univariate representations in $\mathrm{UR}_{\Pi,i}(\mathcal{P})$ are bounded by $d^{O(k_\omega)\cdots O(k_{i+1})}$. The complexity in D is bounded by

$$s^{(k_\omega+1)\cdots(k_{i+1}+1)}d^{(k_1+\cdots+k_i+2(\omega-i))O(k_\omega)\cdots O(k_{i+1})},$$

since the arithmetic is done in a polynomial ring with $k_1 + \cdots + k_i + 2(\omega - i)$ variables.

A similar inductive argument shows that the the size of the set $\mathrm{B}_{\Pi,i}(\mathcal{P})$ is bounded by $s^{(k_\omega+1)\cdots(k_{i+1}+1)}d^{O(k_\omega)\cdots O(k_{i+1})}$, and their degrees are bounded by $d^{O(k_\omega)\cdots O(k_i)}$.

The above analysis shows that the size of the set of pairs $(\mathcal{T}_u, \mathcal{P}_u)$, constructed at the end of the Substitution Phase is

$$s^{(k_\omega+1)\cdots(k_1+1)}d^{O(k_\omega)\cdots O(k_1)},$$

and the degrees are bounded by $d^{O(k_\omega)\cdots O(k_1)}$. It should also be clear that the number of arithmetic operations in D for the Substitution Phase is equally bounded by

$$s^{(k_\omega+1)\cdots(k_1+1)}d^{O(k_\omega)\cdots O(k_1)}.$$

Since the number of triangular systems $\mathcal{T}$ is

$$s^{(k_\omega+1)\cdots(k_1+1)}d^{O(k_\omega)\cdots O(k_1)}.$$

and each call to Algorithm 12.19 (Triangular Sign Determination) takes time

$$d^{\omega O(k_\omega)\cdots O(k_1)} = d^{O(k_\omega)\cdots O(k_1)},$$

the time taken for the Sign Determination Phase, is

$$s^{(k_\omega+1)\cdots(k_1+1)}d^{O(k_\omega)\cdots O(k_1)}.$$

The time required to construct $\mathrm{SIGN}_\Pi(\mathcal{P})$ is again bounded by

$$s^{(k_\omega+1)\cdots(k_1+1)}d^{O(k_\omega)\cdots O(k_1)}.$$

Thus the total time bound for the elimination and sign determination phase is

$$s^{(k_\omega+1)\cdots(k_1+1)}d^{O(k_\omega)\cdots O(k_1)}.$$

If $\mathrm{D}=\mathbb{Z}$, and the bitsizes of the coefficients of the polynomials are bounded by τ, then the bitsizes of the integers appearing in the intermediate computations and the output are bounded by $\tau d^{O(k_\omega)\cdots O(k_1)}$. □

Remark 14.12. In fact, Algorithm 14.2 (Block Structured Signs) does not only computes $\mathrm{SIGN}_\Pi(\mathcal{P})$*, it also produces the set* $\mathcal{V}$ *specifying a* Π*-set of sampling points for* $\mathcal{P}$ *described at the end of Notation 14.10.* □

We have proved the following result:

Theorem 14.13. *Let* $\mathcal{P}$ *be a set of at most* s *polynomials each of degree at most* d *in* k *variables with coefficients in a real closed field* R*, and let* Π *denote a partition of the list of variables* $(X_1,\ldots,X_k)$ *into blocks* $X_{[1]},\ldots,X_{[\omega]}$*, where the block* $X_{[i]}$ *has size* $k_i, 1\leq i\leq\omega$*. Then the size of the set* $\mathrm{SIGN}_\Pi(\mathcal{P})$ *is bounded by*

$$s^{(k_\omega+1)\cdots(k_1+1)}d^{O(k_\omega)\cdots O(k_1)}.$$

Moreover, there exists an algorithm which computes this set with complexity

$$s^{(k_\omega+1)\cdots(k_1+1)}d^{O(k_\omega)\cdots O(k_1)}$$

in D*, where* D *is the ring generated by the coefficients of* $\mathcal{P}$*.*

If $\mathrm{D}=\mathbb{Z}$*, and the bitsizes of the coefficients of the polynomials are bounded by* τ*, then the bitsizes of the integers appearing in the intermediate computations and the output are bounded by* $\tau d^{O(k_\omega)\cdots O(k_1)}$*.*

Using the set $\mathrm{SIGN}_{\Pi}(\mathcal{P})$, it is now easy to solve the general decision problem, which is to design a procedure to decide the truth or falsity of a $(\mathcal{P}, \Pi)$-sentence.

Algorithm 14.3. **[General Decision]**

- **Structure:** an ordered domain D contained in a real closed field R
- **Input:** a $(\mathcal{P}, \Pi)$-sentence Φ, where $\mathcal{P} \subset \mathrm{D}[X_1, ..., X_k]$, and Π is a partition of the variables $X_1, ..., X_k$ into blocks $X_{[1]}, ..., X_{[\omega]}$.
- **Output:** 1 if Φ is true and 0 otherwise.
- **Complexity:** $s^{(k_\omega+1)\cdots(k_1+1)}\ d^{O(k_\omega)\cdots O(k_1)}$, where s is a bound on the number of elements of $\mathcal{P}$, d is a bound on their degree, and k_i is the number of elements of $X_{[i]}$.
- **Procedure:**
  - Compute $\mathrm{SIGN}_{\Pi}(\mathcal{P})$.
  - Trying all possibilities, decide whether

$$\mathrm{Qu}_1\sigma_1 \in \mathrm{SIGN}_{\Pi}(\mathcal{P}) \quad \mathrm{Qu}_2\sigma_2 \in \sigma_1 ... \mathrm{Qu}_\omega\sigma_\omega \in \sigma_{\omega-1} \quad F^\star(\sigma_\omega) = \mathrm{True},$$

    which is clearly a finite verification.

Proof of correctness: Follows from the properties of $\mathrm{SIGN}_{\Pi}(\mathcal{P})$. □

Complexity analysis: Given the complexity of Algorithm 14.2 (Block Structured Signs), the complexity for the general decision algorithm is

$$s^{(k_\omega+1)\cdots(k_1+1)} d^{O(k_\omega)\cdots O(k_1)}$$

in D. Note that the evaluation of the boolean formulas are not counted in this model of complexity since we count only arithmetic operations in D.

If $\mathrm{D} = \mathbb{Z}$, and the bitsizes of the coefficients of the polynomials are bounded by τ, then the bitsizes of the integers appearing in the intermediate computations and the output are bounded by $\tau d^{O(k_\omega)\cdots O(k_1)}$. □

Note that the first step of the computation depend only on $(\mathcal{P}, \Pi)$ and not on Φ. As noted before $\mathrm{SIGN}_{\Pi}(\mathcal{P})$ allows to decide the truth or falsity of every $(\mathcal{P}, \Pi)$-sentence.

We have proved the following result.

Theorem 14.14. [General Decision] *Let $\mathcal{P}$ be a set of at most s polynomials each of degree at most d in k variables with coefficients in a real closed field* R*, and let Π denote a partition of the list of variables $(X_1, ..., X_k)$ into blocks $X_{[1]}, ..., X_{[\omega]}$, where the block $X_{[i]}$ has size $k_i, 1 \leq i \leq \omega$. Given a $(\mathcal{P}, \Pi)$-sentence Φ, there exists an algorithm to decide the truth of Φ with complexity*

$$s^{(k_\omega+1)\cdots(k_1+1)} d^{O(k_\omega)\cdots O(k_1)}$$

in D*, where* D *is the ring generated by the coefficients of $\mathcal{P}$. If $\mathrm{D} = \mathbb{Z}$, and the bitsizes of the coefficients of the polynomials are bounded by τ, then the bitsizes of the integers appearing in the intermediate computations and the output are bounded by $\tau d^{O(k_\omega)\cdots O(k_1)}$.*

14.2 Quantifier Elimination

In our Quantifier Elimination Algorithm, we use a parametrized version of Algorithm 12.8 (Multivariate Sign Determination) to solve the following problem.

Notation 14.15. *Let* D *be a ring contained in a real closed field* R. *A* ***parametrized triangular system*** *with parameters* $Y = (Y_1, ..., Y_\ell)$ *and variables* $T_1, ..., T_\omega$ *is a list* $\mathcal{T} = \mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_\omega$ *where*

$$\begin{aligned} \mathcal{T}_1(Y) &\in \mathrm{D}[Y, T_1] \\ \mathcal{T}_2(Y) &\in \mathrm{D}[Y, T_1, T_2] \\ &\vdots \\ \mathcal{T}_\omega(Y) &\in \mathrm{D}[Y, T_1, ..., T_\omega]. \end{aligned}$$

Given a parametrized triangular system $\mathcal{T} = \mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_\omega$ with parameters $Y = (Y_1, ..., Y_\ell)$, a set of polynomials $\mathcal{P} \subset \mathrm{D}[Y, T_1, ..., T_i]$ and a point $y \in \mathrm{R}^\ell$ such that $\mathcal{T}(y)$ is zero-dimensional, we denote by $\mathrm{SIGN}(\mathcal{P}(y), \mathcal{T}(y))$ the list of sign conditions satisfied by $\mathcal{P}(y)$ at the zeros of $\mathcal{T}(y)$. We want to compute a quantifier free formula such that $\Phi(z)$ holds if and only if

$$\mathrm{SIGN}(\mathcal{P}(z), \mathcal{T}(z)) = \mathrm{SIGN}(\mathcal{P}(y), \mathcal{T}(y)). \qquad \square$$

Algorithm 14.4. **[Inverse Sign Determination]**

- **Structure:** an ordered domain D contained in a real closed field R.
- **Input:**
  - a parametrized triangular system of polynomials, $\mathcal{T}$ with parameters $Y = (Y_1, ..., Y_\ell)$,
  - a point $y \in \mathrm{R}^\ell$, specified by a Thom encoding, such that $\mathcal{T}(y)$ is zero-dimensional,
  - a subset $\mathcal{P} \subset \mathrm{D}[Y, T_1, ..., T_\omega]$.
- **Output:**
  - a family $\mathcal{A}(y) \subset \mathrm{D}[Y]$,
  - a quantifier free $\mathcal{A}(y)$-formula $\Phi(y)(Y)$ such that for any $z \in \mathrm{R}^\ell$, the formula $\Phi(y)(z)$ is true if and only if $\mathcal{T}(z)$ is zero-dimensional and

$$\mathrm{SIGN}(\mathcal{P}(y), \mathcal{T}(y)) = \mathrm{SIGN}(\mathcal{P}(z), \mathcal{T}(z)).$$

- **Complexity:** $s^{\ell+1}(d'^{\omega}d)^{O(\ell)}$, where s is a bound on the number of elements of $\mathcal{P}$ and d is a bound on the degrees of the polynomials in $\mathcal{T}$ and $\mathcal{P}$.
- **Procedure:**
  - Use Algorithm 12.19 (Triangular Sign Determination) to compute $\mathrm{SIGN}(\mathcal{Q}(y), \mathcal{T}(y))$. Form the list

$$B(\mathrm{SIGN}(\mathcal{Q}(y), \mathcal{T}(y))) \subset \{0, 1, 2\}^{\mathcal{Q}},$$

    using Remark 10.69 and its notation.

- Using Algorithm 12.18 (Parametrized Bounded Algebraic Sampling) with input $\mathcal{T}_1^1 + \cdots + \mathcal{T}_k^2$, output a finite set $\mathcal{U}$ of parametrized univariate representations.
- For every $\alpha \in B(\mathrm{SIGN}(\mathcal{Q}(y) \cup \mathrm{Der}(\mathcal{T}(y)), \mathcal{T}(y)))$ and every $u = (f, g_0, ..., g_k) \in \mathcal{U}$, compute the signed subresultant coefficients of f and $\mathcal{Q}_u^\alpha$, using Algorithm 8.21 (Signed subresultant) and place them in a set $\mathcal{A}(y) \subset \mathrm{D}[Y]$.
- Using Algorithm 13.1 (Computing realizable sign conditions), output the set $\mathrm{SIGN}(\mathcal{A}(y))$ of realizable sign conditions on $\mathcal{A}(y)$ and the subset $\Sigma(y)$ of $\mathrm{SIGN}(\mathcal{A}(y))$ of ρ such that for every z in the realization of ρ, the Tarski-queries of $f(z, T)$ and $\mathcal{Q}_u^\alpha(z, T)$ give rise to a list of non-empty sign conditions $\mathrm{SIGN}(\mathcal{P}(z), \mathcal{T}(z))$ that coincides with $\mathrm{SIGN}(\mathcal{P}(y), \mathcal{T}(y))$.
- Output $\mathcal{A}(y)$ and

$$\Phi(y)(Y) = \bigvee_{\sigma \in \Sigma(y)} \bigwedge_{Q \in \mathcal{A}(y)} \mathrm{sign}(Q(Y)) = \sigma(Q).$$

Proof of correctness: It follows from the correctness of Algorithm 12.19 (Triangular Sign Determination), Remark 10.69, Algorithm 12.18 (Parametrized Bounded Algebraic Sampling), Algorithm 8.21 (Signed subresultant) and Algorithm 13.1 (Computing realizable sign conditions). □

Complexity analysis: Suppose that the degree of f_i is bounded by d' and the degrees of all the polynomials in $\mathcal{P}$ are bounded by d, and that the number of polynomials in $\mathcal{P}$ is s. Using the complexity of Algorithm 12.19 (Triangular Sign Determination), the number of arithmetic operations in D in Step 1 is bounded by $s\ {d'}^{O(\omega)}$. The number of elements of $B(\mathrm{SIGN}(\mathcal{Q}(y), \mathcal{T}(y)))$ is bounded by $s\ O(d')^\omega d$, using Remark 10.69. The number of arithmetic operations in $\mathrm{D}[Y]$ is bounded by $s\, {d'}^{O(\omega)} d^{O(1)}$. The degree in Y in the intermediate computations is bounded by ${d'}^{O(\omega)} d^{O(1)}$, using the complexity of Algorithm 12.19 (Triangular Sign Determination). Using the complexity analyses of Algorithms 8.4 (Addition of multivariate polynomials), 8.5 (Multiplication of multivariate polynomials), and 8.6 (Exact division of multivariate polynomials), the number of arithmetic operations in D is bounded by $s(d'^\omega d)^{O(\ell)}$. The number of elements in $\mathcal{A}(y)$ is $s\ {d'}^{O(\omega)} d^{O(1)}$. Using the complexity of Algorithm 13.1 (Computing realizable sign conditions), the final complexity is $s^{\ell+1}(d'^\omega d)^{O(\ell)}$.

If $\mathrm{D} = \mathbb{Z}$, and the bitsizes of the coefficients of the polynomials are bounded by τ, then the bitsizes of the integers appearing in the intermediate computations and the output are bounded by $\tau(d'^\omega d)^{O(\ell)}$. □

We now describe our algorithm for the quantifier elimination problem. We make use of Algorithm 14.2 (Block Structured Signs) and Algorithm 14.4 (Inverse Sign Determination).

Let $\mathcal{P} \subset \mathrm{R}[X_1, ..., X_k, Y_1, ..., Y_\ell]$ be finite and let Π denote a partition of the list of variables $X = (X_1, ..., X_k)$ into blocks, $X_{[1]}, ..., X_{[\omega]}$, where the block $X_{[i]}$ is of size k_i, $1 \leq i \leq \omega$, $\sum_{1 \leq i \leq \omega} k_i = k$. We proceed in the same manner as the algorithm for the general decision problem, starting with the set $\mathcal{P}$ of polynomials and eliminating the blocks of variables to obtain a set of polynomials $\mathrm{B}_\Pi(\mathcal{P})$ in the variables Y. For a fixed $y \in \mathrm{R}^\ell$, the truth or falsity of the formula $\Phi(y)$ can be decided from the set $\mathrm{SIGN}_\Pi(\mathcal{P})(y)$. We next apply Algorithm 13.1 (Sampling) to the set of polynomials $\mathrm{B}_\Pi(\mathcal{P}) \subset D[Y]$, to obtain points in every semi-algebraically connected component of a realizable sign condition of $\mathrm{B}_\Pi(\mathcal{P})$. For each sample point y so obtained, we determine whether or not y satisfies the given formula using the set $\mathrm{SIGN}_\Pi(\mathcal{P})(y)$. If it does, then we use the Inverse Sign Determination Algorithm with the various $\mathcal{T}_u, \mathcal{P}_{u,\omega}, y$ as inputs to construct a formula $\Psi_y(Y)$. The only problem left is that this formula contains the infinitesimal quantities introduced by the general decision procedure. However we can replace each equality, or inequality in $\Psi_y(Y)$, by an equivalent larger formula without the infinitesimal quantities by using the ordering amongst the infinitesimal quantities. We output the disjunction of the formulas $\Psi_y(Y)$ constructed above.

We now give a more formal description of the algorithm and prove the bounds on the time complexity and the size of the output formula.

Algorithm 14.5. **[Quantifier Elimination]**

- **Structure:** an ordered domain D contained in a real closed field R.
- **Input:** a finite subset $\mathcal{P} \subset \mathrm{D}[X_1, ..., X_k, Y_1, ..., Y_\ell]$ of s polynomials of degree at most d, a partition Π of the list of variables $X = (X_1, ..., X_k)$ into blocks, $X_{[1]}, ..., X_{[\omega]}$, where the block $X_{[i]}$ is of size $k_i, 1 \leq i \leq \omega$, with $\sum_{1 \leq i \leq \omega} k_i = k$ and a $(\mathcal{P}, \Pi)$-formula $\Phi(Y)$.
- **Output:** a quantifier free formula $\Psi(Y)$ equivalent to $\Phi(Y)$.
- **Complexity:** $s^{(k_\omega+1)\cdots(k_1+1)(\ell+1)} d^{O(k_\omega)\cdots O(k_1)O(\ell)}$.
- **Procedure:**
  - Block Elimination Phase: Perform the Block Elimination Phase of Algorithm 14.2 (Block Structured Signs) on the set of polynomials $\mathcal{P}$, with $\omega + 1$ blocks of variables (Y, $X_{[1]}$, ..., $X_{[\omega]}$ to obtain the set $\mathcal{U}$ consisting of triangular systems $\mathcal{T}_u$ and the set of polynomials $\mathcal{P}_{u,\omega+1}$.
  - Formula Building Phase: For every $u = (u_1, ..., u_{\omega+1}) \in \mathcal{U}$ and every point y associated to u_1, compute $\mathrm{SIGN}(\mathcal{T}_u(y), \mathcal{P}_{u,\omega}(y))$, using Algorithm 12.19 (Triangular Sign Determination). Output the set $\mathrm{SIGN}_\Pi(\mathcal{P})(y)$ from the set $\{\mathrm{SIGN}(\mathcal{T}_u(y), \mathcal{P}_{u,\omega}(y)) \mid u \in \mathcal{U}\}$, and hence decide whether the formula $\Phi(y)$ is true.
  - If $\Phi(y)$ is true, apply Algorithm 14.4 (Inverse Sign Determination) with

$$\mathcal{T}_u, \mathcal{P}_{u,\omega}, y$$

as inputs to get the formulas $\Psi_{u,y}(Y)$. Let $\Psi_y(Y) = \bigwedge_u \Psi_{u,y}(Y)$, and let $\overline{\Psi(Y)} = \bigvee_y \Psi_y(Y)$, where the disjunction is over all the y for which $\Phi(y)$ is true in the previous step.

– Output $\Psi(Y) := \mathrm{Remo}_{\varepsilon_1,\delta_1,\ldots,\varepsilon_{\omega+1},\delta_{\omega+1}}(\overline{\Psi(Y)})$ (Notation 14.6).

Proof of correctness: The correctness of the algorithm follows from the correctness of Algorithm 14.3 ([General Decision), Algorithm 14.4 (Inverse Sign Determination), and Proposition 14.7. □

Complexity analysis: The elimination phase takes at most

$$s^{(k_\omega+1)\cdots(k_1+1)(\ell+1)}d^{O(k_\omega)\cdots O(k_1)O(\ell)}$$

arithmetic operations, and the number of sign conditions produced is also bounded by

$$s^{(k_\omega+1)\cdots(k_1+1)(\ell+1)}d^{O(k_\omega)\cdots O(k_1)O(\ell)}.$$

The degrees in the variables T_1, ..., T_ω, $T_{\omega+1}$, ε_1, δ_1, ..., $\varepsilon_{\omega+1}$, $\delta_{\omega+1}$ in the polynomials produced, are all bounded by $d^{O(k_\omega)\cdots O(k_1)O(\ell)}$.

Invoking the bound on the Algorithm 14.4 (Inverse Sign Determination), and the bound on the number of tuples produced in the elimination phase, which is $s^{(k_\omega+1)\cdots(k_1+1)\ell}d^{O(k_\omega)\cdots O(k_1)O(\ell)}$ we see that the formula building phase takes no more than

$$s^{(k_\omega+1)\cdots(k_1+1)\ell+\ell}d^{O(k_\omega)\cdots O(k_1)O(\ell)}$$

operations. Since the degrees of the variables $\varepsilon_{\omega+1}$, $\delta_{\omega+1}$, ..., ε_1, δ_1, are all bounded by $d^{O(k_\omega)\cdots O(k_1)O(\ell)}$, each atom is expanded to a formula of size at most $d^{(O(k_\omega)\cdots O(k_1)O(\ell)}$.

The bound on the size of the formula is an easy consequence of the bound on the number of tuples produced in the elimination phase, and the bound on the formula size produced by Algorithm 14.4 (Inverse Sign Determination).

If $\mathrm{D} = \mathbb{Z}$, and the bitsizes of the coefficients of the polynomials are bounded by τ, then the bitsizes of the integers appearing in the intermediate computations and the output are bounded by $\tau d^{O(k_\omega)\cdots O(k_1)O(\ell)}$. □

This proves the following result.

Theorem 14.16. [Quantifier Elimination] *Let $\mathcal{P}$ be a set of at most s polynomials each of degree at most d in $k+\ell$ variables with coefficients in a real closed field* R*, and let* Π *denote a partition of the list of variables $(X_1, ..., X_k)$ into blocks, $X_{[1]}, ..., X_{[\omega]}$, where the block $X_{[i]}$ has size k_i, for $1 \leq i \leq \omega$. Given $\Phi(Y)$, a $(\mathcal{P}, \Pi)$-formula, there exists an equivalent quantifier free formula,*

$$\Psi(Y) = \bigvee_{i=1}^{I} \bigwedge_{j=1}^{J_i} \Big(\bigvee_{n=1}^{Ni,j} \mathrm{sign}(P_{ijn}(Y)) = \sigma_{ijn} \Big),$$

where $P_{ijn}(Y)$ *are polynomials in the variables* Y, $\sigma_{ijn} \in \{0, 1, -1\}$,

$$\begin{aligned} I &\leq s^{(k_\omega+1)\cdots(k_1+1)(\ell+1)} d^{O(k_\omega)\cdots O(k_1)O(\ell)}, \\ J_i &\leq s^{(k_\omega+1)\cdots(k_1+1)} d^{O(k_\omega)\cdots O(k_1)}, \\ N_{ij} &\leq d^{O(k_\omega)\cdots O(k_1)}, \end{aligned}$$

and the degrees of the polynomials $P_{ijk}(y)$ *are bounded by* $d^{O(k_\omega)\cdots O(k_1)}$. *Moreover, there is an algorithm to compute* $\Psi(Y)$ *with complexity*

$$s^{(k_\omega+1)\cdots(k_1+1)(\ell+1)} d^{O(k_\omega)\cdots O(k_1)O(\ell)}$$

in D, *denoting by* D *the ring generated by the coefficients of* $\mathcal{P}$.

If $\mathrm{D} = \mathbb{Z}$, *and the bitsizes of the coefficients of the polynomials are bounded by* τ, *then the bitsizes of the integers appearing in the intermediate computations and the output are bounded by* $\tau d^{O(k_\omega)\cdots O(k_1)O(\ell)}$.

Remark 14.17. Note that, for most natural geometric properties that can be expressed by a formula in the language of ordered fields, the number of alternations of quantifiers in the formula is small (say at most five or six) while the number of variables can be arbitrarily big. A typical illustrative example is the formula describing the closure of a semi-algebraic set. In such situations, using Theorem 14.16, the complexity of quantifier elimination is singly exponential in the number of variables. □

Exercise 14.1. Design an algorithm computing the minimum value (maybe $-\infty$) of a polynomial of degree d defined on R^k with complexity $d^{O(k)}$. Make precise how this minimum value is described.

14.3 Local Quantifier Elimination

In this section we discuss a variant of Algorithm 14.5 (Quantifier Elimination) whose complexity is slightly better. A special feature of this algorithm is that the quantifier-free formula output will not necessarily be a disjunction of sign conditions, but will have a more complicated nested structure reflecting the logical structure of the input formula.

For this purpose, we need a parametrized version of Algorithm 12.20 (Triangular Thom Encoding). This algorithm will be based on Algorithm 14.6 (Parametrized Sign Determination).

Algorithm 14.6. **[Parametrized Sign Determination]**

- **Structure:** an ordered domain D contained in a real closed field R.
- **Input**: a parametrized triangular system $\mathcal{T}$ with parameters $(Y_1, ..., Y_\ell)$, and variables $(X_1, ..., X_k)$ and a finite set $\mathcal{Q} \subset \mathrm{D}[Y_1, ..., Y_\ell, X_1, ..., X_k]$.
- **Output**:
  - a finite set $\mathcal{A} \subset \mathrm{D}[Y]$, with $Y = (Y_1, ..., Y_k)$.

- for every $\rho \in \mathrm{SIGN}(\mathcal{A})$, a list $\mathrm{SIGN}(\mathcal{Q}, \mathcal{T})(\rho)$ of sign conditions on $\mathcal{Q}$ such that, for every y in the realization $\mathrm{Reali}(\rho)$ of ρ, $\mathrm{SIGN}(\mathcal{Q}, \mathcal{T})(\rho)$ is the list of sign conditions realized by $\mathcal{Q}(y)$ on the zero set $Z(y)$ of $\mathcal{T}(y)$.

- **Complexity**:] $s^{\ell(\ell+1)+1} {d'}^{O(k\ell)} d^{O(\ell)}$, where s is a bound on the number of polynomials in $\mathcal{Q}$ and d is a bound on the degrees of the polynomials in $\mathcal{T}$ and $\mathcal{Q}$.
- **Procedure**:
  - Step 1: Perform Algorithm 12.18 (Parametrized Bounded Algebraic Sampling) with input $\mathcal{T}_1^2 + ..., + \mathcal{T}_k^2$, for $\mathcal{T}_i \in \mathcal{T}$ and output $\mathcal{U}$.
  - Step 2: Consider for every $u = (f, g_0, ..., g_k) \in \mathcal{U}$ and every $Q \in \mathcal{Q}$ the finite set $\mathcal{F}_{u,Q}$ containing Q_u (Notation 13.8) and all the derivatives of f with respect to T, and compute

$$\mathcal{D}_{u,Q} = \mathrm{RElim}_T(f, \mathcal{F}_{u,Q}) \subset \mathrm{D}[Y],$$

    using Algorithm 11.19 (Restricted Elimination).
  - Step 3: Define $\mathcal{D} = \bigcup_{u \in \mathcal{U}, Q \in \mathcal{Q}} \mathcal{D}_{u,Q}$. Perform Algorithm 13.1 (Sampling) with input $\mathcal{D}$. Denote by $\mathcal{S}$ the set of sample points output.
  - Step 4: For every sample point y, perform Algorithm 14.4 (Inverse Sign Determination) and output the set $\mathcal{A}(y) \subset \mathrm{D}[Y]$, as well as $\mathrm{SIGN}(\mathcal{Q}(y), \mathcal{T}(y))$ and $\Phi(y)(Y)$.
  - Step 5: Define $\mathcal{A} = \mathcal{D} \cup \bigcup_{y \in \mathcal{S}} \mathcal{A}(y)$. Compute the set of realizable sign conditions on $\mathcal{A}$ using Algorithm 13.1 (Sampling).
  - Step 6: For every $\rho \in \mathrm{SIGN}(\mathcal{A})$ denote by y the sample point of $\mathrm{Reali}(\rho)$. Define $\mathrm{SIGN}(\mathcal{Q}, \mathcal{T})(\rho)$ as $\mathrm{SIGN}(\mathcal{Q}(y), \mathcal{T}(y))$, computed by Algorithm 12.19 (Triangular Sign Determination).

Proof of correctness: Follows from the correctness of Algorithm 12.18 (Parametrized Bounded Algebraic Sampling), Algorithm 11.19 (Restricted Elimination), Algorithm 13.1 (Sampling), Algorithm 14.4 (Inverse Sign Determination), Algorithm 13.1 (Sampling) and Algorithm 12.19 (Triangular Sign Determination). □

Complexity analysis: We estimate the complexity in terms of the number of parameters ℓ, the number of variables k, the number s of polynomials in $\mathcal{P}$, a bound d' on the degrees of the polynomials in $\mathcal{T}$ and a bound d on the degrees of the polynomials in $\mathcal{P}$.

- Step 1: Using the complexity analysis of Algorithm 12.18 (Parametrized Bounded Algebraic Sampling), the complexity of this step is ${d'}^{O(k)}$ in the ring $\mathrm{D}[Y]$. The polynomials output are of degree $O(d')^k$ in T and of degrees ${d'}^{O(k)}$ in Y. Finally, the complexity is ${d'}^{O(k\ell)}$ in the ring D. The number of elements of $\mathcal{U}$ is $O(d')^k$.
- Step 2: The complexity of this step is $s\, {d'}^{O(k\ell)} d^{O(\ell)}$, using the complexity analysis of Algorithm 11.19 (Restricted Elimination). The number of polynomials output is $s\, {d'}^{O(k)} d^{O(1)}$.

- Step 3: The complexity of this step is $s^{\ell} {d'}^{O(k\ell)} d^{O(1)}$, using the complexity analysis of Algorithm 13.1 (Sampling). There are $s^{\ell} {d'}^{O(k\ell)} d^{O(\ell)}$ points output.
- Step 4: For each sample point, the complexity is $s^{\ell+1} {d'}^{O(k\ell)} d^{O(\ell)}$ using the complexity analysis of Algorithm 14.4 (Inverse Sign Determination). So the complexity of this step is $s^{2\ell+1} {d'}^{O(k\ell)} d^{O(\ell)}$. The number of elements of $\mathcal{A}(y)$ is bounded by $s\, {d'}^{O(k)} d^{O(1)}$ and the degrees of the elements of $\mathcal{A}(y)$ are bounded by ${d'}^{O(k)} d^{O(1)}$.
- Step 5: The number of elements in $\mathcal{A}$ is $s^{\ell+1} {d'}^{O(k\ell)} d^{O(\ell)}$, and the degrees of the elements of $\mathcal{A}$ are bounded by ${d'}^{O(k)} d^{O(1)}$. The complexity of this step is $s^{\ell(\ell+1)} {d'}^{O(k\ell)} d^{O(\ell)}$, using the complexity analysis of Algorithm 13.1 (Sampling).
- Step 6: For every ρ, the complexity is $s\, {d'}^{O(k\ell)} d^{O(\ell)}$. So the complexity of this step is $s^{\ell(\ell+1)+1} {d'}^{O(k\ell)} d^{O(\ell)}$ using the complexity analysis of Algorithm 12.19 (Triangular Sign Determination).

Finally the complexity is $s^{\ell(\ell+1)+1} {d'}^{O(k\ell)} d^{O(\ell)}$.

If $\mathrm{D} = \mathbb{Z}$, and the bitsizes of the coefficients of the polynomials are bounded by τ, then the bitsizes of the integers appearing in the intermediate computations and the output are bounded by $\tau {d'}^{O(k\ell)} d^{O(\ell)}$. □

We now define parametrized triangular Thom encodings.

A **parametrized triangular Thom encoding** of level k with parameters $Y = (Y_1, ..., Y_\ell)$ specified by $\mathcal{A}, \rho, \mathcal{T}, \sigma$ is

- a finite subset $\mathcal{A}$ of $\mathrm{R}[Y]$,
- a sign condition ρ on $\mathcal{A}$,
- a triangular system of polynomials $\mathcal{T}$, where $\mathcal{T}_i \in \mathrm{R}[Y, X_1, ..., X_i]$,
- a sign condition σ on $\mathrm{Der}(\mathcal{T})$ such that for every $y \in \mathrm{Reali}(\rho)$, there is a zero $z(y)$ of $\mathcal{T}(y)$ with triangular Thom encoding ρ.

Algorithm 14.7. **[Parametrized Triangular Thom Encoding]**

- **Structure**: an ordered integral domain D contained in a real closed field R.
- **Input**: a parametrized triangular system $\mathcal{T}$ with parameters $(Y_1, ..., Y_\ell)$ and variables $(X_1, ..., X_k)$.
- **Output**:
  - a finite set $\mathcal{A} \subset \mathrm{D}[Y]$, with $Y = (Y_1, ..., Y_k)$,
  - for every $\rho \in \mathrm{SIGN}(\mathcal{A})$, a list of sign conditions on $\mathrm{Der}(\mathcal{T})$ specifying for every $y \in \mathrm{Reali}(\rho)$, the list of triangular Thom encodings of the roots of $\mathcal{T}(y)$.
- **Complexity**: ${d'}^{O(k\ell)}$ where d' is a bound on the degrees of the polynomials in $\mathcal{T}$.
- **Procedure:** Apply Algorithm 14.6 (Parametrized Sign Determination) to $\mathcal{T}$ and $\mathrm{Der}(\mathcal{T})$.

Proof of correctness: Immediate. □

Complexity analysis: The complexity is $d'^{O(k\ell)}$, using the complexity of Algorithm 14.6 (Parametrized Sign Determination). The number of elements in $\mathcal{A}$ is $d'^{O(k\ell)}$, and the degrees of the elements of $\mathcal{A}$ are bounded by $d'^{O(k)}$. □

We follow the notations introduced in the last two sections.

Let $\mathcal{P} \subset \mathrm{R}[X_1, ..., X_k, Y_1, ..., Y_\ell]$ be finite and let Π denote a partition of the list of variables $X = (X_1, ..., X_k)$ into blocks, $X_{[1]}, ..., X_{[\omega]}$, where the block $X_{[i]}$ is of size $k_i, 1 \leq i \leq \omega$, $\sum_{1 \leq i \leq \omega} k_i = k$.

Recall that (Notations 14.8 and 14.10) for every $i = \omega - 1, ..., 0$, the elements of $\mathrm{UR}_{\Pi,i}(\mathcal{P})$, are parametrized univariate representations in the variable T_{i+1}, contained in $\mathrm{D}[Y, X_{[1]}, ..., X_{[i]}, \varepsilon_{i+1}, \delta_{i+1}][T_{i+1}]^{k_{i+1}+2}$. Let

$$u = (u_0, ..., u_{\omega-1}) \in \mathcal{U} = \prod_{i=0}^{\omega-1} \mathrm{UR}_{\Pi,i}(\mathcal{P}),$$

with

$$u_{i-1} = (f^{[i]}, g_0^{[i]}, g_1^{[i]}, ..., g_{k_i}^{[i]}).$$

Also recall that we denote,

$$\begin{aligned} \mathcal{T}_{u,i} &= (f^{[1]}(T_1), f_{u,1}^{[2]}(T_1, T_2), \cdots, f_{u,i-1}^{[i]}(T_1, T_2, ..., T_i)), \\ \mathcal{T}_u &= (f^{[1]}(T_1), f_{u,1}^{[2]}(T_1, T_2), \cdots, f_{u,\omega-1}^{[\omega]}(T_1, T_2, ..., T_\omega)). \end{aligned}$$

We now introduce the following notation which is used in the description of the algorithm below.

Notation 14.18. Let $u = (u_0, ..., u_{j-1}) \in \mathcal{U}_i = \prod_{j=0}^{i-1} \mathrm{UR}_{\Pi,j}(\mathcal{P})$. We denote by $\mathcal{L}_{u,i}$ the set of all possible triangular Thom encodings of roots of $\mathcal{T}_{u,i}$ as y vary over $\mathrm{R}\langle \varepsilon_1, \delta_1, ..., \varepsilon_\omega, \delta_\omega \rangle^\ell$. □

Algorithm 14.8. **[Local Quantifier Elimination]**

- **Structure:** an ordered domain D contained in a real closed field R.
- **Input:** a finite subset $\mathcal{P} \subset \mathrm{R}[X_1, ..., X_k, Y_1, ..., Y_\ell]$, a partition Π of the list of variables $X = (X_1, ..., X_k)$ into blocks, $X_{[1]}, ..., X_{[\omega]}$ and a $(\mathcal{P}, \Pi)$-formula $\Phi(Y)$.
- **Output:** a quantifier free formula, $\Psi(Y)$, equivalent to $\Phi(Y)$.
- **Complexity:** $s^{(k_\omega+1)...(k_1+1)} d^{\ell O(k_\omega)...O(k_1)}$ where s is a bound on the number of elements of $\mathcal{P}$, d is a bound on the degree of elements of $\mathcal{P}$, and k_i is the size of the block $X_{[i]}$.
- **Procedure:**
  - Initialize $\mathrm{B}_{Pi,\omega}(\mathcal{P}) := \mathcal{P}$.
  - Block Elimination Phase: Compute

$$\mathrm{B}_{\Pi,i}(\mathcal{P}) = \mathrm{BElim}_{X_{[i+1]}}(\mathrm{Bor}_{\Pi,i+1}(\mathcal{P})),$$

for $1 \leq i \leq \omega - 1$, applying repeatedly $\mathrm{BElim}_{X_{[i+1]}}$, using Algorithm 14.1 (Block Elimination).

Compute $\mathrm{UR}_{\Pi,i}(\mathcal{P})$, for every $i = \omega - 1, ..., 0$. The elements of $\mathrm{UR}_{\Pi,i}(\mathcal{P})$ are parametrized univariate representations in the variable T_{i+1}, contained in

$$\mathrm{D}[Y, X_{[1]}, ..., X_{[i]}, \varepsilon_{i+1}, \delta_{i+1}][T_{i+1}]^{k_{i+1}+2}.$$

– For every

$$u = (u_0, ..., u_{\omega-1}) \in \mathcal{U} = \prod_{i=0}^{\omega-1} \mathrm{UR}_{\Pi,i}(\mathcal{P}),$$

with

$$u_{i-1} = (f^{[i]}, g_0^{[i]}, g_1^{[i]}, ..., g_{k_i}^{[i]}),$$

compute the corresponding triangular system,

$$\mathcal{T}_u = (f^{[1]}(Y, T_1), f_{u,1}^{[2]}(Y, T_1, T_2), ..., f_{u,\omega-1}^{[\omega]}(Y, T_1, T_2, ..., T_\omega)).$$

(see Notation 14.10).

For $i = 0...\omega - 1$ compute the sets $\mathcal{L}_{u,i}$, using Algorithm 14.7 (Parametrized Triangular Thom Encoding) with input $\mathcal{T}_{u,i}$.

– Let

$$\Phi(Y) = (\mathrm{Qu}_1 X_{[1]})...(\mathrm{Qu}_\omega X_{[\omega]}) F(X, Y)$$

where $\mathrm{Qu}_i \in \{\forall, \exists\}$, $Y = (Y_1, ..., Y_\ell)$ and $F(X, Y)$ is a quantifier free $\mathcal{P}$-formula.

For every atom of the form $\mathrm{sign}(P) = \sigma, P \in \mathcal{P}$ occurring in the input formula F, and for every

$$u = (u_0, ..., u_{\omega-1}) \in \mathcal{U} = \prod_{i=0}^{\omega-1} \mathrm{UR}_{\Pi,i}(\mathcal{P}),$$

with

$$u_{i-1} = (f^{[i]}, g_0^{[i]}, g_1^{[i]}, ..., g_{k_i}^{[i]}),$$

and $\tau \in \mathcal{L}_{u,\omega}$ compute using Algorithm 14.5 (Quantifier Elimination) a quantifier-free formula $\phi_{u,\tau}$ equivalent to the formula

$$(\exists T_1, ..., T_\omega) \bigwedge_{i=1}^{\omega} \mathrm{SIGN}(\mathrm{Der}(f_{u,i-1}^{[i]})) = \tau_i \bigwedge \mathrm{SIGN}(P_{u,\omega}) = \sigma.$$

Let $F_{u,\tau}$ denote the quantifier-free formula obtained by replacing every atom ϕ in F by the corresponding formula $\phi_{u,\tau}$.

Also, for every

$$u = (u_0, ..., u_{\omega-1}) \in \mathcal{U} = \prod_{i=0}^{\omega-1} \mathrm{UR}_{\Pi,i}(\mathcal{P})),$$

with

$$u_{i-1} = (f^{[i]}, g_0^{[i]}, g_1^{[i]}, ..., g_{k_i}^{[i]}), \tau \in \mathcal{L}_{u,\omega}$$

and for every $j, 1 \leq j \leq \omega$, compute using Algorithm 14.5 (Quantifier Elimination) a quantifier-free formula $\psi_{u,\tau,j}$ equivalent to the formula,

$$(\exists T_1, ..., T_j) \bigwedge_{i=1}^{j} \text{SIGN}(\text{Der}(f_{u,i-1}^{[i]})) = \tau_i.$$

– For $u \in \mathcal{U}$ and $\tau \in \mathcal{L}_{u,\omega}$, let

$$\Phi_{\omega,u,\tau} = F_{u,\tau}.$$

Compute inductively for i from $\omega - 1$ to 0, and for every

$$u = (u_0, ..., u_{i-1}) \in \mathcal{U}_i = \prod_{j=0}^{i-1} \text{UR}_{\Pi,j}(\mathcal{P}),$$

and $\tau \in \mathcal{L}_{u,i}$,

$$\begin{aligned} \Phi_{i,u,\tau} &= \bigwedge_{(v,\rho),\bar{v}=u,\bar{\rho}=\tau} (\psi_{v,\rho,i+1} \wedge \Phi_{i+1,v,\rho}) \text{ if } \text{Qu}_{i+1} = \exists, \\ &= \bigwedge_{(v,\rho),\bar{v}=u,\bar{\rho}=\tau} (\psi_{v,\rho,i+1} \Longrightarrow \Phi_{i+1,v,\rho}) \text{ if } \text{Qu}_{i+1} = \forall. \end{aligned}$$

Take $\overline{\Phi(Y)} = \Phi_0$.

– Output $\Psi(Y) = \text{Remo}_{\varepsilon_1,\delta_1,...,\varepsilon_\omega,\delta_\omega}(\overline{\Phi(Y)})$ (Notation 14.6).

Complexity analysis: It follows from the complexity analysis of Algorithm a14.1 (Block Elimination), Algorithm 14.7 (Parametrized Triangular Thom Encoding) and Algorithm 14.5 (Quantifier Elimination) that the complexity is bounded by $s^{(k_\omega+1)\cdots(k_1+1)} d^{\ell O(k_\omega)\cdots O(k_1)}$.

If $\text{D} = \mathbb{Z}$, and the bitsizes of the coefficients of the polynomials are bounded by τ, then the bitsizes of the integers appearing in the intermediate computations and the output are bounded by $\tau d^{O(k_\omega)\cdots O(k_1)}$.

Note that the only improvement compared to Algorithm 14.5 (Quantifier Elimination) is that the exponent of s does not depend on the number of free variables ℓ. Note also that the total number of polynomials in Y appearing in the formula is $s^{(k_\omega+1)\cdots(k_1+1)} d^{\ell O(k_\omega)\cdots O(k_1)}$. Determining which are the realizable sign conditions on these polynomials would cost $s^{(\ell+1)(k_\omega+1)\cdots(k_1+1)} d^{\ell O(k_\omega)\cdots O(k_1)}$, but this computation is not part of the algorithm. □

We now give an application of Algorithm (Local Quantifier Elimination) 14.8 to the closure of a semi-algebraic set.

Let S be a semi-algebraic set described by a quantifier free $\mathcal{P}$-formula $\text{F}(X)$, where $\mathcal{P}$ is a finite set of s polynomials of degree at most d in k variables. The closure of S is described by the following quantified formula $\Psi(X)$

$$\forall\, Z \quad \exists\, Y \quad \|X - Y\|^2 < Z^2 \wedge F(Y).$$

Note that $\Psi(X)$ is a first-order formula with two blocks of quantifiers, the first with one variable and the second one with k variables. Denote by $\mathcal{R}$ the set of polynomials in k variables obtained after applying twice Algorithm 14.1 (Block Elimination) to the polynomials appearing in the formula describing the closure of S in order to eliminate Z and Y. These polynomials have the property that the closure of S is the union of semi-algebraically connected components of sets defined by sign conditions over $\mathcal{R}$. According to Theorem 14.16 the set $\mathcal{R}$ has $s^{2k+1}d^{O(k)}$ polynomials and each of these polynomials has degree at most $d^{O(k)}$.The complexity for computing $\mathcal{R}$ is $s^{2(k+1)}d^{O(k)}$. Note that we cannot ensure that the closure of S is described by polynomials in $\mathcal{R}$. However, performing Algorithm 14.8 (Local Quantifier Elimination) gives a quantifier-free description of the closure of S in time $s^{2(k+1)}d^{O(k)}$ by $s^{2k+1}d^{O(k)}$ polynomials of degree at most $d^{O(k)}$.

14.4 Global Optimization

We describe an algorithm for finding the infimum of a polynomial on a semi-algebraic set as well as a minimizer if there exists one.

Algorithm 14.9. **[Global Optimization]**

- **Structure:** an ordered domain D contained in a real closed field R.
- **Input**: a finite subset $\mathcal{P} \subset \mathrm{D}[X_1, ..., X_k]$, a $\mathcal{P}$-semi-algebraic set S described by a quantifier free formula $\Phi(X)$ and $F \in \mathrm{D}[X_1, ..., X_k]$.
- **Output:** the infimum w of F on S, and a minimizer, i.e. a point $x \in S$ such that $F(x) = w$ if such a point exists.
- **Complexity:** $s^{2k+1}\, d^{O(k)}$, where s is a bound on the number of elements of $\mathcal{P}$ and d is a bound on degree of F and of the elements of $\mathcal{P}$.
- **Procedure:**
  - Let Y be a new variable and $G = Y - F \in \mathrm{D}[Y, X_1, ..., X_k]$. Denote by $S' \subset R^{k+1}$ the realization of $\Phi \wedge G = 0$.
  - Call Algorithm 14.1 (Block Elimination) with block of variables $X_1, ...,$ X_k and set of polynomials $\mathcal{P} \cup \{G\} \subset \mathrm{D}[Y, X_1, ..., X_k]$. Let $\mathcal{B} \subset \mathrm{D}[Y]$ denote $\mathrm{BElim}_X(\mathcal{P} \cup \{G\})$.
  - Call Algorithm 10.19 (Univariate Sample Points) with input $\mathcal{B}$ and denote by $\mathcal{C}$ the set of sample points so obtained. Each element of $\mathcal{C}$ is a Thom encoding (h, σ).
  - Fore each $y = (h, \sigma) \in \mathcal{C}$, the points associated to $\mathrm{UR}_X(\mathcal{P} \cup \{\mathcal{G}\})(y)$ intersect every semi-algebraically connected component of every realizable sign condition of the set $\mathcal{P} \cup \{\mathcal{G}\}(y)$ in $\mathrm{R}\langle\varepsilon, \delta\rangle^k$. Compute the subset $\mathcal{C}'$ of elements $y \in \mathcal{C}$ such that the set of ˆpoints associated to $\mathrm{UR}_X(\mathcal{P} \cup \{\mathcal{G}\})(y)$ meets the extension of S' to $\mathrm{R}\langle\varepsilon, \delta\rangle$ using Algorithm 12.19 (Triangular Sign Determination).

- If there is no root y of a polynomial in $\mathcal{B}$ such that for all $y' \in \mathcal{C}'$, $y' \geqslant y$ holds, define w as $-\infty$. Otherwise, define w as the maximum $y \in \mathcal{C}$ which is a root of a polynomial in $\mathcal{B}$ and such that for all $y' \in \mathcal{C}'$, $y' \geqslant y$ holds.
- If $w = (h, \sigma) \in \mathcal{C}'$, pick $u = (f, g_0, ..., g_k) \in \mathrm{UR}_X(\mathcal{P} \cup \{\mathcal{G}\})(w)$ with associated point in the extension of S' to $\mathrm{R}\langle\varepsilon, \delta\rangle$. Replace δ and ε by appropriately small elements from the field of quotients of D using Algorithm 11.20 (Removal of Infinitesimals) with input f, its derivatives and the P_u, $P \in \mathcal{P}$ and using Remark 11.27. Then clear denominators to obtain univariate representation with entries in $\mathrm{D}[T]$.

Proof of correctness: Follows clearly from the correctness of Algorithm 14.1 (Block Elimination). □

Complexity analysis: The call to Algorithm 14.1 (Block Elimination) costs $s^{k+1}\, d^{O(k)}$. The call to Algorithm 10.19 (Univariate Sample Points) costs $s^{2k}\, d^{O(k)}$ since there are at most $s^k\, d^{O(k)}$ polynomials of degree at most $d^{O(k)}$. Each call to Algorithm 12.19 (Triangular Sign Determination) costs $s\, d^{O(k)}$ and there are $s^{2k} d^{O(k)}$ such calls. The call to Algorithm 11.20 (Removal of Infinitesimals) costs $(s + d^{O(k)})\, d^{O(k)}$ which is $s\, d^{O(k)}$. The total complexity is thus $s^{2k+1}\, d^{O(k)}$. □

14.5 Dimension of Semi-algebraic Sets

Let S be a semi-algebraic set described by a quantifier free $\mathcal{P}$-formula $\Phi(X)$

$$S = \{x \in \mathrm{R}^k \mid \Phi(x)\}$$

where $\mathcal{P}$ is a finite set of s polynomials in k variables with coefficients in a real closed field R. We denote by $\mathrm{SSIGN}(\mathcal{P})$ the set of **strict realizable sign conditions of** $\mathcal{P}$, i.e. the realizable sign conditions $\sigma \in \{0, 1, -1\}^{\mathcal{P}}$ such that for every $P \in \mathcal{P}$, $P \neq 0$, $\sigma(P) \neq 0$.

Proposition 14.19. *The dimension of S is k if and only is there exists $\sigma \in \mathrm{SSIGN}(\mathcal{P})$ such that* $\mathrm{Reali}(\sigma) \subset S$.

Proof: The dimension of S is k if and only if there exists a point $x \in S$ and $r > 0$ such that $B(x, r) \subset S$. The sign condition satisfied by $\mathcal{P}$ at such an x is necessarily strict. In the other direction, if the sign condition σ satisfied by $\mathcal{P}$ at such an x is strict, $\mathrm{Reali}(\sigma)$ is open, and contained in S since S is defined by a quantifier free $\mathcal{P}$-formula. □

It is reasonable to expect that the dimension of S is $\geq j$ if and only if the dimension of $\pi(S)$ is j, where π is a linear surjection of R^k to R^j.

Using results from Chapter 13, we are going to prove that using $j(k-j)+1$ well chosen linear surjections is enough. Recall that we have defined in Notation 13.26 a family

$$\mathcal{L}_{k,k-j} = \{V_i \mid 0 \leq i \leq j(k-j)\}.$$

of $j\ (k-j)+1$ vector spaces such that any linear subspace T of R^k of dimension $k' \geq j$ is such that there exists $0 \leq i \leq j(k-j)$ such that V_i and T span R^k (see Corollary 13.28). We denoted by $v_k(x)$ the Vandermonde vector

$$(1, x, ..., x^{k-1}).$$

and by V_ℓ the vector subspace of R^k generated by

$$v_k(\ell), v_k(\ell+1), ..., v_k(\ell+k-k'-1).$$

We also defined in Notation 13.26 a linear bijection $L_{j,i}$ such that $L_{j,i}(V_i)$ consists of vectors of R^k having their last j coordinates equal to 0. We denoted by $M_{k',\ell} = (d_{k-k',\ell})^{k'} L^{-1}_{k',\ell}$, with

$$d_{k-k',\ell} = \det(v_{k-k'}(\ell), ..., v_{k-k'}(\ell+k-k'-1)),$$

and remarked that $M_{k',\ell}$ plays the same role as the inverse of $L_{k',\ell}$ but is with integer coordinates.

We denote by π_j the canonical projection of R^k to R^j forgetting the first $k-j$ coordinates.

Proposition 14.20. *Let $0 \leq j \leq k$. The dimension of S is $\geq j$ if and only if there exists $0 \leq i \leq j(k-j)$ such that the dimension of $\pi_j(L_{j,i}(S))$ is j.*

Proof: It is clear that if the dimension of $\pi_j(L_{j,i}(S))$ is j, the dimension of S is $\geq j$. In the other direction, if the dimension of S is $k' \geq j$, by Proposition 5.53, there exists a smooth point x of S of dimension k' with tangent space denoted by T. By Corollary 13.28, there exists $0 \leq i \leq j(k-j)$, such that V_i and T span R^k. Since $L_{j,i}(V_i)$ consists of vectors of R^k having their last j coordinates equal to 0, and $L_{j,i}(V_i)$ and $L_{j,i}(T)$ span R^k, $\pi_j(L_{j,i}(T))$ is R^j. Then the dimension of $\pi_j(L_{j,i}(S))$ is j. □

The idea for computing the dimension is simple: check whether the dimension of S is k or -1 (i.e. is empty) using Proposition 14.19. If it is not the case, try $k-1$ or 0 or, then $k-2$ or 1, etc.

Algorithm 14.10. **[Dimension]**

- **Structure:** an ordered domain D contained in a real closed field R.
- **Input**: a finite subset $\mathcal{P} \subset \mathrm{D}[X_1, ..., X_k]$, and a semi-algebraic set S described by a quantifier free $\mathcal{P}$-formula $\Phi(X)$.

- **Output:** the dimension k' of S.
- **Complexity:**

$$\begin{cases} s^{(k-k')k'}d^{O(k'(k-k'))} & \text{if } k' \geq k/2 \\ s^{(k-k'+1)(k'+1)}d^{O(k'(k-k'))} & \text{if } k' < k/2. \end{cases}$$

  where s is a bound on the number of elements of $\mathcal{P}$ and d is a bound on their degree.
- **Procedure:**
  - Initialize $j := 0$.
  - ($\star$) Consider the block structure Π_{k-j} with two blocks of variables: $X_{j+1}, ..., X_k$ and $X_1, ..., X_j$.
  - For every $i = 0, ..., j(k-j)$ let $\mathcal{P}_{k-j,i} = \mathcal{P}(M_{k-j,i})$, using Notation 13.26 and

$$S_{k-j,i} = \{x \in \mathrm{R}^k \mid \Phi(M_{k-j,i}(x))\}.$$

    - Compute $\mathrm{SIGN}_{\Pi_{k-j}}(\mathcal{P}_{k-j,i})$ using Algorithm 14.2 (Block Structured Signs).
    - Defining $X_{\leq j} = X_1..., X_j$, compute

$$\mathrm{SSIGN}(\mathrm{BElim}_{X_{\leq j}}(\mathcal{P}_{k-j,i}))$$

      using Algorithm 13.2 (Sampling). Note, using Remark 14.12, that every sample point output by Algorithm 14.2 (Block Structured Signs) is above a sample point for $\mathrm{BElim}_{X_{\leq j}}(\mathcal{P}_{k-j,i})$ output by Algorithm 13.2 (Sampling).
    - Check whether one of the strict sign conditions in

$$\mathrm{SSIGN}(\mathrm{BElim}_{X_{\leq j}}(\mathcal{P}_{k-j,i}))$$

      is satisfied at some point of $\pi_{k-j}(S_{k-j,i})$.
    - If one of the strict sign conditions in

$$\mathrm{SSIGN}(\mathrm{BElim}_{X_{\leq j}}(\mathcal{P}_{k-j,i}))$$

      is satisfied at some point of $\pi_{k-j}(S_{k-j,i})$, output $k-j$.
  - Consider the block structure Π_j with two blocks of variables: $X_{k-j+1}, ..., X_k$ and $X_1, ..., X_{k-j}$.
  - For every $i = 0, ..., j(k-j)$ let $\mathcal{P}_{j,i} = \mathcal{P}(M_{j,i})$, using Notation 13.30 and

$$S_{j,i} = \{x \in \mathrm{R}^k \mid \Phi(M_{j,i}(x))\}.$$

    - Compute $\mathrm{SIGN}_{\Pi_j}(\mathcal{P}_{j,i})$ using Algorithm 14.2 (Block Structured Signs).
    - Defining $X_{\leq k-j} = X_1..., X_{k-j}$, compute

$$\mathrm{SSIGN}(\mathrm{BElim}_{X_{\leq k-j}}(\mathcal{P}_{j,i}))$$

using Algorithm 13.2 (Sampling). Note, using Remark 14.12, that every sample point output by Algorithm 14.2 (Block Structured Signs) is above a sample point for $\mathrm{BElim}_{X_{\leq k-j}}(\mathcal{P}_{j,i})$ output by Algorithm 13.2 (Sampling).

- Check whether one of the strict sign conditions in

$$\mathrm{SSIGN}(\mathrm{BElim}_{X_{\leq k-j}}(\mathcal{P}_{j,i}))$$

is satisfied at some point of $\pi_j(S_{j,i})$.

- If for every $i = 0 \ldots j(k-j)$ none of the strict sign conditions in

$$\mathrm{SSIGN}(\mathrm{BElim}_{X_{\leq k-j}}(\mathcal{P}_{j,i}))$$

is satisfied at some point of $\pi_j(S_{j,i})$, output $j-1$.
- Otherwise define $j := j+1$ and go to $(\star)$.

Proof of correctness: Follows clearly from Proposition 14.19, Proposition 14.20, the correctness of of Algorithm 14.1 (Block Elimination), Algorithm 13.2 (Sampling). □

Complexity analysis: There are at most $(k+1)/2$ values of j considered in the algorithm.

For a given j, the complexity of the call to Algorithm 14.2 (Block Structured Signs) performed is $s^{(j+1)(k-j+1)}d^{O(j(k-j))}$, using the complexity analysis of Algorithm 14.2 (Block Structured Signs).

The call to Algorithm 13.2 (Sampling) for $\mathrm{BElim}_{X_{\leq j}}(\mathcal{P}_{k-j,i})$, has complexity $s^{(j+1)(k-j+1)}d^{O(j(k-j))}$, using the complexity analysis of Algorithm 14.1 (Block elimination) and 13.2 (Sampling), since the number of polynomials is $s^{j+1}d^{O(j)}$, their degrees are $d^{O(j)}$ and their number of variables is $k-j$.

Similarly, the call to Algorithm 13.2 (Sampling) for $\mathrm{BElim}_{X_{\leq k-j}}(\mathcal{P}_{j,i})$, has complexity $s^{(j+1)(k-j+1)}d^{O(j(k-j))}$, using the complexity analysis of Algorithm 14.1 (Block elimination) and 13.2 (Sampling), since the number of polynomials is $s^{k-j+1}d^{O(k-j)}$, their degrees are $d^{O(k-j)}$ and their number of variables is j.

Finally the total cost of the algorithm is

$$\begin{cases} s^{(k-k')k'}d^{O(k'(k-k'))} & \text{if } k' \geq k/2 \\ s^{(k-k'+1)(k'+1)}d^{O(k'(k-k'))} & \text{if } k' < k/2. \end{cases}$$

If $\mathrm{D} = \mathbb{Z}$, and the bitsizes of the coefficients of the polynomials are bounded by τ, then the bitsizes of the integers appearing in the intermediate computations and the output are bounded by $\tau d^{O(k'(k-k'))}$.

Note that this complexity result is output sensitive, which means that the complexity depends on the output of the algorithm. □

14.6 Bibliographical Notes

The idea of designing algorithms taking into account the block structure is due to Grigor'ev [76], who achieved doubly exponential complexity in the number of blocks for the general decision problem. It should be noted that for a fixed value of ω, this is only singly exponential in the number of variables. Heintz, Roy and Solerno [85] and Renegar [133] extended this result to quantifier elimination. Renegar's [133] algorithms solved the general decision problem in time $(s\ d)^{O(k_\omega)\cdots O(k_1)}$, and the quantifier elimination problem in time $(s\,d)^{O(k_\omega)\cdots O(k_1)O(\ell)}$.

Most of the results presented in this chapter are based on [13]. In terms of algebraic complexity (the degree of the equations), the complexity of quantifier elimination presented here is similar to [133]. However the bounds in this chapter are more precise in terms of combinatorial complexity (the dependence on the number of equations). Similarly, the complexity of Algorithm 14.10, coming from [19] improves slightly the result of [163] which computes the dimension of a semi-algebraic set with complexity $(s\,d)^{O(k'(k-k'))}$.

The local quantifier elimination algorithm is based on results in [12].