

# The Complete Root Classification of a Parametric Polynomial on an Interval

Songxin Liang  
Department of Applied  
Mathematics, The University  
of Western Ontario, London,  
Ontario, Canada  
sliang22@uwo.ca

David J. Jeffrey  
Department of Applied  
Mathematics, The University  
of Western Ontario, London,  
Ontario, Canada  
djeffrey@uwo.ca

Marc Moreno Maza  
Department of Computer  
Science, The University of  
Western Ontario, London,  
Ontario, Canada  
moreno@csd.uwo.ca

## ABSTRACT

Given a real parametric polynomial  $p(x)$  and an interval  $(a, b) \subset \mathbb{R}$ , the Complete Root Classification (CRC) of  $p(x)$  on  $(a, b)$  is a collection of all possible cases of its root classification on  $(a, b)$ , together with the conditions its coefficients must satisfy for each case. In this paper, a new algorithm is proposed for the automatic computation of the complete root classification of a parametric polynomial on an interval. As a direct application, the new algorithm is applied to some real quantifier elimination problems.

## Categories and Subject Descriptors

I.1.2 [Symbolic and Algebraic Manipulation]: Algorithms—Algebraic algorithms

## General Terms

Algorithms

## Keywords

Complete root classification, real root, parametric polynomial, interval, real quantifier elimination

## 1. INTRODUCTION

The counting and classifying of the roots of a polynomial have been the subject of many investigations. This paper concerns the complete root classification of a parametric polynomial on an interval.

**RC and CRC.** Let  $p(x)$  be a real polynomial with constant coefficients. The *root classification (RC)* of  $p(x)$  on  $\mathbb{R}$  is denoted by

$$[L_1, L_2] = [[n_1, n_2, \dots], [m_1, -m_1, m_2, -m_2, \dots]],$$

where  $n_k$  are the multiplicities of the distinct real roots of  $p(x)$  on  $\mathbb{R}$ , and  $m_k$  are the multiplicities of the distinct complex conjugate pairs of  $p(x)$ , and  $L_1 = [n_1, n_2, \dots]$  is called

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISSAC'08, July 20–23, 2008, Hagenberg, Austria.

Copyright 2008 ACM 978-1-59593-904-3/08/07 ...\$5.00.

the *real RC* of  $p(x)$  on  $\mathbb{R}$ . Let  $a, b \in \mathbb{R} \cup \{-\infty, +\infty\}$ . The *RC of  $p(x)$  on  $(a, b)$*  is denoted by a list  $L = [n_1, n_2, \dots]$ , where  $n_1, n_2, \dots$  are the multiplicities of the distinct real roots of  $p(x)$  on  $(a, b)$ . For a real polynomial  $p(x)$  with parametric coefficients, the *complete root classification (CRC) of  $p(x)$  on  $(a, b)$*  is a collection of all possible cases of its RC on  $(a, b)$ , together with the conditions its coefficients must satisfy for each case.

The history of CRC is short. The CRC of a real parametric quartic polynomial on  $\mathbb{R}$  was found by Arnon in 1988 [2]; the first method for establishing the CRC of a real parametric polynomial of any degree on  $\mathbb{R}$  was given by Yang, Hou and Zeng in 1996 [10]. They illustrated their method by computing the CRC of a reduced sextic polynomial. The first automatic generation of CRCs was described and implemented by Liang and Zhang [7], with some improvements added in [5]. Further improvements to the algorithm were made in [6] by replacing the ‘revised sign lists’ (Definition 4 below) with the direct use of ‘sign lists’. As well as offering greater efficiency, the new algorithm offers a better filter for eliminating non-realizable conditions.

All works above are on  $\mathbb{R}$ , and applications often need CRC on an interval. For example, in robust control [1] and problems concerning program termination [12], we have to determine the conditions on the parametric coefficients of  $p(x)$  such that  $\forall x > 0, p(x) > 0$ , or the conditions such that  $\forall x \in (a, b), p(x) \neq 0$ . Therefore, it is meaningful to develop an algorithm for computing the CRC of a parametric polynomial on an interval.

However, in order to develop such an algorithm, we have to face two challenging problems. The first problem is the determination of the conditions for a parametric polynomial having a given number of real roots on an interval. One naturally thinks of the well-known Sturm sequence. The Sturm sequence of a polynomial with known, constant coefficients is a good tool for computing the number of real roots on an interval, but it is inconvenient and inefficient when the given polynomial has parametric coefficients. A better solution uses the fact that we know how to determine the conditions for a polynomial having a given number of real roots on  $\mathbb{R}$  [6], and converts the problem of the determination of conditions on an interval into a problem on  $\mathbb{R}$ . This is done in Section 3, where Theorem 4 is given. Let  $p \in \mathbb{R}[x]$  with  $p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0$  and  $a_n \neq 0$ . Let  $a, b \in \mathbb{R}$  such that  $p(a) \neq 0$  and  $p(b) \neq 0$ . Let  $\Psi_1(x) = (1-x)^n p\left(\frac{b-ax}{1-x}\right)$  and  $\Psi_2(x) = \Psi_1(-x^2) =$



If there is no such  $m$ , then  $s'$  is the empty list. We define inductively

$$\text{PmV}(s) = \begin{cases} 0, & s' = \emptyset, \\ \text{PmV}(s') + \epsilon_{n-m} \text{sgn}(s_n s_m), & n - m \text{ odd}, \\ \text{PmV}(s'), & n - m \text{ even}, \end{cases}$$

where  $\epsilon_{n-m} = (-1)^{(n-m)(n-m-1)/2}$ .

The following theorem gives the number of distinct roots in terms of sign lists.

**THEOREM 2.** *Let  $D = [D_1, \dots, D_n]$  be the discriminant sequence of a real polynomial  $p(x)$  of degree  $n$ , and  $\ell$  be the maximal index such that  $D_\ell \neq 0$ . If  $\text{PmV}(D) = r$ , then  $p(x)$  has  $r + 1$  distinct real roots and  $\frac{1}{2}(\ell - r - 1)$  pairs of distinct complex conjugate roots.*

The next theorem can be used to detect the non-realizable sign lists in output conditions.

**THEOREM 3.** *Let  $S = [s_1, \dots, s_n]$  and  $R = [r_1, \dots, r_n]$  be the sign list and the revised sign list of  $p(x)$  respectively. Then  $\text{PmV}(S) = \text{PmV}(R)$ .*

At last, we review a result given by Yang and Xia [9][11] for computing the number of real roots on intervals, which gives us some clue for solving the first problem mentioned in Section 1.

Let  $p \in \mathbb{R}[x]$  with  $p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0$  and  $a_n \neq 0$ . Let  $a, b \in \mathbb{R}$  such that  $p(a) \neq 0$  and  $p(b) \neq 0$ . Let  $\Psi_1(x) = (1-x)^n p\left(\frac{b-ax}{1-x}\right)$  and  $\Psi_2(x) = \Psi_1(-x^2) = (1+x^2)^n p\left(\frac{b+ax^2}{1+x^2}\right)$ . Then, it is easy to see that  $\text{coeff}(\Psi_1, x, n) = (-1)^n p(a) \neq 0$ ,  $\text{coeff}(\Psi_2, x, 2n) = p(a) \neq 0$  and  $\Psi_1(0) = \text{coeff}(\Psi_1, x, 0) = \text{coeff}(\Psi_2, x, 0) = p(b) \neq 0$ . Furthermore

**PROPOSITION 4.**  $\#\{x \in (a, b) | p(x) = 0\} = \frac{1}{2} \#\{x \in \mathbb{R} | \Psi_2(x) = 0\}$ .

### 3. BASIS OF THE ALGORITHM

In this section, we establish the basis for the new algorithm. The main idea is that we transfer the computation of CRC for a parametric polynomial on an interval to the computation of CRC for a parametric polynomial on  $\mathbb{R}$ .

**THEOREM 4.** *Let  $p(x), \Psi_1(x), \Psi_2(x)$  be defined as in Section 2. Then,  $[r_1, r_2, \dots, r_k]$  is the RC of  $p(x)$  on  $(a, b)$ , if and only if  $[r_1, r_1, r_2, r_2, \dots, r_k, r_k]$  is the real RC of  $\Psi_2(x)$  on  $\mathbb{R}$ .*

**PROOF.** Since  $[r_1, r_2, \dots, r_k]$  is the RC of  $p(x)$  on  $(a, b)$ , we can decompose  $p(x)$  in  $\mathbb{C}$  as

$$p(x) = a_n \prod_{i=1}^k (x - \alpha_i)^{r_i} \prod_{j=1}^s (x - \beta_j)^{f_j} \prod_{u=1}^t (x - \gamma_u)^{h_u} (x - \overline{\gamma_u})^{h_u},$$

where,  $\alpha_i (i = 1, \dots, k), \beta_j (j = 1, \dots, s) \in \mathbb{R}$  are all distinct real roots of  $p(x)$ ,  $\gamma_u, \overline{\gamma_u} (u = 1, \dots, t) \in \mathbb{C}$  are all pairs of distinct complex-conjugate roots of  $p(x)$ , and  $a < \alpha_i < b, \beta_j \notin [a, b]$ . Then, since  $a, b$  are not roots of  $p(x)$ , we have  $\Psi_1(x) =$

$$a_n \prod_{i=1}^k (\alpha_i - a)^{r_i} \left(x - \frac{\alpha_i - b}{\alpha_i - a}\right)^{r_i} \prod_{j=1}^s (\beta_j - a)^{f_j} \left(x - \frac{\beta_j - b}{\beta_j - a}\right)^{f_j}.$$

$$\prod_{u=1}^t (\gamma_u - a)^{h_u} \left(x - \frac{\gamma_u - b}{\gamma_u - a}\right)^{h_u} (\overline{\gamma_u} - a)^{h_u} \left(x - \frac{\overline{\gamma_u} - b}{\overline{\gamma_u} - a}\right)^{h_u}.$$

Notice that  $\frac{\alpha_i - b}{\alpha_i - a} \neq \frac{\alpha_j - b}{\alpha_j - a} (i \neq j)$ ,  $\frac{\alpha_i - b}{\alpha_i - a} < 0, \frac{\beta_j - b}{\beta_j - a} > 0$  and  $\frac{\gamma_u - b}{\gamma_u - a}, \frac{\overline{\gamma_u} - b}{\overline{\gamma_u} - a} \notin \mathbb{R}$  because of  $\alpha_i \neq \alpha_j, a < \alpha_i < b, \beta_j \notin [a, b]$  and  $a \neq b$  respectively.

Based on the facts above, we can conclude that  $\Psi_1(x)$  in  $(-\infty, 0)$  has exactly  $k$  distinct real roots of respective multiplicities  $r_1, \dots, r_k$ . Therefore, the RC of  $\Psi_1(x)$  on  $(-\infty, 0)$  is  $[r_1, r_2, \dots, r_k]$ , if and only if the RC of  $p(x)$  on  $(a, b)$  is  $[r_1, r_2, \dots, r_k]$ . Noticing  $\Psi_1(0) = p(b) \neq 0$ , by similar discussion, we can conclude that the real RC of  $\Psi_2(x)$  on  $\mathbb{R}$  is  $[r_1, r_1, r_2, r_2, \dots, r_k, r_k]$ , if and only if the RC of  $\Psi_1(x)$  on  $(-\infty, 0)$  is  $[r_1, r_2, \dots, r_k]$ . Finally, the desired result follows.  $\square$

From Theorem 4, for intervals  $(-\infty, a)$  and  $(a, +\infty)$ , we have the following corollary.

**COROLLARY 1.** *Let notations be as in Theorem 4. Then,  $[r_1, r_2, \dots, r_k]$  is the RC of  $p(x)$  on  $(-\infty, a)$  iff  $[r_1, r_1, r_2, r_2, \dots, r_k, r_k]$  is the real RC of  $\Psi_2(x) := p(-x^2 + a)$  on  $\mathbb{R}$ ;  $[r_1, r_2, \dots, r_k]$  is the RC of  $p(x)$  on  $(a, +\infty)$  iff  $[r_1, r_1, r_2, r_2, \dots, r_k, r_k]$  is the real RC of  $\Psi_2(x) := p(x^2 + a)$  on  $\mathbb{R}$ .*

Based on Theorem 4 and Corollary 1, the conditions for  $p(x)$  having  $[r_1, r_2, \dots, r_k]$  as its RC on an interval can be obtained by computing the conditions for  $\Psi_2(x)$  having  $[r_1, r_1, r_2, r_2, \dots, r_k, r_k]$  as its real RC on  $\mathbb{R}$ .

### 4. THE ALGORITHM

Let  $p(x), \Psi_2(x)$  be defined as in Section 2. In this section, we propose an algorithm for computing the CRC of  $p(x)$  on  $(a, b)$ , where  $a, b \in \mathbb{R} \cup \{-\infty, +\infty\}$ . We need some functions to present the algorithm.

- **ClassifySL:** Input a set of sign lists and a positive integer  $i$ ; output the subset of sign lists of which the maximal index of non-vanishing members is  $i$ . For example, if the input is  $S = \{[1, 1, 1, -1, 0, 0], [1, 0, 0, -1, 0, 0], [1, 1, 1, -1, 1, 1], [1, -1, 0, 0, 1, 1]\}$ , then  $\text{ClassifySL}(S, 4) = \{[1, 1, 1, -1, 0, 0], [1, 0, 0, -1, 0, 0]\}$ ,  $\text{ClassifySL}(S, 6) = \{[1, 1, 1, -1, 1, 1], [1, -1, 0, 0, 1, 1]\}$ , but  $\text{ClassifySL}(S, 1) = \text{ClassifySL}(S, 2) = \text{ClassifySL}(S, 3) = \text{ClassifySL}(S, 5) = \emptyset$ .
- **MinusOne:** Input an RC  $[r_1, \dots, r_k]$ ; output  $[r_1 - 1, \dots, r_k - 1]$  (all elements with value 0 are removed).
- **Op:** Input a set  $\{a_1, \dots, a_n\}$  or a list  $[a_1, \dots, a_n]$ ; output the sequence  $a_1, \dots, a_n$ .
- **DRC:** input an RC  $[r_1, \dots, r_k]$ ; output  $[r_1, r_1, \dots, r_k, r_k]$ .
- **AllRC:** Input  $n \in \mathbb{N}$ ; output the union of the sets of partitions of  $0, 1, \dots, n$ .

By Theorem 4 and Corollary 1, the conditions for  $p$  having  $L = [r_1, r_2, \dots, r_k]$  as its RC on  $(a, b)$  can be obtained by computing the conditions for  $\Psi_2$  having  $L_2 = [r_1, r_1, r_2, r_2, \dots, r_k, r_k]$  as its real RC on  $\mathbb{R}$ . So, what we have to do is to find the conditions for  $\Psi_2$  having  $L_2$  as its real RC on  $\mathbb{R}$ .

We first compute all possible sign lists of  $\Psi_2$  for  $\Psi_2$  having  $L_2$  as its real RC on  $\mathbb{R}$ .

### PolySL.

Input:  $\Psi_2$  and  $L_2$ .

Output: The set of all possible sign lists of  $\Psi_2$ .

Procedure:

- Compute the discriminant sequence  $D = [D_1, \dots, D_{2n}]$  of  $\Psi_2$ .
- Compute the set  $S_0$  of all possible sign lists from  $D$ : for  $1 \leq k \leq 2n$ , if  $D_k \in \mathbb{R}$ , then  $D_k \rightarrow \text{sgn}(D_k)$ ; otherwise,  $D_k \rightarrow \{-1, 0, 1\}$ . For example, if  $D = [1, -2, a]$ , then  $S_0 = \{[1, -1, -1], [1, -1, 0], [1, -1, 1]\}$ .
- Compute  $S = \{s \in S_0 \mid \text{PmV}(s) = \text{PmV}(\text{rsl}(s)) = 2k - 1\}$ , where  $\#L_2 = 2k$ .
- Return  $S$ .

Then  $S = \text{PolySL}(\Psi_2, L_2)$  is the set of all possible sign lists of  $\Psi_2$  for  $\Psi_2$  having  $L_2$  as its real RC on  $\mathbb{R}$ . In order to make the multiplicities of the  $2k$  distinct real roots of  $\Psi_2$  be  $r_1, r_1, r_2, r_2, \dots, r_k, r_k$  respectively, we also have to determine the possible sign lists of the polynomials in the  $\Delta$ -sequence of  $\Psi_2$ . According to Proposition 1,  $\Delta^1(\Psi_2)$  can be determined by the maximal index  $\ell$  of non-vanishing members in the sign list of  $\Psi_2$ , which actually is the total number of distinct (real and complex) roots of  $\Psi_2$ . Since  $L_2$  does not contain information about the number of distinct complex roots of  $\Psi_2$ , the maximal index  $\ell$  is not uniquely determined. Therefore, unlike the case of RC on  $\mathbb{R}$  [6], there may be more than one  $\Delta^1(\Psi_2)$  for the real RC  $L_2$ , and consequently the conditions for  $\Psi_2(x)$  having  $L_2$  as its real RC on  $\mathbb{R}$  would be more complicated. The following observations can be used to solve the problem.

Let  $G_i = \text{ClassifySL}(S, i)$ . Then  $S = \bigcup_{i=k}^n G_{2i}$  because  $\Psi_2$  has at least  $2k$  distinct roots and, by Theorem 4 and its proof, the number of distinct roots of  $\Psi_2$  is always a multiple of 2. For  $k \leq i \leq n$ , if the sign list of  $\Psi_2$  belongs to  $G_{2i}$ , then the number of distinct roots of  $\Psi_2$  is  $2i$ . So by Proposition 1,  $\Delta^1(\Psi_2) = P_{2n-2i}(\Psi_2, \Psi_2')$ . It is a polynomial of degree  $2n - 2i$ . By Proposition 2, the real RC of  $\Delta^1(\Psi_2)$  on  $\mathbb{R}$  is  $\text{MinusOne}(L_2)$ . Then the computation above can be repeated for  $\Delta^1(\Psi_2)$  and  $\text{MinusOne}(L_2)$  until the termination conditions (see below) are reached. Therefore, unlike the case of RC on  $\mathbb{R}$ , we have to divide all possible sign lists of  $\Psi_2$  into different groups. For each group, there is a distinct  $\Delta^1(\Psi_2)$ .

For the following three *termination conditions*, we can determine just by  $S$  that the real RC of  $\Psi_2$  on  $\mathbb{R}$  is  $L_2$  without further computation for the sign lists of other polynomials in the  $\Delta$ -sequence of  $\Psi_2$ .

1.  $k = n$ ,
2.  $k = 0$ ,
3. the last entry of the sign list of  $\Psi_2$  is nonzero.

Therefore, the algorithm for generating the necessary and sufficient conditions for  $p(x)$  having  $L = [r_1, r_2, \dots, r_k]$  as its RC on  $(a, b)$  is as follows. Let  $\Psi_2$  and  $L_2$  be defined as above. The output conditions are a sequence of *mixed lists*. Each mixed list consists of a polynomial in the  $\Delta$ -sequence of  $\Psi_2$ , followed by all of its possible sign lists. We denote the empty sequence by NULL. Notice that if the empty sequence is returned, then  $L_2$ , and consequently  $L$  are not realizable.

### IntCond

Input:  $\Psi_2$  and a real RC  $L_2$ .

Output: A sequence of mixed lists (the conditions for  $\Psi_2$  having  $L_2$  as its real RC on  $\mathbb{R}$ ).

Procedure:

```

t ← #L2
m ← deg(Ψ2)
S ← PolySL(Ψ2, L2)
if S = ∅ then
  return NULL
elif t = 0 or t = m or ClassifySL(S, m) = S then
  return [Ψ2, Op(S)] (*)
else
  for i from t to m by 2 do
    Gi ← ClassifySL(S, i)
    if Gi ≠ ∅ then
      if i = m then
        return [Ψ2, Op(Gi)] (**)
      else
        Ci ← IntCond(Δ1(Ψ2), MinusOne(L2))
        if Ci ≠ NULL then
          return [Ψ2, Op(Gi), Ci]
        else
          return NULL

```

*Remark 1.* Notice that  $\Delta^1(\Psi_2)$  depends on  $i$ . By Proposition 1, it is a polynomial of degree  $2n - i$ . In the algorithm,  $t$  is a nonnegative integer representing the number of distinct real roots of  $\Psi_2$ .  $m$  is a positive integer representing the degree of  $\Psi_2$ .  $S$  is a set of sign lists representing all possible sign lists of  $\Psi_2$  for  $\Psi_2$  having  $L_2$  as its real RC on  $\mathbb{R}$ .  $G_i$  is the subset of  $S$  such that the maximal index of non-vanishing members in the sign lists of  $G_i$  is  $i$ . Finally,  $C_i$  is a sequence of mixed lists representing the conditions for  $\Delta^1(\Psi_2)$  having  $\text{MinusOne}(L_2)$  as its real RC on  $\mathbb{R}$ .

**PROOF.** First, observe that the number of recursions in the algorithm is finite. So the algorithm will terminate in finite steps.

Second, the output conditions can be viewed as a couple of trees whose nodes consist of mixed lists. Their roots are those mixed lists of which the first member is  $\Psi_2$ , and their leaves are the mixed lists at points (\*) or (\*\*) in the algorithm.

For the case of RC on  $\mathbb{R}$  [6], the necessary and sufficient conditions for a polynomial having a given RC is a single chain of mixed lists connected using conjunction ( $\wedge$ ). Now suppose the leaves are labeled by  $i$  ( $i = 1, \dots, N$ ). Let  $\Theta_i$  be the chain of mixed lists obtained by connecting the nodes from the  $i$ th leaf to its root using conjunction. Then, the output conditions can be expressed as the disjunction of these chains,  $\bigvee_{i=1}^N \Theta_i$ .

Using Theorem 2 and Proposition 2 and 3, as we did in [6], we can prove that  $\bigvee_{i=1}^N \Theta_i$  is the necessary and sufficient conditions for  $\Psi_2$  having  $L_2$  as its real RC on  $\mathbb{R}$ .

Finally, by Theorem 4, the correctness of IntCond follows.  $\square$

Based on the algorithm IntCond above, we can now propose an algorithm for generating the CRC of a parametric polynomial  $p(x)$  on an interval  $(a, b)$ .

### IntCRC

Input: A real parametric polynomial  $p(x)$ , and  $a, b \in \mathbb{R} \cup \{-\infty, +\infty\}$ .

Output: The CRC of  $p(x)$  on  $(a, b)$ .

Procedure:

```

 $\mathcal{L} \leftarrow \text{AllRC}(\text{deg}(p))$ 
compute  $\Psi_2$ 
for  $L$  in  $\mathcal{L}$  do
   $C \leftarrow \text{IntCond}(\Psi_2, \text{DRC}(L))$ 
  if  $C \neq \text{NULL}$  then
    return  $L$  and  $C$ 

```

**Optimization of algorithm.** Finally we discuss the optimization of the algorithm. In comparison with the case on  $\mathbb{R}$ , the output conditions of the CRC of a parametric polynomial on an interval is usually large, especially when the parametric polynomial has a general form. So there remains the work of condensing the output conditions. Suppose  $[D_1, \dots, D_{2n}]$  is the discriminant sequence of  $\Psi_2$ , and  $S$  is the set of all possible sign lists of  $\Psi_2$  for  $\Psi_2$  having  $L_2 = [r_1, r_1, \dots, r_k, r_k]$  as its real RC on  $\mathbb{R}$ . Notice that the length of any sign list in  $S$  is  $2n$  and the first element of it can always be assumed to be 1.

For  $S_1, S_2, S_3 \in S$ , if there is an index  $i$  such that for any index  $j \neq i$ , we have  $S_1[j] = S_2[j] = S_3[j]$  and  $\{S_1[i], S_2[i], S_3[i]\} = \{-1, 0, 1\}$ , then  $S_1, S_2, S_3$  can be represented by a single list  $[S_1[1], \dots, S_1[i-1], *, S_1[i+1], \dots, S_1[2n]]$ , where ‘\*’ means that  $D_i$  is unconstrained.

For  $S_1, S_2 \in S$ , if there is an index  $i$  such that for any index  $j \neq i$ , we have  $S_1[j] = S_2[j]$  and  $\{S_1[i], S_2[i]\} = \{0, 1\}$ , then  $S_1, S_2$  can be represented by a single list  $[S_1[1], \dots, S_1[i-1], 0+, S_1[i+1], \dots, S_1[2n]]$ , where ‘0+’ means that  $D_i \geq 0$ .

Similarly, for  $S_1, S_2 \in S$ , if there is an index  $i$  such that for any index  $j \neq i$ , we have  $S_1[j] = S_2[j]$  and  $\{S_1[i], S_2[i]\} = \{0, -1\}$ , then  $S_1, S_2$  can be represented by a single list  $[S_1[1], \dots, S_1[i-1], 0-, S_1[i+1], \dots, S_1[2n]]$ , where ‘0-’ means that  $D_i \leq 0$ .

At last, for  $S_1, S_2 \in S$ , if there is an index  $i$  such that for any index  $j \neq i$ , we have  $S_1[j] = S_2[j]$  and  $\{S_1[i], S_2[i]\} = \{1, -1\}$ , then  $S_1, S_2$  can be represented by a single list  $[S_1[1], \dots, S_1[i-1], <>, S_1[i+1], \dots, S_1[2n]]$ , where ‘<>’ means that  $D_i \neq 0$ .

In this way,  $S$  can be condensed into a smaller set  $S'$ . For the sake of simplicity, we still call the elements in  $S'$  sign lists. For example, if  $S = \{[1, 0, 1, 1], [1, 0, 0, 1], [1, 0, -1, 1]\}$ , then  $S' = \{[1, 0, *, 1]\}$ .

**Complexity analysis.** We give an upper bound for the running time of the algorithm **IntCond**. We count the number of operations  $T(n)$  on the coefficients, regarding all polynomials as univariate in  $x$ . We recall that the input polynomial  $\Psi_2$  has degree  $2n$ . We first estimate the call  $S = \text{PolySL}(\Psi_2, L_2)$  and make three observations: (1) computing the discriminant sequence  $D$  of  $\Psi_2$  can be done in  $O(n^2)$  coefficient operations, (2) computing the set  $S_0$  of all possible sign lists from  $D$  requires  $O(3^{2n})$  operations on coefficients (and integers), (3) for each  $s \in S_0$ , checking whether  $s$  belongs to the output set  $S$  can be done in linear time (w.r.t.  $n$ ), and there are  $3^{2n}$  sign lists in  $S_0$ . So computing  $S$  requires  $O(n3^{2n})$  operations

Consequently the call  $S = \text{PolySL}(\Psi_2, L_2)$  runs in  $O(n3^{2n})$  operations on coefficients. Returning to the body of the al-

gorithm **IntCond**, we observe that all the calls  $\text{ClassifySL}(S, i)$  together have a linear cost w.r.t. the size of  $S$ , hence these costs can be absorbed in the cost for creating  $S$ . The remaining costs in the body of **IntCond** are (at most)  $n-1$  recursive calls with respective costs  $T(n-1), T(n-2), \dots, T(1)$ . Therefore, we have

$$T(n) \leq O(n3^{2n}) + T(n-1) + T(n-2) + \dots + T(2) + T(1).$$

Unrolling the recurrence, performing elementary computations we obtain:  $T(n) \in O(n9^n)$ .

## 5. EXAMPLES

In this section, we present some examples of the CRCs for some parametric polynomials on intervals. As a direct application, we show how to apply the CRC to questions concerning real quantifier elimination. As with many symbolic computations, solving general cases of high degree polynomials usually would result in very large solutions. Therefore, we only focus on sparse parametric polynomials. All computations were performed with Maple 10 running on a 1.6 GHz Pentium CPU. Notice that, in the output conditions below, different chains of mixed lists (see the proof of **IntCond**) are separated by semicolons (;).

*Example 1.* The following is the CRC of  $p_3 = x^3 + ax + b$  on  $(0, 2)$ , where the initial conditions correspond to  $p_3(0) \neq 0$  and  $p_3(2) \neq 0$ . It takes 1.18 seconds to generate the CRC. After the output, there is an explanation of its interpretation.

(\*)  $p_3 = x^3 + a*x + b$

The CRC of  $p_3$  on  $(0, 2)$  is:

([polynomial, its all possible sign lists])

(1) [ ], if and only if [P6, [1, -1, 0, 0, <>, -1], [1, -1, -1, 1, 0, 0], [1, -1, -1, 0+, 0, -1], [1, -1, -1, 1, -1, -1], [1, -1, -1, \*, 1, -1], [1, 0, 0, -1, 1, -1], [1, 1, 1, -1, 1, -1]]

(2) [1], if and only if [P6, [1, 1, 1, -1, 0, 0], [1, 0, 0, -1, 0, 0], [1, -1, -1, -1, 0, 0]], [P62, [1, -1]]; [P6, [1, 1, 1, -1, 1, 1], [1, -1, 0, 0, 1, 1], [1, 0-, -1, -1, 1, 1], [1, 1, 1, <>, -1, 1], [1, 1, 1, 0, 0, 1], [1, -1, -1, 1, 1, 1], [1, -1, -1, 0, 0, 1], [1, -1, -1, -1, -1, 1], [1, 0, 0, -1, <>, 1]]

(3) [2], if and only if [P6, [1, 1, 1, -1, 0, 0], [1, 0, 0, -1, 0, 0], [1, -1, -1, -1, 0, 0]], [P62, [1, 1]]

(4) [1, 1], if and only if [P6, [1, -1, -1, -1, -1, -1], [1, 1, 1, -1, -1, -1], [1, 0, 0, -1, -1, -1]],

where,

P6: =  $8+2*a+4*a*x^2+2*a*x^4+b+3*b*x^2+3*b*x^4+b*x^6$ ,  
P62: =  $b*a^2*x^2+9*b^2*x^2+2*a^3*x^2+6*b*a*x^2$   
 $+b*a^2+8*a^2+2*a^3$ .

And the initial conditions are

$b <> 0, 8+2*a+b <> 0$

Let us explain the CRC of  $p_3$  on  $(0, 2)$ . We assume that the initial conditions  $b \neq 0$  and  $2a + b + 8 \neq 0$  hold. All possible cases of RC for  $p_3$  on  $(0, 2)$  are [], [1], [2], [1, 1], [3], [1, 2], [1, 1, 1].

The conditions for  $p_3$  having [1] (one single root) as its RC in  $(0, 2)$  can be obtained by computing the conditions for  $P_6$  (which is  $\Psi_2$  in the algorithm) having [1, 1] as its real RC on  $\mathbb{R}$ .

All possible sign lists of  $P_6$  would be  $[1, 1, 1, -1, 0, 0]$ ,  $[1, 0, 0, -1, 0, 0]$ ,  $[1, -1, -1, -1, 0, 0]$ ,  $[1, 1, 1, -1, 1, 1]$ ,  $[1, -1, 0, 0, 1, 1]$ ,  $[1, 0, -1, -1, 1, 1]$ ,  $[1, 1, 1, \langle \rangle, -1, 1]$ ,  $[1, 1, 1, 0, 0, 1]$ ,  $[1, -1, -1, 1, 1, 1]$ ,  $[1, -1, -1, 0, 0, 1]$ ,  $[1, -1, -1, -1, -1, 1]$ ,  $[1, 0, 0, -1, \langle \rangle, 1]$ .

Now these sign lists of  $P_6$  can be divided into two groups:  $G_4 = \{[1, 1, 1, -1, 0, 0], [1, 0, 0, -1, 0, 0], [1, -1, -1, -1, 0, 0]\}$  and  $G_6 = \{[1, 1, 1, -1, 1, 1], [1, -1, 0, 0, 1, 1], [1, 0, -1, -1, 1, 1], [1, 1, 1, \langle \rangle, -1, 1], [1, 1, 1, 0, 0, 1], [1, -1, -1, 1, 1, 1], [1, -1, -1, 0, 0, 1], [1, -1, -1, -1, -1, 1], [1, 0, 0, -1, \langle \rangle, 1]\}$ .

If the sign list of  $P_6$  belongs to  $G_4$ , then the number of distinct roots of  $P_6$  is 4. So the 'repeated part'  $\Delta^1(P_6) = P_{62}$  and the RC of  $P_{62}$  is  $\text{MinusOne}([1, 1]) = []$ . For  $P_{62}$  and  $[], \text{IntCond}$  is called again, obtaining that the condition for  $P_{62}$  having  $[], \text{IntCond}$  on  $\mathbb{R}$  is its sign list being  $[1, -1]$ .

At this point, the termination condition 2 is satisfied, so  $\text{IntCond}$  terminates. If the sign list of  $P_6$  belongs to  $G_6$ , then the termination condition 3 is satisfied, and  $\text{IntCond}$  terminates.

In summary,  $p_3$  has  $[1]$  as its RC on  $(0, 2)$ , if and only if the sign list of  $P_6$  belongs to  $G_4$  and the sign list of  $P_{62}$  is  $[1, -1]$ , or the sign list of  $P_6$  belongs to  $G_6$ . The cases  $[], [2]$  and  $[1, 1]$  can be explained similarly.

For the cases  $[3], [1, 2], [1, 1, 1]$ , since the output of  $\text{IntCond}$  is the empty sequence  $\text{NULL}$ , they are not realizable. Based on the CRC of  $p_3$ , we can answer some questions concerning real quantifier elimination. The discriminant sequence of  $P_6$  is  $[1, D_2, D_3, D_4, D_5, D_6]$ , where

$$\begin{aligned} D_2 &= -3b^2 - 2ab, \\ D_3 &= -a^2b(2a + 3b), \\ D_4 &= a^2b(a^2b + 9b^2 + 2a^3 + 6ab), \\ D_5 &= -b(a^2b + 2a^3 + 6ab + 9b^2)(4a^3 + 27b^2), \\ D_6 &= -(8 + 2a + b)b(4a^3 + 27b^2)^2. \end{aligned}$$

The necessary and sufficient condition for  $\forall x \in (0, 2)[p_3 \neq 0]$  is that case (1) holds, and case (1) holds iff the sign list of  $P_6$  be one of the following:  $[1, -1, 0, 0, \langle \rangle, -1]$ ,  $[1, -1, -1, 1, 0, 0]$ ,  $[1, -1, -1, 0, 0, -1]$ ,  $[1, -1, -1, 1, -1, -1]$ ,  $[1, -1, -1, *, 1, -1]$ ,  $[1, 0, 0, -1, 1, -1]$ ,  $[1, 1, 1, -1, 1, -1]$ . Therefore, the necessary and sufficient condition for  $\forall x \in (0, 2)[p_3 \neq 0]$  is

$$\begin{aligned} &[D_2 < 0 \wedge D_3 = 0 \wedge D_4 = 0 \wedge D_5 \neq 0 \wedge D_6 < 0] \vee [D_2 < 0 \wedge D_3 < 0 \wedge D_4 > 0 \wedge D_5 = 0 \wedge D_6 = 0] \vee [D_2 < 0 \wedge D_3 < 0 \wedge D_4 \geq 0 \wedge D_5 = 0 \wedge D_6 < 0] \vee [D_2 < 0 \wedge D_3 < 0 \wedge D_4 > 0 \wedge D_5 < 0 \wedge D_6 < 0] \vee [D_2 < 0 \wedge D_3 < 0 \wedge D_5 > 0 \wedge D_6 < 0] \vee [D_2 = 0 \wedge D_3 = 0 \wedge D_4 < 0 \wedge D_5 > 0 \wedge D_6 < 0] \vee [D_2 > 0 \wedge D_3 > 0 \wedge D_4 < 0 \wedge D_5 > 0 \wedge D_6 < 0]. \end{aligned}$$

Similarly, the necessary and sufficient condition for  $p_3$  having two distinct real roots of multiplicities 1 on  $(0, 2)$  is that case (4) holds. That is

$$\begin{aligned} &[D_2 < 0 \wedge D_3 < 0 \wedge D_4 < 0 \wedge D_5 < 0 \wedge D_6 < 0] \vee [D_2 > 0 \wedge D_3 > 0 \wedge D_4 < 0 \wedge D_5 < 0 \wedge D_6 < 0] \vee [D_2 = 0 \wedge D_3 = 0 \wedge D_4 < 0 \wedge D_5 < 0 \wedge D_6 < 0]. \end{aligned}$$

*Example 2.* The following is the CRC of  $p_4 = x^4 + ax + b$  on  $(-\infty, -1)$ , where the initial condition corresponds to  $p_4(-1) \neq 0$ . It takes 0.75 second to generate the CRC.

(\*)  $p4:=x^4+a*x+b$   
The CRC of  $p4$  on  $(-\text{Infinity}, -1)$  is:  
([polynomial, its all possible sign lists])

(1)  $[],$  if and only if

$[P8, [1, -1, 0, 0, 0, 0, 0, 0], [1, -1, 0, 0, -1, -1, -1, 1], [1, -1, 0, 0, 0, 0, \langle \rangle, 1], [1, -1, 0, 0, \langle \rangle, -1, 0, 0], [1, -1, 0, 0, 1, *, -1, 1], [1, -1, 0, 0, \langle \rangle, -1, 1, 1]]$

(2)  $[1],$  if and only if  
 $[P8, [1, -1, 0, 0, 0, 0, -1, -1], [1, -1, 0, 0, 1, -1, -1, -1], [1, -1, 0, 0, 1, 0, 0, -1], [1, -1, 0, 0, 1, 1, 1, -1], [1, -1, 0, 0, -1, -1, -1, -1]]$

(3)  $[1, 1],$  if and only if  
 $[P8, [1, -1, 0, 0, 1, 1, 1, 1]]$

Where,

(#1)  $P8:=1+4*x^2+6*x^4+4*x^6+x^8-a*a*x^2+b,$   
and the initial condition is  
 $1-a+b \langle \rangle 0$

*Example 3.* Find the conditions on  $a, b, c$  such that  $\forall x > 0, p_5 = x^5 + ax^2 + bx + c > 0$ .

This problem was studied by Yang in [9]. Here we try to use the CRC method to solve this problem. First, we compute the CRC of  $p_5$  on  $(0, +\infty)$ . It takes 6.33 seconds.

(\*)  $p5:=x^5+a*x^2+b*x+c$   
The CRC of  $p5$  on  $(0, +\text{Infinity})$  is:  
([polynomial, its all possible sign lists])

(1)  $[],$  if and only if  
 $[P10, [1, 0, 0, 0, 0, -1, 1, *, -1, -1], [1, 0, 0, 0, 0, *, -1, 1, 0, 0], [1, 0, 0, 0, 0, -1, 1, 1, 1, -1], [1, 0, 0, 0, 0, -1, 1, 1, 0, 0], [1, 0, 0, 0, 0, -1, 0, 0, -1, -1], [1, 0, 0, 0, 0, *, -1, 0, +, 0, -1], [1, 0, 0, 0, 0, 0, +, 0, -1, 1, -1], [1, 0, 0, 0, 0, 0, -, 0, 1, *, -1], [1, 0, 0, 0, 0, *, -1, *, 1, -1], [1, 0, 0, 0, 0, -1, 0, 0, 0, 0], [1, 0, 0, 0, 0, 1, 1, -1, 1, -1], [1, 0, 0, 0, 0, *, -1, 1, -1, -1], [1, 0, 0, 0, 0, *, 0, 0, 1, -1], [1, 0, 0, 0, 0, -, 0, 0, 0, -1], [1, 0, 0, 0, 0, -1, 1, 0, +, 0, -1], [1, 0, 0, 0, 0, 0, -, 0, 1, 0, 0]]$

(2)  $[1, 2],$  if and only if  
 $[P10, [1, 0, 0, 0, 0, 1, 1, 1, 0, 0]], [P102, [1, 1]]$

(3)  $[1],$  if and only if  
 $[P10, [1, 0, 0, 0, 0, 1, 0, 0, 0, 0]], [P104, [1, -1, 0, 0], [1, -1, 1, 1], [1, 0, 0, 1], [1, -1, 0, 1], [1, 1, -1, 1], [1, 0, -1, 1], [1, -1, -1, 1]]; [P10, [1, 0, 0, 0, 0, 1, 1, -1, 0, 0], [1, 0, 0, 0, 0, *, -1, -1, 0, 0], [1, 0, 0, 0, 0, 0, +, 0, -1, 0, 0]], [P102, [1, -1]]; [P10, [1, 0, 0, 0, 0, -1, 1, 1, 1, 1], [1, 0, 0, 0, 0, 1, 1, 0, -, 0, 1], [1, 0, 0, 0, 0, 1, 1, -1, 1, 1], [1, 0, 0, 0, 0, *, -1, 0, -, 0, 1], [1, 0, 0, 0, 0, 1, 0, 0, -1, 1], [1, 0, 0, 0, 0, 1, 1, *, -1, 1], [1, 0, 0, 0, 0, *, 0, 0, 1, 1], [1, 0, 0, 0, 0, -, 0, 1, 1, 1], [1, 0, 0, 0, 0, *, -1, *, 1, 1], [1, 0, 0, 0, 0, *, -1, -1, -1, 1], [1, 0, 0, 0, 0, 0, +, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, -, 0, 1, *, 1]]$

(4)  $[2],$  if and only if  
 $[P10, [1, 0, 0, 0, 0, 1, 0, 0, 0, 0]], [P104, [1, 0, -1, -1], [1, -1, -1, -1], [1, 1, -1, -1], [1, 1, 1, -1], [1, 1, 0, -1], [1, 0, 0, -1]]; [P10, [1, 0, 0, 0, 0, 0, -1, 0, 0], [1, 0, 0, 0, 0, 1, -1, -1, 0, 0], [1, 0, 0, 0, 0, -1, -1, 0, 0], [1, 0, 0, 0, 0, -1, -1, 0, 0], [1, 0, 0, 0, 0, 1, 1, -1, 0, 0], [1, 0, 0, 0, 0, 1, 0, -1, 0, 0]], [P102, [1, -1]]$

(5)  $[3],$  if and only if  $[P10, [1, 0, 0, 0, 0, 1, 0, 0, 0, 0]], [P104, [1, 1, 0, 0]], [P1042, [1, 1]]$

(6)  $[1, 1],$  if and only if

[P10, [1, 0, 0, 0, 0, 1, 1, 1, 0, 0]], [P102, [1, -1]]; [P10,  
 [1, 0, 0, 0, 0, 1, 1, 1, -1], [1, 0, 0, 0, 0, 1, 0, 0, -1, -1],  
 [1, 0, 0, 0, 0, 1, 1, \*, -1, -1], [1, 0, 0, 0, 0, 1, 1, 0+, 0, -1],  
 [1, 0, 0, 0, 0, 0+, 0, -1, -1, -1], [1, 0, 0, 0, 0, \*, -1, -1, -1, -1]]

(7) [1, 1, 1], if and only if

[P10, [1, 0, 0, 0, 0, 1, 1, 1, 1, 1]]

Where,

(#1) P1042:=-2\*b\*x^2-5\*c

(#2) P102:=54\*a^4\*c+27\*b\*a^4\*x^2-225\*x^2\*c^2\*a^2  
 +600\*a\*c^2\*b+720\*a\*x^2\*c\*b^2-320\*c\*b^3-256\*x^2\*b^4,

(#3) P10:=x^10+a\*x^4+b\*x^2+c,

(#4) P104:=-3\*a\*x^4-4\*b\*x^2-5\*c,

and the initial condition is  
 $c < 0$

The discriminant sequence of  $P_{10}$  is  $[1, 0, 0, 0, 0, D_6, D_7, D_8, D_9, D_{10}]$ , where

$$\begin{aligned} D_6 &= -a^5, & D_7 &= -a^3(27a^4 + 300abc - 160b^3), \\ D_8 &= (300bac - 160b^3 + 27a^4)(720acb^2 - 256b^4 + 27a^4b - 225a^2c^2), \\ D_9 &= -(720acb^2 - 256b^4 + 27a^4b - 225a^2c^2)(-1600b^3ca + 256b^5 - 27a^4b^2 + 2250ba^2c^2 + 3125c^4 + 108a^5c), \\ D_{10} &= -c(-1600b^3ca + 256b^5 - 27a^4b^2 + 2250ba^2c^2 + 3125c^4 + 108a^5c)^2 \end{aligned}$$

Again, we assume that the initial condition  $c \neq 0$  holds. Then  $(\forall x > 0)[p_5 = x^5 + ax^2 + bx + c > 0]$  iff case (1) holds. That is  $[D_6 < 0 \wedge D_7 > 0 \wedge D_9 < 0 \wedge D_{10} < 0] \vee [D_7 < 0 \wedge D_8 > 0 \wedge D_9 = 0 \wedge D_{10} = 0] \vee [D_6 < 0 \wedge D_7 > 0 \wedge D_8 > 0 \wedge D_9 > 0 \wedge D_{10} < 0] \vee [D_6 < 0 \wedge D_7 > 0 \wedge D_8 > 0 \wedge D_9 = 0] \vee [D_6 < 0 \wedge D_7 = 0 \wedge D_8 = 0 \wedge D_9 < 0 \wedge D_{10} < 0] \vee [D_7 < 0 \wedge D_8 \geq 0 \wedge D_9 = 0 \wedge D_{10} < 0] \vee [D_6 \geq 0 \wedge D_7 = 0 \wedge D_8 < 0 \wedge D_9 > 0 \wedge D_{10} < 0] \vee [D_6 \leq 0 \wedge D_7 = 0 \wedge D_8 > 0 \wedge D_{10} < 0] \vee [D_7 < 0 \wedge D_9 > 0 \wedge D_{10} < 0] \vee [D_6 < 0 \wedge D_7 = 0 \wedge D_8 = 0 \wedge D_9 = 0 \wedge D_{10} = 0] \vee [D_6 > 0 \wedge D_7 > 0 \wedge D_8 < 0 \wedge D_9 > 0 \wedge D_{10} < 0] \vee [D_7 < 0 \wedge D_8 > 0 \wedge D_9 < 0 \wedge D_{10} < 0] \vee [D_7 = 0 \wedge D_8 = 0 \wedge D_9 > 0 \wedge D_{10} < 0] \vee [D_6 \leq 0 \wedge D_7 = 0 \wedge D_8 = 0 \wedge D_9 = 0 \wedge D_{10} < 0] \vee [D_6 < 0 \wedge D_7 > 0 \wedge D_8 \geq 0 \wedge D_9 = 0 \wedge D_{10} < 0] \vee [D_6 \leq 0 \wedge D_7 = 0 \wedge D_8 > 0 \wedge D_9 = 0 \wedge D_{10} = 0]$ .

This solution appears to be different from that given by Yang [9]. To obtain the same form as he gave, one might apply additional transformations to the result. For example, one writes  $D_7 = -a^3(27a^4 + 300abc - 160b^3) = a^3d_8$  with  $d_8$  taking the obvious value. Then the condition  $D_7 > 0$  can be written as  $(a^3 > 0 \wedge d_8 > 0) \vee (a^3 < 0 \wedge d_8 < 0)$ . It is now possible to use Boolean algebra to transform the above result into Yang's form. Furthermore, the CRC of  $p_5$  on  $(0, +\infty)$  can be used to solve other problems easily. For example, the necessary and sufficient condition for  $p_5$  having two distinct positive roots of multiplicities 1 is that case (6) holds. Thus the condition is

$$\begin{aligned} [D_6 > 0 \wedge D_7 > 0 \wedge D_8 > 0 \wedge D_9 = 0 \wedge D_{10} = 0 \wedge E_2 < 0] \vee [D_6 > 0 \wedge D_7 > 0 \wedge D_8 > 0 \wedge D_9 > 0 \wedge D_{10} < 0] \vee [D_6 > 0 \wedge D_7 = 0 \wedge D_8 = 0 \wedge D_9 < 0 \wedge D_{10} < 0] \vee [D_6 > 0 \wedge D_7 > 0 \wedge D_9 < 0 \wedge D_{10} < 0] \vee [D_6 > 0 \wedge D_7 > 0 \wedge D_8 \geq 0 \wedge D_9 = 0 \wedge D_{10} < 0] \vee [D_6 \geq 0 \wedge D_7 = 0 \wedge D_8 < 0 \wedge D_9 < 0 \wedge D_{10} < 0] \vee [D_7 < 0 \wedge D_8 < 0 \wedge D_9 < 0 \wedge D_{10} < 0], \end{aligned}$$

where  $[1, E_2]$  is the discriminant sequence of  $P_{102}$ , and  $E_2 = -c(300bac - 160b^3 + 27a^4)(720acb^2 - 256b^4 + 27a^4b - 225a^2c^2)$ .

Also, we can conclude that it is impossible for  $p_5$  to have four distinct positive roots.

## 6. CONCLUSION

In this paper, we have proposed a new algorithm for the automatic computation of the complete root classification of a parametric polynomial on an interval. However, some issues deserve further consideration. Although Theorem 3 is used in the algorithm to filter non-realizable sign lists, it is not guaranteed that all non-realizable sign lists are detected and deleted. Furthermore, the output conditions are basically equalities and inequalities in terms of the parametric coefficients. A further step would be to determine what are the possible values of the parametric coefficients such that the conditions described are satisfied. This is essentially the problem of solving semi-algebraic systems, a problem well-known to be difficult. This problem may be addressed using interval analysis [4] or method based on Gröbner basis [8]. We will leave these issues in further work.

## 7. REFERENCES

- [1] H. Anai and H. Yanami. SyNRAC: A Maple-package for solving real algebraic constraints. *Lecture Notes in Computer Science*, 2657:828-837, 2003.
- [2] D. S. Arnon. Geometric reasoning with logic and algebra. *Artificial Intelligence*, 37:37-60, 1988.
- [3] S. Basu, R. Pollack, M.-F. Roy. *Algorithms in Real Algebraic Geometry*, 2nd edition. Springer-Verlag, 2003.
- [4] A. Colagrossi and A. M. Miola Computing real zeros of polynomials with parametric coefficients. *ACM SIGSAM Bulletin*, 17(1):12-15, 1983.
- [5] S. Liang and D. J. Jeffrey. An Algorithm for computing the complete root classification of a parametric polynomial. *Lecture Notes in Computer Science*, 4120:116-130, 2006.
- [6] S. Liang and D. J. Jeffrey. The automatic computation of the complete root classification for a parametric polynomial. Electronic proceedings of MEGA 2007, [www.ricam.oeaw.ac.at/mega2007/electronic/30.pdf](http://www.ricam.oeaw.ac.at/mega2007/electronic/30.pdf).
- [7] S. Liang and J. Zhang. A complete discrimination system for polynomials with complex coefficients and its automatic generation. *Science in China (Series E)*, 42(2):113-128, 1999.
- [8] F. Rouillier. On solving parametric systems. In Workshop on *Challenges in Linear and Polynomial Algebra in Symbolic Computation Software*. Banff International Research Center, 2005.
- [9] L. Yang. Recent advances on determining the number of real roots of parametric polynomials. *Journal of Symbolic Computation*, 28:225-242, 1999.
- [10] L. Yang, X. Hou, and Z. Zeng. Complete discrimination system for polynomials. *Science in China (Series E)*, 39(6):628-646, 1996.
- [11] L. Yang and B. Xia. Explicit criterion to determine the number of positive roots of a polynomial. *MM Research Preprints*, 15:134-145, 1997.
- [12] L. Yang and B. Xia. Quantifier elimination for quartics. *Lecture Notes in Computer Science*, 4120:131-145, 2006.