Deciding Polynomial-Exponential Problems

Melanie Achatz
Faculty of Mathematics and Informatics
University of Passau
D-94030 Passau, Germany
Melanie_Achatz@gmx.de

Scott McCallum
Department of Computing
Macquarie University
NSW 2109, Australia
scott@ics.mq.edu.au

Volker Weispfenning
Faculty of Mathematics and Informatics
University of Passau
D-94030 Passau, Germany
weispfen@uni-passau.de

ABSTRACT

This paper presents a decision procedure for a certain class of sentences of first order logic involving integral polynomials and the exponential function in which the variables range over the real numbers. The inputs to the decision procedure are prenex sentences in which only the outermost quantified variable can occur in the exponential function. The decision procedure has been implemented in the computer logic system REDLOG. Closely related work is reported in [2, 7, 16, 20, 24].

Categories and Subject Descriptors

G.4 [**Mathematics of Computation**]: Mathematical software

General Terms

Algorithms, Theory

Keywords

Decision procedure, exponential polynomials

1. INTRODUCTION

In 1948 Tarski published a proof that the first order theory of the real numbers is decidable: indeed he exhibited a decision method for this theory [23]. In his monograph Tarski briefly considered an extended system in which one introduces a unary function symbol for exponentiation with respect to a fixed base. He remarked that the decision problem for such a system, which was still an open problem in 1948, is of great theoretical and practical interest. Over the following years many efforts were made to resolve this decision problem. The problem was conditionally solved in the positive sense in 1996 by Macintyre and Wilkie [15]. Their solution relies upon the plausible yet unproven Schanuel's conjecture in transcendental number theory. Their solution also is an indirect one, using powerful model-theoretic machinery; so it is not well suited for implementation.

A number of papers have addressed, without recourse to Schanuel's conjecture and with practical implementability in mind, decision problems for fragments of the full first order theory of the reals with a specific transcendental function. For example, each of the works [2, 24] shows how to decide a certain kind of linear-transcendental problem using a relatively elementary and explicit approach well suited for implementation. Examples of related work are provided by a line of research initiated by Richardson [20]. This line of work addresses the computation of one and two dimensional polynomial-exponential systems using a variant of Sturm theory [16, 17]. Further examples of related work, in which algorithms are proposed (though not implemented) and relevant complexity bounds derived, can be found in works such as [19].

The present paper is in the spirit of [2, 24] in that its method addresses a certain fragment of the first order theory of the reals with exponentiation, relies on no unproven conjecture, and has been implemented in the computer logic system REDLOG [10]. In particular an unconditional and implementable decision method is presented for prenex sentences (that is, prenex formulas containing no free variables) having bound variables $x_1, x_2, \ldots, x_n$, in which only x_1 occurs as argument of the exponential function. This nontrivial fragment of the extended system of real algebra is thus decided in both theory and practice. Our method and the REDLOG module embodying it thus provide an extension of computer algebra tools for real algebra into real analysis. Further extension to problems with several exponential variables appears difficult while Schanuel's conjecture remains unresolved.

Our decision method is based upon an algorithm for isolating the real zeros of a univariate exponential polynomial $p(x, \exp(x))$ (where $p(x, y)$ is a given bivariate integral polynomial). Our recursive real root isolation algorithm uses pseudodifferentiation and Rolle's theorem in the spirit of [8], and also relies upon a classical result of Lindemann [22] (see Section 2). The root isolation method of [16], in contrast, is based upon the construction of local Sturm sequences. We have reason to believe that our root isolation method is more efficient than that of [16] (see Section 3).

The remainder of the paper is organised as follows. Section 2 presents the formal framework and recalls the most essential background material for reading the paper. In Section 3 we outline an algorithm which decides *univariate* polynomial-exponential problems. This involves a careful study of real zero isolation for exponential polynomials. Section 4 presents a decision procedure for more general polynomial-exponential problems which uses the method of Section 3 as a subalgorithm. Section 5 reports on the REDLOG implementation of the decision procedure of Section 4, and discusses some examples. Section 6 discusses planned extensions of the work reported.

2. FORMAL FRAMEWORK AND BACKGROUND MATERIAL

In this section we introduce exponential polynomials, define the extension of the first order theory of the reals with which we shall be concerned, and specify some properties of the exponential function and exponential polynomials.

2.1 Exponential Polynomials

The symbol exp will denote the exponential function $x \mapsto e^x$ defined and analytic for all $x \in \mathbb{R}$.

Definition 1. ***(exponential polynomial)***
Let $p(x, y)$ be an integral polynomial in x and y, i.e. $p(x, y) \in \mathbb{Z}[x, y]$. Putting $y = \exp(x)$ we obtain the associated **exponential polynomial** $p^*(x) := p(x, \exp(x))$, analytic on the whole real line.

The following result is not difficult to prove.

THEOREM 2.1. *Let $p(x, y)$ be an integral polynomial in x and y, and let $p^*(x) = p(x, \exp(x))$ be the associated exponential polynomial. Then the mapping $p(x, y) \to p^*(x)$ is an injective homomorphism of $\mathbb{Z}[x, y]$ into the ring of all real valued analytic functions defined on the whole real line.*

Definition 2. ***(pseudodegree, pseudoderivative)***
Let $p^*(x) = p(x, \exp(x))$ be the exponential polynomial associated to $p(x, y) = \sum_{i=0}^{n} p_i(x) y^i$. We define the following concepts:

- The **pseudodegree** of $p^*(x) \neq 0$, denoted by pdeg p^*, is the pair $(m, n) \in \mathbb{N}^2 = \mathbb{N} \times \mathbb{N}$, where $m = deg_y\ p(x, y)$ and $n = deg_x\ p(x, 0)$. (If $p(x, 0) = 0$, we set $n = 0$).
- The **pseudoderivative** of $p(x, y)$ is, with a slight abuse of notation, the polynomial $p'(x, y) = \partial p/\partial x + y \partial p/\partial y$ (cf. the notion of false derivative in [16, 20]). Notice that the derivative $(p^*)'$ of p^* thus satisfies $(p^*)'(x) = (p')^*(x)$.

2.2 Extension of the First-Order Theory of the Reals

The *extension of the first-order theory of the real numbers* with which we shall be concerned, denoted by $\mathcal{T}_{\text{exp}}$, is a certain class of true sentences for the structure $\langle \overline{\mathbb{R}}, \exp \rangle$, where $\overline{\mathbb{R}} = \langle \mathbb{R}; +, -, \cdot, 0, 1, < \rangle$ denotes the real ordered field. Sentences of $\mathcal{T}_{\text{exp}}$ are expressed in the language $\mathcal{L}_{\text{exp}}$ in which the variables are $x_1, x_2, \ldots$, the constant symbols are $0, 1$, the binary function symbols are $+, -$ and $\cdot$, the unary function symbol is exp, to be applied *only* to the variable x_1, and the binary order relation symbol is $<$. *Terms* of this language are integral polynomials in the variables $x_1, x_2, \ldots$ and another variable y, where every occurrence of y is replaced by $\exp(x_1)$. By an *atomic formula* we mean an equation, inequation or inequality of the form $\tau = 0$, $\tau \neq 0$, $\tau > 0$, etc. where τ is a term. *Formulas* are constructed from atomic formulas using boolean connectives and quantifiers. A *sentence* is a formula without free variables, usually expressed in prenex form:

$$(Q_1 x_1)(Q_2 x_2)...(Q_n x_n)\psi(x_1, x_2, ...x_n)$$

where ψ is a quantifier-free formula, and the $(Q_i x_i)$ are quantifiers.

EXAMPLES 2.2. **(prenex sentences)***:*

1. $(\forall x_1)(\exists x_2)[(x_1 + x_2) + x_2^2 y \neq x_1 y^2 \vee x_1 - x_2 = 0]$*, where* $y = \exp(x_1)$.
2. $(\exists x_1)[y - x_1 - 1 = 0 \wedge x_1 > 0]$*, where* $y = \exp(x_1)$.

In Sections 3 and 4 we shall present a decision method for $\mathcal{T}_{\text{exp}}$.

2.3 Properties of the Exponential Function and Exponential Polynomials

Here we introduce the concept of an *admissible* function and show that the exponential function is admissible. Moreover, we establish crucial properties of exponential polynomials which we will need in our algorithm that isolates the real zeros of such functions and our algorithm that decides univariate polynomial-exponential problems.

Recall that a complex number is *transcendental* if it is not algebraic. F. Lindemann proved the following important result [22] concerning the transcendence of the values of the exponential function:

THEOREM 2.3. **(Lindemann's theorem)**
If z is a nonzero algebraic number, then e^z is transcendental.

The following concept was introduced in [2]:

Definition 3. ***(strongly transcendental function)***
A real or complex analytic function $f(x)$ is called **strongly transcendental** (with exceptional points $\alpha_1, \alpha_2, ..., \alpha_k$) if for all real or complex numbers x excluding the α_i $(x \neq \alpha_i)$ not both x and f(x) are algebraic.

COROLLARY 2.4. *The complex function e^z is strongly transcendental with exceptional point 0.*

The following concept is intended to capture the requirement on a real analytic, strongly transcendental function $trans(x)$ which makes conceivable the provision of a finite description of the zeros of any polynomial function of $trans(x)$.

Definition 4. ***(admissible function)***
A strongly transcendental real valued function $trans(x)$, with exceptional points $\alpha_1, ..., \alpha_n$, analytic on the whole real line is called an **admissible** function with exceptional points α_i if for every nonzero integral polynomial $p(x, t)$, the associated real analytic function $p^*(x) = p(x, trans(x))$ has only finitely many zeros in the real line.

THEOREM 2.5. *The exponential function* $\exp(x)$ *is admissible with the exceptional point* 0.

PROOF. By Corollary 2.4, the exponential function $\exp(x)$ is strongly transcendental. According to Hardy [[11], page 17] a one-variable exponential function which does not vanish identically has only finitely many real zeros. Hence, for every nonzero $p(x, y) \in \mathbb{Z}[x, y]$, the associated real analytic function $p^*(x)$ has only finitely many real zeros. $\square$

We present some important properties of exponential polynomials.

Lemma 2.6. *Let $p(x, y)$ be an irreducible element of $\mathbb{Z}[x, y]$ of positive degree in y. Let $p'(x, y)$ be the pseudoderivative of $p(x, y)$ (defined in Sec. 2.1). Let $r(x) = res_y(p, p')$. Then $r(x) \neq 0$ unless $p(x, y) = \pm y$ or $p'(x, y) = 0$.*

Proof. Suppose $r(x) = 0$. Then p and p' have a common factor with positive degree in y. Since p is irreducible it follows that $p|p'$. Hence $p(x, 0)|p'(x, 0)$. But $p'(x, 0)$ is the ordinary derivative (with respect to x) of $p(x, 0)$. Hence we must have $p'(x, 0) = 0$. Therefore y is a factor of $p'(x, y)$. Let $q = \frac{p'}{y}$. Since $p|yq$ and p is irreducible, either $p|y$ or $p|q$. In the former case $p = \pm y$. In the latter case, observe that the degree in y of q is less than that of p, hence $q = 0$. Therefore $p' = 0$ also. □

Theorem 2.7. *Let $p(x, y)$ be an irreducible element of $\mathbb{Z}[x, y]$. Then the only possible non-simple real zero α of $p^*(x)$ is the exceptional point of $\exp(x)$ i.e. $\alpha = 0$.*

Proof. p^* has no non-simple zeros if p has degree 0 in y. So suppose that p has positive degree in y. With $p'(x, y)$ denoting the pseudoderivative of $p(x, y)$, let $r(x) = res_y(p, p')$ and suppose that $r(x) = 0$. By the previous lemma, $p(x, y) = \pm y$ or $p'(x, y) = 0$. In the former case, $p^*(x) = \pm \exp(x)$. In the latter case, $p^*(x)$ is a nonzero constant. In both these cases $p^*(x)$ has no real zeros, hence no non-simple real zeros. So henceforth suppose that $r(x) \neq 0$. Let α be a non-simple zero of $p^*(x)$. Then $p^*(\alpha) = (p^*)'(\alpha) = 0$. Hence, with $\beta = \exp(\alpha)$, we have

$$p(\alpha, \beta) = p'(\alpha, \beta) = 0.$$

Therefore α is a root of $r(x) \neq 0$, hence algebraic. Now β is a root of the polynomial $p(\alpha, y)$, which is nonzero by the irreducibility of p. Hence β is also algebraic. By Lindemann's theorem (see Theorem 2.3) α must be an exceptional point of $\exp(x)$ (i.e. $\alpha = 0$). □

Theorem 2.8. *Let $p(x, y)$ and $q(x, y)$ be relatively prime nonzero elements of $\mathbb{Z}[x, y]$. Then the only possible common real zero α of $p^*(x)$ and $q^*(x)$ is the exceptional point of $\exp(x)$ i.e. $\alpha = 0$.*

Proof. $p^*(x)$ and $q^*(x)$ have no common zeros if both p and q have degree 0 in y. So suppose that at least one of p and q has positive degree in y. Let α be a common zero of $p^*(x)$ and $q^*(x)$ which is not the exceptional point of $\exp(x)$, i.e. $\alpha \neq 0$. Then, with $\beta = \exp(\alpha)$, we have

$$p(\alpha, \beta) = q(\alpha, \beta) = 0.$$

Therefore α is a root of the resultant $res_y(p, q)$, which is a nonzero polynomial since p and q are assumed relatively prime. Hence β is also algebraic, since β is a root of the polynomials $p(\alpha, y)$ and $q(\alpha, y)$, at least one of which is nonzero by relative primality. But this contradicts Lindemann's theorem (see Theorem 2.3), which implies that not both α and β can be algebraic. □

Corollary 2.9. *Let $p(x, y) \in \mathbb{Z}[x, y]$ be nonzero and squarefree. Then the only possible non-simple real zero α of $p^*(x)$ is the exceptional point of $\exp(x)$ i.e. $\alpha = 0$.*

Proof. Since $p(x, y)$ is squarefree it is a product of pairwise relatively prime irreducible elements of $\mathbb{Z}[x, y]$. The corollary follows immediately by application of the two preceding theorems. □

3. DECIDING UNIVARIATE POLYNOMIAL-EXPONENTIAL PROBLEMS

The goal of this section is to present a decision method for those sentences of the theory $\mathcal{T}_{\exp}$ defined in Sec. 2 which involve only the variable x_1. Indeed, we shall describe an algorithm DUPEP that decides univariate polynomial-exponential problems. In this section, for simplicity, we use the variable x instead of x_1. The section is organized as follows. First we consider the problem of determining bounds for the real zeros of exponential polynomials. Next we present a key subalgorithm ISOL that isolates the real zeros of an exponential polynomial. Finally, a full description of the algorithm DUPEP is given.

3.1 Real Zero Bound for Exponential Polynomials

Our real zero isolation algorithm ISOL will require the determination of a real zero bound for exponential polynomials.

Theorem 3.1. (**upper bound for real zeros**) *Let $p(x, y) = \sum_{i=0}^{n} p_i(x)y^i$ with $p_i(x) \in \mathbb{Z}[x]$ and $p_n(x) \neq 0$. Then an upper bound C for the real zeros of $p^*(x)$ can be obtained with the following procedure:*

1. *Find $C_1 > 0$ such that for all $x > C_1$, $|p_n(x)| \geq 1$.*
2. *Find $C_2 > 0$ and $k \in \mathbb{N}$ such that for all i in the range $0 \leq i < n$ and for all $x > C_2$, $|p_i(x)| \leq \frac{x^k}{n}$.*
3. *Find $C_3 > 0$ such that for all $x > C_3$, $x^k < \exp(\frac{x}{2})$.*
4. *Set $C \leftarrow \max\{C_1, C_2, C_3\}$.*

Proof. Let $x > C$. Then we can derive the inequality

$$|\sum_{i=0}^{n-1} p_i(x) \exp(x)^i| < |p_n(x)|e^{nx}$$

by applying 1, 2 and 3. Therefore, for $x > C$, we have $p^*(x) \neq 0$. Thus C is an upper bound for the real zeros of $p^*(x)$. Complete details concerning the determination of the numbers C_1, C_2, C_3 can be found in [1]. □

A lower bound for the real zeros of $p^*(x)$ can be obtained by applying an analogous procedure to the exponential polynomial $g^*(y) = e^{ny}p^*(-y)$. The details can be found in [1]. See [16] for an alternative method for finding a real zero bound for exponential polynomials.

3.2 Isolating Real Zeros of Exponential Polynomials

We shall describe an algorithm to isolate the real zeros of a nonzero exponential polynomial $p^*(x) = p(x, \exp(x))$. This algorithm is based on differentiation and recursion on the pseudodegree pdeg p^* of $p^*(x) \neq 0$. (Recall that the pseudodegree of $p^*(x)$ was defined in Section 2.1.) Related algorithms for polynomial real root isolation can be found in [9, 13]. We shall use the lexicographic order $\leq$ on $\mathbb{N}^2$ (defined, for example, in Exercise 4.61 of [4]): $(k, l) \leq (m, n)$ means that $k < m$ or ($k = m$ and $l < n$) or ($k = m$ and $l = n$). Theorem 4.62 of [4] implies that this linear (indeed admissible) order on $\mathbb{N}^2$ is a *well-order*, that is, every nonempty subset S of $\mathbb{N}^2$ has a least element. Hence the principle of *noetherian induction* can be used to prove a claim of the kind, “$P(m,n)$ is true for all $(m,n)\in\mathbb{N}^2$”, as explained in [4]. Our real zero isolation algorithm uses recursion on the pseudodegree, and we will demonstrate its validity using noetherian induction.

Our algorithm uses the concept of a *modulus of continuity (moc)* [5, 6] for a real valued function $f(x)$ on a nonempty compact interval $[a,b]$ of the real line. A positive real valued function δ defined on the set of all positive real numbers is called a *moc* for $f(x)$ on $[a,b]$ if for all $\epsilon>0$ and for all $x,y\in[a,b]$ $|x-y|\le\delta(\epsilon)$ implies $|f(x)-f(y)|\le\epsilon$. By definition $f(x)$ is uniformly continuous on $[a,b]$ if and only if there is a moc for $f(x)$ on $[a,b]$. Moreover by a standard theorem of analysis the assertion of the preceding sentence remains valid if one omits the modifier ‘uniformly’. The following theorem provides an explicit linear moc for a continuously differentiable function.

THEOREM 3.2. *Let $f(x)$ be real valued and continuously differentiable on $I=[a,b]$. Let M be a positive number with $M\ge\max_{x\in I}|f'(x)|$. Then a linear moc δ for $f(x)$ on I can be obtained by putting $\delta(\epsilon)=\epsilon/M$.*

PROOF. Let $\epsilon>0$, let $x,y\in I$, and suppose that $|x-y|\le\delta(\epsilon)$. By the mean value theorem, $f(x)-f(y)=f'(c)(x-y)$, for some c between x and y. Therefore

$$|f(x)-f(y)|=|f'(c)||x-y|\le M|x-y|\le M\delta(\epsilon)=M\epsilon/M=\epsilon.$$

This completes the proof. □

We define some terminology. An *isolation list* for a real-valued function $f(x)$ defined on the whole real line is a list $L=(I_1,I_2,\ldots,I_k)$, such that

(a) k is the number of distinct real zeros of f;

(b) each $I_j=(a_j,b_j)$, where a_j and b_j are binary rational numbers;

(c) each I_j contains a unique zero of f; and

(d) $a_1<b_1\le a_2<b_2\le\ldots\le a_k<b_k$.

ALGORITHM 3.3. **(real root isolation of exponential polynomials)**

$$L\leftarrow \mathrm{ISOL}(p)$$

Input: $p(x,y)\in\mathbb{Z}[x,y]$, a nonzero squarefree integral polynomial.
Output: L, an isolation list for $p^*(x)=p(x,\exp(x))$.

(1) [Basis.] Set $(m,n)\leftarrow$ pdeg $p^*(x)$. If $(m,n)=(0,0)$ then {Set $L\leftarrow()$. Return}.

(2) [Recursion.] Compute $p'(x,y)$, the pseudoderivative of $p(x,y)$. If $n>0$ then set $s(x,y)\leftarrow\mathrm{gsfd}(p'(x,y))$ else {Set $\hat{p}(x,y)\leftarrow p'(x,y)/y$. Set $s(x,y)\leftarrow\mathrm{gsfd}(\hat{p}(x,y))$}. Set $L'\leftarrow\mathrm{ISOL}(s)$. [In this step ‘gsfd’ denotes ‘greatest squarefree divisor’. By construction, pdeg $s(x,y)<$ pdeg $p(x,y)$. By noetherian induction hypothesis, L' is an isolation list for $s^*(x)$, hence for $(p')^*(x)=(p^*)'(x)$.]

(3) [Bound for real zeros.] Set $\gamma\leftarrow$ a binary rational real zero bound for $p^*(x)$, using the method of Section 3.1. [Every real zero α of $p^*(x)$ satisfies $|\alpha|<\gamma$.]

(4) [Prepare for induction step.] Let $L'=(I_1,I_2,\ldots,I_k)$, with $I_j=(a_j,b_j)$. Let $\alpha'_0=-\infty$ and $\alpha'_{k+1}=\infty$, and for $1\le j\le k$, let α'_j denote the unique zero of $(p^*)'(x)$ in I_j. Observe that, by Corollary 2.9, $s^*(a_j)s^*(b_j)<0$, for every j in the range $1\le j\le k$, unless $\alpha'_j=0$. Without loss of generality assume that, in case $k>0$, $-\gamma\le a_1$ and $b_k\le\gamma$. Put $b_0=-\gamma$ and $a_{k+1}=\gamma$. For $i=0,1,\ldots,k$, put $J_i=(b_i,a_{i+1})$. [In case $b_i=a_{i+1}$ J_i is empty.] [By Rolle's theorem, each interval $[\alpha'_i,\alpha'_{i+1}]$ contains at most one zero of $p^*(x)$. Hence each complementary interval $J_i=(b_i,a_{i+1})$ contains at most one such zero. Moreover, by Corollary 2.9, $p^*(x)$ has no non-simple zeros, with the possible exception of 0. Hence neither α'_i nor α'_{i+1} is a zero of $p^*(x)$, unless $\alpha'_i=0$ or $\alpha'_{i+1}=0$. The next step will ensure that, after suitable refinement of the I_js, no $[a_j,b_j]$ contains a zero of $p^*(x)$, unless $\alpha'_j=0$.] Set $L\leftarrow()$.

(5) [Interval refinement.] For $j=1,2,\ldots,k$ {If $\alpha'_j=0$ and $p^*(0)=0$ then insert I_j into L else {Compute a linear moc δ for $p^*(x)$ on the (initial) interval $[a_j,b_j]$. Repeatedly bisect $I_j=(a_j,b_j)$, always retaining the subinterval of I_j which contains α'_j, that is, always maintaining the invariance of the relation $s^*(a_j)s^*(b_j)<0$. But if $s^*(m_j)=0$, with $m_j=(a_j+b_j)/2$, then retain the subinterval $(a_j+(b_j-a_j)/4, a_j+3(b_j-a_j)/4)$ centred at $m_j=\alpha'_j$. Where $\epsilon=\min(|p^*(a_j)|,|p^*(b_j)|)/2$, terminate the bisection process when $\epsilon>0$ and (then) $b_j-a_j\le\delta(\epsilon)$.} } [For each j the repeated bisection process must terminate. For suppose that this is not the case. Then the repeated bisection process defines infinite sequences of values for a_j and b_j, hence for ϵ. Observe $p^*(\alpha'_j)\neq 0$, by Corollary 2.9. Now the sequence of values of ϵ tends to the limit $|p^*(\alpha'_j)|/2$, which is positive by the above observation. Hence the sequence of values of $\delta(\epsilon)$ tends to the limit $\delta(|p^*(\alpha'_j)|/2)>0$ (as δ is a linear function). Hence eventually $\epsilon>0$ and $b_j-a_j\le\delta(\epsilon)$, contradicting the assumption about non-termination. Now $|b_j-\alpha'_j|<b_j-a_j$. Hence upon termination $|p^*(b_j)-p^*(\alpha'_j)|\le\epsilon<|p^*(b_j)|$. Therefore $\mathrm{sign}(p^*(\alpha'_j))=\mathrm{sign}(p^*(b_j))\neq 0$. Hence $[\alpha'_j,b_j]$ contains no zero of $p^*(x)$. We can similarly show that $[a_j,\alpha'_j]$ contains no zero of $p^*(x)$. Combining two of our conclusions we see that $[a_j,b_j]$ contains no zero of $p^*(x)$.]

(6) [Completion of induction step.] For $i=0,1,\ldots,k$ {If $p^*(b_i)p^*(a_{i+1})<0$ then insert J_i into L }. [By step 5, $(\alpha'_i,\alpha'_{i+1})$ contains a zero of $p^*(x)$ if and only if J_i does, which occurs if and only if $p^*(b_i)p^*(a_{i+1})<0$.] Return.

Computation of a linear moc δ for $p^*(x)$ on $[a_j,b_j]$, as required by step 5 of the above algorithm, could be done by applying Theorem 3.2. (To apply this theorem here one has to obtain a bound M for $|(p^*)'(x)|$ on $[a_j,b_j]$. Such a bound could be obtained using the triangle inequality and appropriate estimates for the component terms of $|(p^*)'(x)|$ on $[a_j,b_j]$.) The astute reader will note that ISOL requires a method to evaluate the sign of an exponential polynomial $s^*(x)$ at a binary rational number, say r, in case $s^*(r)\neq 0$. Such a method could be readily obtained by iterating sufficiently often a standard numerical procedure for computing an interval $[u, w]$ of specified length $\epsilon > 0$ guaranteed to contain the value $v = s^*(r)$.

For the record, we assert the validity of our algorithm.

THEOREM 3.4. *For all $(m, n) \in \mathbb{N}^2$, the following statement $P(m, n)$ is true: for every valid input $p(x, y)$, with* $\mathrm{pdeg}\, p = (m, n)$*, ISOL returns an isolation list L for $p^*(x) = p(x, \exp(x))$.*

PROOF. The main ingredients of the proof of this theorem by noetherian induction on (m, n) are supplied as comments in the above description of ISOL. □

For ease of exposition and proof we have kept our description of ISOL conceptually simple. In practice there are elementary improvements which could be made to enhance the efficiency of the method. For example, we could insert an initial algorithm step (step 0, say) which finds the content $c(x)$ of $p(x, y)$ with respect to y and computes an isolation list L_0 for $c(x)$ using any highly efficient real root isolation algorithm for $\mathbb{Z}[x]$ [9, 13]. After setting $p(x, y) \leftarrow p(x, y)/c(x)$ we proceed with steps 1-6 (in which step 5 could be simplified slightly, by Theorem 3.5 below). We could then append a final algorithm step (step 7, say) which refines the isolating intervals in $L \leftarrow L_0 \cup L$ into an isolation list for the original $p^*(x)$.

Some comparison of ISOL with the root isolation method of [16] is warranted. Actually [16] is chiefly concerned with counting the number of real roots of a given exponential polynomial $p^*(x)$ in some interval (a, b). The method is based on construction of local Sturm sequences. It requires computation of the real roots of two auxiliary univariate polynomials whose degrees are high relative to $\deg_x p$ and $\deg_y p$. It also involves evaluation of the signs of a list of exponential polynomials at rational points. So, although no systematic empirical comparison of the two root isolation algorithms has yet been undertaken, we think that our method is more efficient than that of [16].

3.3 Sample Points

Let $f^*(x)$ be the exponential polynomial associated to some given $f(x, y) \in \mathbb{Z}[x, y]$. We make no assumption about the squarefreeness of $f(x, y)$, so we cannot directly apply algorithm ISOL to isolate the zeros of $f^*(x)$. Let us nevertheless consider the zeros of $f^*(x)$, which determine a decomposition of the real line, which we term an *f^*-invariant* decomposition. Assuming that $f^*(x)$ has n zeros, then the decomposition consists of n 0-cells (the zeros) and $n+1$ 1-cells (the open intervals between the zeros). A *sample point* for a cell is an exact representation of a particular algebraic or non-algebraic (i.e. transcendental) number belonging to that cell. Suppose that we could somehow obtain an isolation list for $f^*(x)$. (A method for doing so, which uses the algorithm ISOL, will be described in the next subsection.) Then as sample points for the 1-cells we use appropriately chosen rational endpoints from the isolating intervals obtained. If a 0-cell α is an algebraic number we use a standard representation for α by its minimal polynomial and an isolating interval, as described in [14]. For a 0-cell α which is not an algebraic number we represent α by the irreducible factor $p(x, y)$ of $f(x, y)$ for which $p^*(\alpha) = 0$ and an isolating interval for α.

We now describe how to determine the sign of a given exponential polynomial $g^*(x)$ at a given sample point $\alpha \neq 0$ of such an f^*-invariant decomposition of the real line. Such sign determination will be an important component of the algorithm DUPEP, described in the next subsection.

Suppose first that α is algebraic. For determining whether or not $g^*(\alpha) = 0$ the following theorem is relevant.

THEOREM 3.5. *Let $h(x, y) \in \mathbb{Z}[x, y]$ be primitive and of positive degree in y. Let $\alpha \neq 0$ be an algebraic number. Then $h^*(\alpha) \neq 0$.*

PROOF. Suppose that $h^*(\alpha) = 0$. Let $\beta = \exp(\alpha)$. Then $h(\alpha, \beta) = h^*(\alpha) = 0$. But $h(\alpha, y) \neq 0$, by the primitivity of $h(x, y)$. Therefore β is algebraic, contradicting Theorem 2.3. □

We find the content $c(x)$ and the primitive part $h(x, y)$ of $g(x, y)$. Then $g^*(\alpha) = 0$ if and only if $c(\alpha) = 0$, by the above theorem. Suppose now that $g^*(\alpha) \neq 0$. Let I be an isolating interval for α. Using a method analogous to that of step 5 of ISOL, one refines I about α so that, after refinement, I contains no zero of $g^*(x)$. Then, by evaluating g^* at the left or right endpoint of the refined I, we can determine the sign of $g^*(\alpha)$.

Suppose on the other hand that α is not algebraic. Recall that α is represented by the irreducible factor $p(x, y)$ of $f(x, y)$ for which $p^*(\alpha) = 0$ and an isolating interval say I for α. Clearly $g^*(\alpha) = 0$ if p is a factor of g. So suppose that p is not a factor of g. Then, by Theorem 2.8, $g^*(\alpha) \neq 0$. Using a method analogous to that of step 5 of ISOL, one refines I about α so that, after refinement, I contains no zero of $g^*(x)$. Then, by evaluating g^* at the left or right endpoint of the refined I, we can determine the sign of $g^*(\alpha)$.

3.4 The Algorithm DUPEP

ALGORITHM 3.6. **(deciding univariate polynomial-exponential problems)**

$$v \leftarrow \mathrm{DUPEP}(\varphi)$$

Input: A prenex sentence φ in $\mathcal{L}_{\exp}$ involving only x.
Output: The truth value v of φ over the reals.

1. **Extraction.** The input φ is of the following form
$$(Qx)\psi(x)$$
where ψ is a quantifier-free formula of $\mathcal{L}_{\exp}$ and (Qx) is a quantifier. Extract the list $P := \{p_1, p_2, \ldots, p_n\}$ of those polynomials $p_i(x, y) \in \mathbb{Z}[x, y]$ for which $p_i^*(x)$ occurs in $\psi(x)$.

2. **Contents and primitive parts.** Compute the set $cont_y(P)$ of contents (w.r.t. y) of the elements of P and the set $pp_y(P)$ of primitive parts (w.r.t. y) of elements of P of positive degree in y.

3. **Squarefree bases.** Compute squarefree bases K and Q of $cont_y(P)$ and $pp_y(P)$, respectively.

4. **Root isolation.** Apply algorithm ISOL to each polynomial $q(x, y)$ in Q and to each polynomial $c(x)$ in K, individually.

5. **Isolation list for product.** By their relative primality, for any pair of distinct elements p and q of $K \cup Q$, p^* and q^* have common real zeros only at $x = 0$ (see Theorem 2.8). Hence by refining the isolating intervals for the zeros of all the $p^*(x) \in K \cup Q^*$ we obtain an isolation list for the product $f^*(x)$ of all $p^*(x)$.

6. **Sample points.** Use the isolation list for $f^*(x)$ to construct sample points for all of the cells of the decomposition of the real line determined by the zeros of $f^*(x)$, as described in the previous subsection.

7. **Evaluation.** Use the sample points to decide the original function $(Qx)\psi(x)$. [This can be done as follows. For each i express $p_i(x, y)$ as a product of elements of $K \cup Q$. By evaluating the signs of the exponential polynomials associated with the factors of each p_i at each sample point as described in the previous subsection, the sign of each p_i^* at each sample point can be determined. Then the truth value of the original formula φ can be decided.]

4. DECIDING MULTIVARIATE POLYNOMIAL-EXPONENTIAL PROBLEMS

In Section 3 we described an algorithm DUPEP that decides univariate polynomial-exponential problems. In this section we outline an extention to this procedure, i.e. an algorithm that decides polynomial-exponential problems in general.

Our decision algorithm DPEP accepts as input a prenex sentence φ in $\mathcal{L}_{\text{exp}}$. It produces as output the truth value v of the input φ over the real numbers.

The algorithm DPEP has two basic phases.

(1) **First Phase.** Now the input prenex sentence φ has the form

$$(Q_1x_1)(Q_2x_2)\dots(Q_nx_n)\psi(x_1, x_2, \dots, x_n)$$

where ψ is a quantifier-free formula of $\mathcal{L}_{\text{exp}}$ and the (Q_ix_i) are quantifiers. Recall that ψ involves polynomials in the x_i and $\exp(x_1)$. We apply a quantifier-elimination (QE) algorithm for elementary real algebra (such as QE by CAD [3, 7]) to the following formula

$$\varphi' = (Q_2x_2)(Q_3x_3)\dots(Q_nx_n)\psi(x_1, x_2, \dots, x_n)$$

obtained by removing (Q_1x_1) from φ. (As the variable x_1 is not quantified in φ', the quantifier elimination algorithm can proceed without any precautions concerning the exponential function.) The output is a quantifier-free formula $\psi_1(x_1)$ which is equivalent to φ'.

(2) **Second Phase.** We combine the quantifier (Q_1x_1) of the input sentence φ with the output of the QE algorithm $\psi_1(x_1)$, thus obtaining a univariate polynomial-exponential decision problem instance:

$$\varphi'' = (Q_1x_1)\psi_1(x_1).$$

We apply algorithm DUPEP from Section 3 to this problem instance φ'', obtaining the truth value v of φ'' over the real numbers. Finally we return v as the truth value of the input sentence φ over the real numbers.

The validity of algorithm DPEP is clear.

5. IMPLEMENTATION AND EXAMPLES

The univariate decision procedure DUPEP described in Section 3 was implemented by the first author [1] in the REDLOG package of the computer algebra system REDUCE [12] under the guidance of the third author. Based on the package for quantifier elimination by cylindrical algebraic decomposition (QE by CAD) in REDLOG [21] this module was extended to an implementation of the more general decision procedure DPEP described in Section 4.

The functions implemented use the logical REDLOG context OFSF for the ordered field of real numbers regarded as a structure for the first-order language of ordered rings [10]. In order to avoid a cumbersome extension of this context for the handling of polynomial-exponential problems the following conventions were used: the variable x_1 is reserved as independent variable of the exponential function; and the value $\exp(x_1)$ of this function is represented by a new free variable y. Sentences φ of the formal language $\mathcal{L}_{\text{exp}}$ are entered into the decision program DPEP in prenex form

$$(Q_1x_1)(Q_2x_2)\dots(Q_nx_n)\psi(x_1, y, x_2, \dots, x_n),$$

where the (Q_ix_i) are quantifiers. Semantically the variables x_i range over the real numbers, the variable y is treated as the value $\exp(x_1)$, and the truth value of φ is evaluated accordingly in the ordered field of real numbers with exponentiation.

In REDLOG the user has the following commands to start, and to obtain verbose output of, the decision procedure: `rldpep`, `rldpepverbose`, and `rldpepiverbose`. The second command turns on the verbose output option. The third command provides an even more detailed trace of the procedure's workings.

The implementation of DPEP was tested on several examples with up to 3 quantifiers and produced correct results with running times in the range of 0.5 s to 12 s on a Pentium 4 (2 Ghz, 128 heap size). Here are a few of the examples used for testing purposes. In each example the correct truth value (true) was obtained in less than a second on the Pentium 4.

$$(\exists x_1)(\exists x_2)[y - x_2^2 = 0 \land x_1 - x_2 = 0], \text{ where } y = \exp(x_1)$$
$$(\forall x_1)[(1 - x_1) \cdot y \leq 1 \lor x_1 \geq 1], \text{ where } y = \exp(x_1)$$
$$(\exists x_1)(\exists x_2)(\exists x_3)[2x_1 - x_2 + x_3 + y^2 = 0 \land 3x_2 - x_3 = 0 \land 2x_1 + x_2 + 3x_3 + y = 0], \text{ where } y = \exp(x_1)$$

Sentences involving polynomials in $x_1, x_2, \dots, x_n$ and a hyperbolic function such as $\cosh(x_1)$ can be massaged into a form for which the program DPEP can be applied. For example suppose we want to decide

$$(\forall x_1)[x_1 > 7 \Rightarrow \cosh(x_1) > p(x_1)]$$

for some integral polynomial $p(x_1)$. Using the defining relation

$$\cosh(x_1) = \frac{\exp(x_1) + \exp(-x_1)}{2}$$

and noting that $\exp(x_1)$ is positive on the real line, we see that the sentence of interest is equivalent to the sentence

$$(\forall x_1)[x_1 > 7 \Rightarrow \exp(x_1)^2 + 1 > 2p(x_1)\exp(x_1)]$$

of $\mathcal{L}_{\text{exp}}$. As a specific example we set $p(x_1) = x_1^3 - 4x_1$. The program DPEP found that the above sentence is true within 17 seconds.

Sentences involving polynomials in x and a Gaussian function such as $\exp(-x^2)$ can also be massaged into sentences of $\mathcal{L}_{\text{exp}}$. For example suppose we want to decide

$$(\exists x)[\exp(-x^2) = p(x)]$$

for some integral polynomial $p(x)$. We introduce a new variable say z to represent $-x^2$ and thus obtain an equivalent sentence:

$$(\exists z)(\exists x)[\exp(z) = p(x) \wedge z + x^2 = 0].$$

Replacing z by x_1 and x by x_2 we obtain an equivalent sentence of $\mathcal{L}_{\exp}$. In a similar manner sentences involving polynomials in x and a more general exponential function of the form $\exp(\pi(x))$, where $\pi(x)$ is a rational polynomial, can be converted into sentences of $\mathcal{L}_{\exp}$.

6. REFINEMENTS AND EXTENSIONS

A refinement of algorithm DPEP on efficiency grounds is contemplated. It would probably be desirable to try to develop a decision method for $\mathcal{T}_{\exp}$ which is analogous to decision by cylindrical algebraic decomposition (CAD). In order to decide a sentence φ in the variables $x_1, x_2, \ldots, x_n$ we would first extract the set A of polynomials occurring in φ. Second, by analogy with the CAD method, we would compute a description of a cylindrical *analytic* decomposition D of the space $\mathbb{R}^n$ compatible with the zeros of the polynomials in A. Third, we would use the description of D to decide the truth or falsity of φ by analogy with QE by CAD. Some progress has already been made in this direction [18].

A small extension of the work reported herein is planned. We will endeavour to extend our decision procedure for $\mathcal{T}_{\exp}$ to decide sentences involving polynomials in x and a more general exponential function $\exp(\pi(x))$ (as described at the end of the previous section) *without* introducing a new variable. Some progress has already been made towards this goal [18]. We also plan to investigate the decidability of analogous theories obtained by introducing into elementary algebra other admissible functions such as the inverse trigonometric function $\arctan(x)$. Furthermore, while the trigonometric functions $\sin(x)$ and $\cos(x)$ are technically not admissible we have some optimism that similar results could be obtained for these functions.

Extension of our decision method to the full first order theory of the reals with exponentiation would of course be highly desirable. However this is probably difficult to achieve while Schanuel's conjecture remains unresolved.

7. REFERENCES

[1] M. Achatz. Deciding polynomial-exponential problems. Diploma thesis, Universität Passau, D-94030 Passau, Germany, Sept. 2006.

[2] H. Anai and V. Weispfenning. Deciding linear-trigonometric problems. In C. Traverso, editor, *ISSAC'2000*, pages 14–22. ACM-Press, 2000.

[3] D. S. Arnon, G. E. Collins, and S. McCallum. Cylindrical algebraic decomposition i: Basic algorithm. In B. Caviness and J. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, Texts and Monographs in Symbolic Computation, pages 136–151. Springer, Wien, New York, 1998.

[4] T. Becker, V. Weispfenning, and H. Kredel. *Gröbner Bases, a Computational Approach to Commutative Algebra*, volume 141 of *Graduate Texts in Mathematics*. Springer, New York, corrected second printing edition, 1998.

[5] E. Bishop. *Foundations of Constructive Analysis*. McGraw-Hill, New York, 1967.

[6] E. Bishop and D. Bridges. *Constructive Analysis*. Grundlehren der math. Wissenschaften. Springer-Verlag, Berlin, 1985.

[7] G. E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In B. Caviness and J. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, Texts and Monographs in Symbolic Computation, pages 85–121. Springer, Wien, New York, 1998.

[8] G. E. Collins and R. Loos. Polynomial real root isolation by differentiation. In *Proceedings of the 1976 Symposium on Symbolic and Algebraic Computation SYMSAC'76*, pages 15–25. ACM, 1976.

[9] G. E. Collins and R. Loos. Real zeros of polynomials. In B. Buchberger, G. E. Collins, R. Loos, and R. Albrecht, editors, *Computer Algebra: Symbolic and Algebraic Manipulation*, pages 83–94. Springer-Verlag, Wien, New York, second edition, 1982.

[10] A. Dolzmann and T. Sturm. Redlog user manual, edition 3.0. Technical report, University of Passau, 2004.

[11] G. H. Hardy. *Orders of Infinity*. Cambridge Univ. Press, Cambridge, 1910.

[12] A. Hearn. Reduce user's manual for version 3.8. Technical report, RAND Corporation, 2004.

[13] J. R. Johnson. Algorithms for polynomial real root isolation. In B. Caviness and J. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, Texts and Monographs in Symbolic Computation, pages 269–299. Springer, Wien, New York, 1998.

[14] R. G. K. Loos. Computing in algebraic extensions. In B. Buchberger, G. Collins, and R. Loos, editors, *Computer Algebra: Symbolic and Algebraic Computation*, pages 173–188. Springer, Wien, New York, 1982.

[15] A. Macintyre and A. Wilkie. On the decidability of the real exponential field. In *Kreiseliana: About and around Georg Kreisel*, pages 441–467. A.K. Peters, 1996.

[16] A. Maignan. Solving one and two-dimensional exponential polynomial systems. In O. Gloor, editor, *Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation*, pages 215–221, New York, Aug. 1998.

[17] A. Maignan. Real solving of elementary-algebraic systems. *Numerical Algorithms*, 27:153–167, 2001.

[18] S. McCallum and V. Weispfenning. Deciding polynomial-exponential problems. Technical report, Macquarie University, 2006.

[19] S. Pericleous and N. Vorobjov. New complexity bounds for cylindrical decompositions of sub-pfaffian sets. In B. Mourrain, editor, *Proceedings of the 2001 International Symposium on Symbolic and Algebraic Computation*, pages 268–275, New York, July 2001.

[20] D. Richardson. Towards computing non algebraic cylindrical decompositions. In S. M. Watt, editor, *Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation*, pages 247–255, Bonn, Germany, July 1991.

[21] A. Seidl. Cylindrical decomposition under application-oriented paradigms. Doctoral dissertation, Universität Passau, D-94030 Passau, Germany, Nov. 2006.

[22] A. B. Shidlovskii. *Transcendental Numbers*. Walter de Gruyter, Berlin, New York, 1989.

[23] A. Tarski. A decision algorithm for elementary algebra and geometry. In B. Caviness and J. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, Texts and Monographs in Symbolic Computation, pages 24–84. Springer, Wien, New York, 1998.

[24] V. Weispfenning. Deciding linear-transcendental problems. In V. Ganzha, E. Mayr, and E. Vorozhtshov, editors, *Computer Algebra in Scientific Computation - CASC 2000*, pages 423–438. Springer, 2000.