ALGEBRAIC CELL DECOMPOSITION IN NC
(preliminary version)

Dexter Kozen[1]
Computer Science Department
Upson Hall
Cornell University
Ithaca, New York 14853

Chee-Keng Yap
Courant Institute of Mathematical Sciences
New York University
251 Mercer Street
New York, New York 10012

Abstract. We give an algorithm to construct a cell decomposition of $\mathbb{R}^d$, including adjacency information, defined by any given set of rational polynomials in d variables. The algorithm runs in single exponential parallel time, and in *NC* for fixed d. The algorithm extends a recent algorithm of Ben-Or, Kozen, and Reif for deciding the theory of real closed fields.

1. Introduction

The theory of real closed fields, or the first-order theory of the real numbers with $+$, $\cdot$, and $=$, is one of the most important logical theories in computer science. It is expressive enough to encode computational problems in several areas: computational geometry [YY], robotics and motion-planning [Y,SS], algebraic geometry and algebraic topology [AM,SS], computer graphics [A3], etc. Thus, efficient decision procedures for this theory can have significant impact. Beginning with Alfred Tarski, alternative or increasingly efficient decision procedures have been proposed by Seidenberg, Cohen, Böge, L. Monk [Mo1], Monk and Solovay [Mo2], Collins [Co], and Ben-Or, Kozen and Reif [BKR]. The algorithm of [BKR], hereafter referred to as BKR, runs in single-exponential parallel time or sequential space, and *NC* (polylog time and polynomially many processors) if the number of variables is fixed; this is the best complexity bound to date. Collins' procedure runs in double-exponential time and space, and has been implemented [A2].

Unfortunately, BKR does not provide some critical pieces of topological information that are useful in certain applications. Collins' algorithm provides some, but not all, of this extra information. Our purpose in this paper is to describe how to extend BKR to provide all the extra topological information needed for the applications we have in mind, without any essential increase in complexity over that of BKR.

Given a set Σ of polynomials with rational coefficients in d variables, a *sign assignment* is a map $\sigma:\Sigma \to \{-1, 0, +1\}$. Each sign assignment σ represents an equivalence class

$$X_\sigma = \{\bar{x} \in \mathbb{R}^d \mid sign(p(\bar{x})) = \sigma(p),\ p \in \Sigma\},$$

called the *sign class* of σ. The sign assignment σ is said to be *consistent* if X_σ is nonempty.

BKR's applicability is limited by the fact that it computes only the consistent sign assignments to a given set Σ of polynomials, representing the sign classes X_σ. This is sufficient for its purposes. In general, however, the X_σ are disconnected,[2] and it may be necessary for certain applications to distinguish between the connected components.

Collins' algorithm does distinguish between connected components. Given Σ, it decomposes $\mathbb{R}^d$ into simply connected regions called *cells* (see Section 4 for a formal definition), such that each connected component of X_σ is a union of cells. Unfortunately, Collins' procedure does not provide the adjacency information necessary to paste the cells together to give the connected components of the X_σ, and it is nontrivial to see how to extend it (see Section 2).

In this paper, we extend BKR to provide a cell decomposition a la Collins, with the extra adjacency information necessary to construct the

[1] Work done at the IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598

[2] By a well-known theorem of Milnor, the number of connected components is at most single exponential in the (generalized) size of Σ.

0272-5428/85/0000/0515$01.00 © 1985 IEEE

connected components of X_σ. Moreover, the algorithm is of roughly the same complexity as BKR, namely single-exponential parallel time on a standard model of parallel computation, and *NC* (poly-log time and polynomially many processors) for fixed dimension d. Our new algorithm contains elements from Collins' algorithm as well as crucial ideas from BKR. On the other hand, it constitutes a significant strengthening of both.

2. Previous work on adjacency computations

Several attempts have been made to extend Collins' algorithm to describe adjacency of cells. Several special cases have appeared in the literature:

• In the special case where the polynomials Σ are linear, so that the variety of Σ is a set of hyperplanes in $\mathbb{R}^d$, the adjacency problem becomes the so-called *hyperplane arrangement problem*. Edelsbrunner, O'Rourke and Seidel [EOS] have recently provided an optimal $O(n^d)$ time and space algorithm, where n is the number of hyperplanes. The importance of this special case is underscored by the notion of *generic queries* introduced by Yao and Yao [YY], which essentially reduces many geometric retrieval problems to the case of hyperplane queries.

• Schwartz and Sharir [SS] provided a relatively efficient method that computes the adjacency relation only for those pairs of cells of which one has full dimension and the other has co-dimension 1. For fixed d. their method takes polynomial time. For general adjacencies, they present a complicated method whose efficiency is yet to be analyzed. In fixed dimension, a polynomial time adjacency algorithm for arbitrary pairs of cells follows from an observation of Arnon (see [Y]); we will exploit this observation again in this paper.

• In dimensions 2 and 3, adjacency algorithms have been provided by S. McCallum [McC], and by Arnon, Collins, and McCallum [ACM2]. The latter algorithm has been implemented [A2].

3. Applications

There are many applications of our algorithm: Hilbert's 16th problem, which asks for the classification of all topological types of curves in the real projective plane (see [AM]); computing the singular homology groups of real algebraic varieties [SS]; computer graphics and display [A3]; geometric retrieval problems [CY,YY]. In the last, our algorithm reduces the preprocessing complexity to *NC*.

One particularly important application is in motion planning in robotics. Motion planning can be reduced to searching a graph whose nodes represent the connected components of the sign classes X_σ and whose edges connect adjacent components of distinct sign classes [Y]. In [SS] an efficient but limited adjacency algorithm was given, which could only be applied to a restricted class of motion planning problems. Our algorithm can be used to build a search structure of optimal size (single exponential) for the general motion planning problem.

4. Problem formulation

Let Σ be a set of polynomials in d variables with rational coefficients. *Sign assignments* $\sigma:\Sigma\to\{-1, 0, +1\}$, *consistent sign assignments*, and *sign classes* X_σ were defined in Section 1.

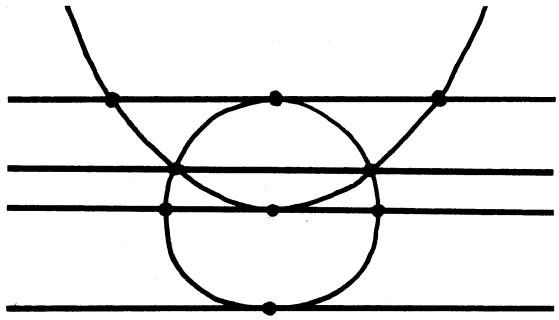
A *cell complex* for Σ is a partition of $\mathbb{R}^d$ into finitely many pairwise disjoint regions R_v, $v \in V$, called *cells*, such that:

(1) each cell R_v is homeomorphic to $\mathbb{R}^{\delta(v)}$ for some $0 \le \delta(v) \le d$;
(2) the closure of R_v is a union of cells R_u;
(3) each R_v is contained in some sign class X_σ; in other words, the sign of each $p \in \Sigma$ is invariant on each R_v.

$\delta(v)$ is called the *dimension* of the cell R_v, and R_v is called a *$\delta(v)$-cell*.

When $d = 1$, a cell complex partitions the real line into an alternating sequence of points (0-cells) and open intervals (1-cells). The roots of the polynomials in Σ are among the 0-cells. To illustrate the case $d = 2$, let us take for example $\Sigma = \{x^2 + y^2 - 1,\ y - x^2\}$. The

following figure represents a decomposition of $\mathbb{R}^2$ into a cell complex consisting of 17 2-cells, 25 1-cells, and 9 0-cells:



A cell complex is represented graphically by a labeled directed graph $G = (V,E,\delta,\tau)$, where V are the indices of the cells, uEv iff R_u is contained in the closure of R_v, $\delta:V\to\mathbf{N}$ gives the dimension of R_v, $v \in V$, and $\tau:\Sigma \times V \to \{-1, 0, +1\}$ gives the sign of each $p \in \Sigma$ on each $v \in V$.

In this paper we address the following problem:

(P1) Given a set Σ of polynomials in d variables, construct a graph $G = (V,E,\delta,\tau)$ representing a cell complex for Σ.

Collins' algorithm constructs a cell complex with δ and τ, but does not determine the full adjacency relation E. Like Collins, the cell complex we compute is *cylindric*[3], and therefore can be used to decide the theory of real closed fields. P1 can then be used to solve the following problem, which has wide application in motion planning:

(P2) Given Σ, construct a graph $G = (V,E,\delta,\tau)$ such that each R_v, $v \in V$, is a maximal connected component of X_σ for some consistent sign assignment σ to Σ, and E, δ, and τ give the adjacency, dimension, and sign information as in P1.

A solution for P2 can be obtained easily from a solution for P1 by collapsing adjacent cells of a cell complex on which the values of τ agree. Note that P2 is not a cell complex, since property (1) of cell complexes is no longer satisfied.

[3] The definition of this term is rather involved and is omitted from this preliminary version.

5. A key idea: naming of cells

Consistent sign assignments $\sigma:\Sigma\to\{-1, 0, +1\}$ do not distinguish the connected components of the sign classes X_σ. However, for an appropriately chosen set of polynomials Δ containing Σ, but not too much bigger, each sign class of Δ is contained in a connected component of some sign class of Σ. Moreover, the set of consistent sign assignments to Δ will determine a cell complex $\{R_v \mid v \in V\}$ for Σ. A key step in our algorithm is the use of Δ to give quantifier-free formulas

$$N_v(x_1, \ldots, x_d), \quad v \in V$$

defining the cell R_v. This will allow us, for example, to test adjacency of cells R_u and R_v by calling BKR to decide a sentence in the language of real closed fields involving N_u and N_v.

The standard representation of the formula N_v (as, say, a Boolean tree with atomic formulas at the leaves) will be far too large. However, it will have a sufficiently succinct representation as a Boolean circuit whose inputs are atomic formulas. We must then argue that BKR suffers no loss of efficiency when inputs are allowed to be of this more general form.

6. Cell decomposition in one dimension

Let Σ be a set of n polynomials in one variable x, all of degree at most n. By using the simple refinement subroutine of BKR, we may assume without loss of generality that the polynomials in Σ are square-free and pairwise relatively prime. A description of a cell complex for the original Σ can be computed easily from a cell complex for the refined Σ.

We show how to derive the coarsest one-dimensional cell complex for Σ decomposing $\mathbb{R}$. This cell complex consists of roots of polynomials in Σ and open intervals between roots. Let Δ consist of Σ plus all polynomials in the Euclidean remainder sequence $p_0 = p$,

$p_1 = p', p_2, \ldots, p_n$ for each $p \in \Sigma$. These polynomials can be produced in *NC* using subresultants [BT,Co,vzG]. We call BKR to generate the set of consistent sign assignments to the polynomials in Δ (see [BKR], Section 2).

We first give a Boolean circuit

$$M^p(\sigma), \quad p \in \Sigma$$

which takes as input a consistent sign assignment σ to Δ and computes the number of real roots of p less than $x \in X_\sigma$. This number is completely determined by σ and is independent of the choice of $x \in X_\sigma$. Each polynomial q in Δ is associated with a pair of Boolean inputs to $M^p(\sigma)$, which encode the sign $\sigma(q) \in \{-1, 0, +1\}$. The circuit implements Sturm's theorem as described in [BKR]. It requires two addition circuits to sum the number of changes of sign in two sign sequences of length n, and a subtraction circuit to subtract the two results.

We now construct a circuit

$$M^\Sigma(\sigma)$$

to sum the outputs of the circuits $M^p(\sigma)$ over all $p \in \Sigma$, giving a binary integer k_σ. By the assumption that the $p \in \Sigma$ are square-free and pairwise relatively prime, k_σ is the number of roots of all the polynomials in Σ less than any $x \in X_\sigma$.

We now incorporate $M^\Sigma(\sigma)$ into a circuit

$$N^\Sigma(\sigma)$$

that calculates the position $i_\sigma \geq 0$ of the cell in the one-dimensional cell complex containing the sign class X_σ:

$$i_\sigma = \begin{cases} 2k_\sigma \text{ if } \bigwedge_{q \in \Sigma} \sigma(q) \neq 0 \\ 2k_\sigma + 1 \text{ if } \bigvee_{q \in \Sigma} \sigma(q) = 0. \end{cases}$$

We call the circuit $N^\Sigma(\sigma)$ the *naming circuit*. This circuit will be used again later.

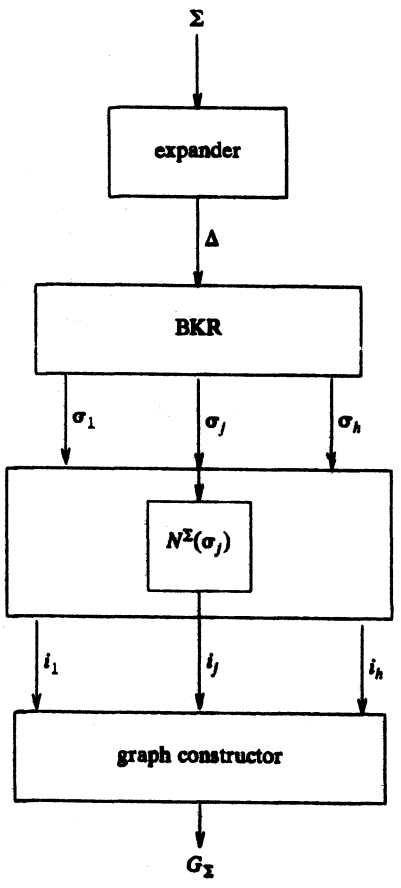


Fig. 1

We now construct a circuit

$$H(\Sigma)$$

which takes as input any set Σ of univariate polynomials, $\|\Sigma\| \leq n$, and outputs a graph G_Σ representing the coarsest one-dimensional cell complex for Σ. The stages of H are as in Fig. 1. The *expander* circuit produces Δ from Σ as explained above. The *BKR-circuit* produces all the consistent sign assignments σ to Δ. Each σ is fed into a separate copy of the circuit $N^\Sigma(\sigma)$, which outputs the cell number i_σ containing X_σ. Finally, the *graph constructor* produces an encoding of the graph representation

$$G_\Sigma = (V, E, \delta, \tau)$$

of the cell complex, as follows. The cell numbers i_σ take on all values between 0 and max i_σ inclusive, so we take $V = \{0,1, \ldots, \max i_\sigma\}$. The dimension $\delta(i)$ of cell i is 0 if i is odd, 1 if i is even. Adjacency is determined by: iEj iff i is odd and $|i-j| = 1$. Finally, the sign of $p \in \Sigma$ on cell i is given by

$$\tau(p,i) = \sigma(p)$$

where σ is any consistent sign assignment such that $i_\sigma = i$.

All these calculations are easily done in *NC*.

7. Cell decomposition in higher dimension

As in BKR, the multidimensional algorithm consists of a generalized version of the one-dimensional algorithm, modified to call itself recursively on problems of lower dimension. For this purpose, the input polynomials $\Sigma_d \subseteq \mathbb{Q}[x_1, \ldots, x_d]$ are considered to be polynomials in the single variable x_d with coefficients in the polynomial ring $\mathbb{Q}[x_1, \ldots, x_{d-1}]$. The multidimensional algorithm will attempt to construct a one-dimensional cell complex in the variable x_d by running the one-dimensional algorithm. However, whenever the one-dimensional algorithm would request information from the coefficients of the input polynomials, the multidimensional algorithm will have to call itself recursively to supply this information.

The first step of the one-dimensional algorithm was to expand the input set Σ to a set Δ containing the Euclidean remainder sequences (ERS's) of all the polynomials in Σ. The elements of Δ were obtained as subresultants, and could be constructed in *NC* [Co,BT,vzG]. In the multidimensional algorithm, the ERS of some $p \in \Sigma_d$ is not determined, since the coefficients of p contain indeterminates $\bar{x} = x_1, \ldots, x_{d-1}$; different choices of reals $\bar{a} = a_1, \ldots, a_{d-1}$ will cause these coefficients to vanish or not, as the case may be, leading to different ERS's. However, there is a small set Δ_d^p of polynomials in $\mathbb{Q}[\bar{x}][x_d]$ and a small set Γ_{d-1}^p of polynomials in $\mathbb{Q}[\bar{x}]$ with the property that all elements of all possible ERS's of p are represented by rational functions of the form

$$\frac{r_1(\bar{x})r_2(\bar{x}) \ldots r_n(\bar{x})q(\bar{x})(x_d)}{s_1(\bar{x})s_2(\bar{x}) \ldots s_m(\bar{x})}$$

where the $r_i, s_j \in \Gamma_{d-1}^p$ and $q \in \Delta_d^p$. Moreover, a consistent sign assignment σ to Γ_{d-1}^p determines a unique ERS for p that is consistent with σ. In other words, given a real $d-1$-tuple $\bar{a}$ for indeterminates $\bar{x}$, the Euclidean algorithm would give a unique ERS $p_0 = p(\bar{a})(x_d)$, $p_1 = p(\bar{a})'(x_d), \ldots, p_n(\bar{a})(x_d)$, and all these polynomials are of the form

$$\frac{r_1(\bar{a})r_2(\bar{a}) \ldots r_n(\bar{a})q(\bar{a})(x_d)}{s_1(\bar{a})s_2(\bar{a}) \ldots s_m(\bar{a})}$$

for some $r_i, s_j \in \Gamma_{d-1}^p$ and $q \in \Delta_d^p$. The signs of the constants $r(\bar{a})$ for $r \in \Gamma_{d-1}^p$ determine the ERS uniquely.

The expander circuit of the multidimensional algorithm will therefore augment the input Σ_d with Δ_d^p and Γ_{d-1}^p for all $p \in \Sigma_d$. This new set will be called Δ_d. The polynomials in Δ_d can be computed quickly as subresultants; a more exact description of these polynomials appears in [vzG].

At this point, the algorithm will call BKR on input Δ_d to determine the set of consistent sign assignments to Δ_d. Again, the set of consistent sign assignments to Δ_d is not uniquely determined, because of the indeterminates $x_1, \ldots, x_{d-1}$; however, BKR generates a set Σ_{d-1} of polynomials in $\mathbb{Q}[x_1, \ldots, x_{d-1}]$ such that a consistent sign assignment σ to Σ_{d-1} uniquely determines the set of consistent sign assignments to Δ_d that are consistent with σ. After Σ_{d-1} has been generated by BKR, the multidimensional algorithm is called recursively on Σ_{d-1}, which generates Δ_{d-1}, etc. This recursive computation corresponds to the expander and BKR circuits of the one-dimensional algorithm of Section 6.

When this part of the computation is complete, we have a sequence of sets of polynomials $\Sigma_d, \Delta_d, \Sigma_{d-1}, \Delta_{d-1}, \ldots, \Sigma_1, \Delta_1$, where $\Sigma_i, \Delta_i \subseteq \mathbb{Q}[x_1, \ldots, x_i]$, with the following properties.

(1) Any consistent sign assignment σ to Σ_{i-1} uniquely determines the set of consistent sign assignments $\rho_1, \ldots, \rho_k$ to Δ_i consistent with σ. In other words, if $\bar{a} = a_1, \ldots, a_{i-1}$ is any $i-1$ -tuple of real numbers such that σ gives the signs of the polynomials in Σ_{i-1} evaluated at $\bar{a}$, then as x_i ranges over all real numbers a_i, the polynomials in Δ_i, evaluated at $\bar{a}, a_i$, take on exactly the sign assignments $\rho_1, \ldots, \rho_k$.

(2) Each consistent sign assignment ρ to Δ_i determines a unique cell number i_ρ in a one-dimensional cell decomposition of Σ_i, regarded as a set of polynomials in x_i, according to the one-dimensional algorithm of Section 6. Thus the set of sign assignments $\rho_1, \ldots, \rho_k$ to Δ_i consistent with a sign assignment σ to Σ_{i-1} determines a unique one-dimensional cell complex for Σ_i.

We may view the output of this part of the computation as a tree of consistent sign assignments to $\Delta_1, \ldots, \Delta_d$. At the uppermost level, the tree contains all consistent sign assignments to Δ_1; and at the ith level, each consistent sign assignment σ to Δ_i gives a consistent sign assignment to Σ_i (since $\Sigma_i \subseteq \Delta_i$), which in turn determines a set of consistent sign assignments to Δ_{i+1} consistent with σ. This tree is computed explicitly by the above circuit.

We define below a d-dimensional cell complex whose cells are in one-to-one correspondence with paths in this tree. Such a path gives a sequence $\sigma_1, \ldots, \sigma_d$ of mutually consistent sign assignments to $\Delta_1, \ldots, \Delta_d$. This sequence of sign assignments determines a unique sequence of cells $i_1, \ldots, i_d$ in the one-dimensional cell complexes for $\Sigma_1, \ldots, \Sigma_d$ computed by the one-dimensional algorithm. The cell in the d-dimensional cell complex corresponding to this path is homeomorphic to the Cartesian product of the cells $i_1, \ldots, i_d$.

Formally, let $G = (V,E,\delta,\tau)$ be a d-dimensional cell complex for $\Gamma = \Sigma_1 \cup \cdots \cup \Sigma_d$, defined as follows. For each dimension i, $1 \leq i \leq d$, the construction of the circuit $N^{\Sigma}(\sigma)$ of Section 6 generalizes immediately to a circuit $N^{\Sigma_i}(\sigma_1, \ldots, \sigma_i)$ which takes as input a sequence of mutually consistent sign assignments $\sigma_1, \ldots, \sigma_i$ to $\Delta_1, \ldots, \Delta_i$ and outputs a binary number giving the position of the uniquely-determined cell in the one-dimensional cell complex for Σ_i, as described above. Combining all these circuits for $1 \leq i \leq d$, we get a circuit

$$N^{\Gamma}(\sigma_1, \ldots, \sigma_d)$$

that takes as input a sequence of mutually consistent sign assignments $\sigma_1, \ldots, \sigma_d$ to $\Delta_1, \ldots, \Delta_d$ and yields a sequence of integers $i_1, \ldots, i_d$ giving the positions of one-dimensional cells in the sequence of one-dimensional cell complexes uniquely determined by $\sigma_1, \ldots, \sigma_d$.

We take

$$V = \{i_1, \ldots, i_d \mid \exists \sigma_1, \ldots, \sigma_d \, N^{\Gamma}(\sigma_1, \ldots, \sigma_d) = i_1, \ldots, i_d\}.$$

The cell defined by $u \in V$ is the set

$$R_u = \{\bar{a} \in \mathbb{R}^d \mid N^{\Gamma}(\sigma(\bar{a})) = u\},$$

where $\sigma(\bar{a})$ is the sequence of sign assignments to the polynomials in $\Delta_1, \ldots, \Delta_d$ obtained by evaluating them at $\bar{a}$.

The dimension $\delta(u)$ of R_u is the sum of the component dimensions; i.e. $\delta(i_1, \ldots, i_d)$ is the number of i_j that are even, $1 \leq j \leq d$.

The sign $\tau(p,u)$ of $p \in \Sigma_i$ on $u \in V$ is given by

$$\tau(p,u) = \sigma_i(p),$$

where $\sigma_1, \ldots, \sigma_d$ is any sequence of mutually consistent sign assignments to $\Delta_1, \ldots, \Delta_d$ such that $N^{\Gamma}(\sigma_1, \ldots, \sigma_d) = u$.

In order to determine the adjacency relation E, we will use the naming circuit $N^{\Gamma}(\bar{\sigma})$ in a sentence of the language of real closed fields. We combine $N^{\Gamma}(\bar{\sigma})$ with a circuit to compare the output to a fixed binary d-tuple u to get a new circuit $N^{\Gamma}_u(\bar{\sigma})$ whose output is *true* or *false*. This circuit can be viewed as a formula in the language of real closed fields with free variables $\bar{x} = x_1, \ldots, x_d$, which states, "$\bar{x}$ lies in cell R_u". In this connection we write $N^{\Gamma}_u(\bar{x})$ instead of $N^{\Gamma}_u(\bar{\sigma})$. The adjacency relation E is determined by calling BKR on the formula

$$\forall \bar{x} \, (N^{\Gamma}_u(\bar{x}) \rightarrow (\forall \varepsilon > 0 \, \exists \bar{y} \, N^{\Gamma}_v(\bar{y}) \wedge |\bar{y} - \bar{x}| < \varepsilon)),$$

which states that R_u is contained in the closure of R_v. If BKR returns *true*, then edge uEv is added to G.

8. Complexity analysis

Except for the call on BKR to determine the adjacency relation E, the depth and size of the circuit are roughly the same as in BKR. The depth is $2^{O(d^2)} \log^{O(d)} n$, which gives an algorithm in *NC* for fixed d and exponential-*NC* for unbounded d (see [BKR]).

The sentence at the end of Section 7, used to determine E, has $2d + 1$ variables. Since the complexity of BKR is exponential in d, this will be the most expensive part of the circuit by far, although for fixed d the complexity will still be *NC*. The formulas $N_u^T(\bar{x})$ used in this sentence are represented by directed acyclic graphs instead of Boolean trees, but a straightforward review of [BKR] shows that this more general form of input does not affect the complexity of BKR.

9. Final remarks and open problems

We have solved the problem of computing a cell complex with adjacency information. Our result is an improvement of BKR and Collins' algorithm in that we can give a representation of the cells with full adjacency information in roughly the same time bound as BKR. It remains an open problem to improve the depth of the circuit: the depth of the BKR circuit, and also in ours, is $2^{O(d^2)} \log^{O(d)} n$ where d is the dimension. It would be a significant improvement to obtain a depth of $2^{O(d)} \log^{O(d)} n$, for instance.

References

[A1] Arnon, D.S., "A cellular decomposition algorithm for semi-algebraic sets," Comp. Sci. Tech. Report No. 353, Univ. Wisconsin, June 1979.

[A2] Arnon, D.S., "Algorithms for the geometry of semi-algebraic sets," PhD thesis, Univ. Wisconsin, 1981

[A3] Arnon, D.S., "Topologically reliable display of algebraic curves," *J. Computer Graphics* 17 (1983), 219-227.

[ACM1] Arnon, D.S., G.E. Collins, S. McCallum, "Cylindrical algebraic decomposition I: the basic algorithm," SIAM J. Comput. 13 (1984), pp. 865-877.

[ACM2] Arnon, D.S., G.E. Collins, S. McCallum, "Cylindrical algebraic decomposition II: an adjacency algorithm for the plane," SIAM J. Comput. 13 (1984), pp. 878-889.

[AM] Arnon, D.S. and S. McCallum, "A polynomial-time algorithm for the topological type of a real algebraic curve -- extended abstract," Rocky Mountain J. Math. 14 (1984), pp. 849-852.

[BKR] Ben-Or, M., D. Kozen, and J. Reif, "The complexity of elementary algebra and geometry," *Proc. 16th ACM Symp. on Theory of Computing,* May 1984, 457-464. To appear, *JCSS*.

[Bo] Borodin, A. "On relating time and space to size and depth," *SIAM J. Comput.* 6:4 (1977), 733-744.

[BT] Brown, W. and J. F. Traub, "On Euclid's algorithm and the theory of subresultants," *J. ACM* 18 (1971), 505-514.

[Co] Collins, G.E., "Quantifier elimination for real closed fields by cylindric algebraic decomposition," *Proc. 2nd GI Conference on Automata Theory and Formal Languages,* Springer-Verlag LNCS 35, Berlin, 1975, 134-183.

[Cs] Csanky, L., "Fast parallel matrix inversion algorithms," *SIAM J. Comput.* 5 (1976), 618-623.

[CY] Cole, R. and C.K. Yap, "Geometric retrieval problems," *Proc. 24th IEEE Symp. on Foundations of Computer Science*, Nov. 1983, 112-121.

[EOS] Edelsbrunner, H., J. O'Rourke, and R. Seidel, "Constructive arrangements of lines and hyperplanes with applications," *Proc. 24th IEEE Symp. on Foundations of Computer Science,* 1983, 83-91.

[Ma] Marden, M. *Geometry of Polynomials.* Amer. Math. Soc., Providence, 1966.

[McC] McCallum, S., "Constructive triangulation of real curves and surfaces," M.Sc. thesis, Univ. Sydney, 1979.

[Mo1] Monk, L. "An elementary-recursive decision procedure for Th(R,+,•)," manuscript, U. C. Berkeley, 1974.

[Mo2] Monk, L. "Elementary recursive decision procedures," PhD thesis, University of California, Berkeley, 1974.

[SS] Schwartz, J.T., and M. Sharir, "On the piano mover's problem II. General techniques for computing topological properties of real algebraic manifolds," *Adv. in Appl. Math* 4 (1983), 298-351.

[T] Tarski, A. "A decision method for elementary algebra and geometry," U. of Calif. Press, 1948; 2nd edition, 1951.

[vzG] von zur Gathen, J., "Parallel algorithms for algebraic problems," *Proc. 15th ACM Symp. on Theory of Computing,* April 1983, 17-23.

[Y] Yap, C.K., "Algorithmic motion planning," in: Advances in robotics, v.1: algorithmic and geometric aspects, ed. J.T. Schwartz and C.K. Yap, Lawrence Erlbaum Assoc., Hillsdale, New Jersey, 1985.

[YY] Yao, A.C., and F.F. Yao, "A general approach to d-dimensional geometric queries", *Proc. 17th ACM Symp. on Theory of Computing,* May 1985, 163-168.