On the complexity of intersecting multiple circles for graphical display

H.A. Kestler, J. Messner, A. Müller, R. Schuler

Ulmer Informatik-Berichte

Nr. 2008-01
Februar 2008

On the complexity of intersecting multiple circles for graphical display*

H.A. Kestler† J. Messner‡ A. Müller§ R. Schuler ¶

Abstract

Many experiments in the biomedical field generate vast amounts of data. This is especially true for microarray experiments which measure the expression levels of thousands of genes simultaneously. In this context the display of functional information attributed to the individual gene is important to obtain an overview of the major processes involved. This set data can be displayed as Euler/Venn diagrams in which the circle size corresponds to the cardinality of the set. Efficient algorithms for the calculation of intersections of circles and their resulting boundary have not been published so far. We present two algorithms (one optimal) for intersecting these different sized circles to display set relationships.

1 Introduction

Microarray technologies are increasingly being found in biological and medical sciences for high-throughput analysis of genetic information on the genome, transcriptome and proteome level. This type of analysis generates vast amounts of data, usually hundreds or thousands of genes, leaving the researcher with the task of identifying the functional relevance. Many functional attributes (terms) may have been assigned to e.g. each gene. Getting an overview over these sets of attributes is often a difficult task. Standard tree representations are in many cases an improper choice for this issue, especially in representing intersections. A Euler/Venn diagram representation of gene sets could reveal more valuable information to the researcher (see Figure 1). Full containment of one set into the other, partial intersection and disjunctness can be seen at a glance with Euler/Venn diagrams [7].

As the intersection area calculation together with its boundary is an important step in constructing an Euler/Venn diagram (if that is possible for a certain configuration in the plane), we present two algorithms (one optimal) for intersecting these different sized circles to display set relationships.

---

*All authors have contributed equally.

†Internal Medicine I and Institute of Neural Information Processing, hans.kestler@uni-ulm.de

‡Institute of Theoretical Computer Science, jochen.messner@uni-ulm.de, Ulm University, Germany

§Cosateq GmbH & Co.KG, Wangen, Germany, andre.mueller@Cosateq.com

¶Logicline GmbH, Stuttgart-Sindelfingen, Germany, rschuler@logicline.de

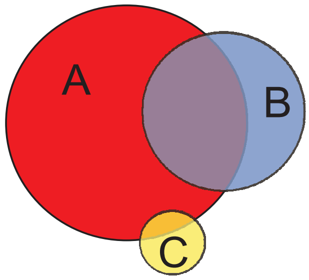

Figure 1: An Euler diagram with three sets A, B and C and their intersections. The different circle sizes represent the cardinality of the attribute sets.

2 Methods

2.1 Intersecting multiple circles

Given a family of intersecting circles $C_i = \{(x, y) \mid (x-a_i)^2 + (y-b_i)^2 = r_i^2\},\ i = 1 \dots n$ with center (a_i, b_i) and radius $r_i > 0$ find a description of the intersection set $U = D_1 \cap D_2 \cap \dots D_n$ of the corresponding disks $D_i = \{(x, y) \mid (x-a_i)^2 + (x-b_i)^2 \leq r_i^2\}$ and compute the area $A(U) = \int_U d(x, y)$.

The border ∂U should be described with a finite set $R = \{\gamma_1, \dots \gamma_K\}$ of non-overlapping x-monotone (i.e. each vertical line has at most one intersection point with the segment) circle segments $\gamma \subset \mathbb{R}^2, k = 1 \dots K$ defined with 6-tuples $(a_k, b_k, r_k, x_{k0}, x_{k1}, u_k)$ with center (a_k, b_k), radius $r_k > 0$, start and stop x-coordinate $a_k - r_k \leq x_{k0} < x_{k1} \leq a_k + k_r$, and a flag $u_k \in \{+1, -1\}$ determining if γ_k is an upper $(u_k = +1)$ or a lower segment $(u_k = -1)$.

A segment is then defined as:

$$\gamma_k = \{(x, y) \in \mathbb{R}^2 \mid y = b_k + u_k\sqrt{r_k^2 - (a_k - x)^2},\ x \in [x_{k0}, x_{k1}]\} \quad . \tag{1}$$

For a nonempty intersection U the circles necessarily intersect each other (or are fully contained in another circle - in this case the outer circle can be left away without changing the result), resulting in a maximum of $n(n-1)$ intersection points.

Algorithm I: A straightforward algorithm for this problem is: Intersect each circle $C_i,\ i = 1 \dots n$ iteratively with all other circles $C_j,\ j \neq i$ and join the resulting segments in the result set R. Hereto each circle $C_i = (a_i, b_i, r_i)$ is split into two x-monotone segments (half-circles) $\gamma_{2i-1} : (a_i, b_i, r_i, a_i - r_i, a_i + r_i, +1)$ and $\gamma_{2i} : (a_i, b_i, r_i, a_i - r_i, a_i + r_i, -1)$ (so $C_i = \gamma_{2i-1} \cup \gamma_{2i}$). These segments are restricted to all other disks to obtain the resulting (probably empty) segment set.

The complexity of the algorithm is $O(n^2K)$. Since the result set contains at most $K = 2n$ segments - each additional circle adds at most two segments to the result - the algorithm runs in the worst case with $\Theta(n^3)$ (although in practice for non-degenerate problems the number of steps will be much smaller).

The result set can be written as union of set differences

$$\partial U = \bigcup_{i=1}^{n} \left((\gamma_{2i-1} \cup \gamma_{2i}) - \bigcup_{j \neq i} D_j \right)$$

where each difference term is the contribution of circle C_i $(i = 1 \ldots n)$ to the resulting set The intersection points of the corresponding circle $\Gamma : (a, b, r)$ of a segment γ with a circle $C : (\hat{a}, \hat{b}, \hat{r})$ can be computed by subtracting the two equalities

$$(x - a)^2 + (y - b)^2 = r^2 \tag{2}$$

$$(x - \hat{a})^2 + (y - \hat{b})^2 = \hat{r}^2 \tag{3}$$

leading to the chordal line

$$H: \quad \underbrace{2(\hat{a} - a)}_{=A} x + \underbrace{2(\hat{b} - b)}_{=B} y + \underbrace{(a^2 + b^2 - r^2) - (\hat{a}^2 + \hat{b}^2 - \hat{r}^2)}_{=C} = 0 \quad , \tag{4}$$

see figure 2.

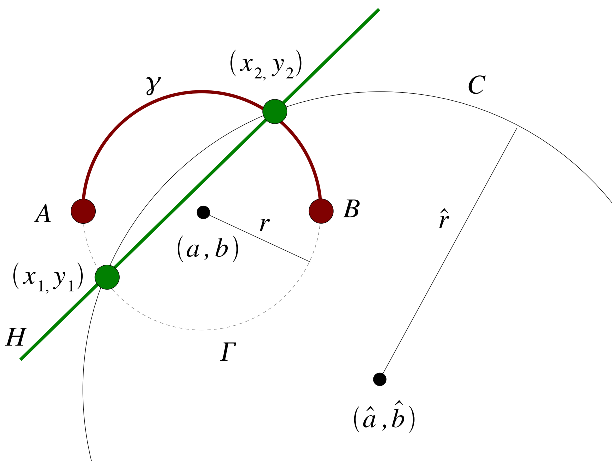

Figure 2: A circle segment γ is restricted with a circle C. The corresponding circle Γ of γ must intersect C (or has to be fully contained in C). If the left and the right endpoints A and B are on different sides of the chordal line H the segment contains one of the intersection points (x_1, y_1) or (x_2, y_2).

Inserting (2) into (4) leads to

$$\underbrace{[A^2 + B^2]}_{=\hat{A}} x^2 + \underbrace{2[AC + ABb - B^2 a]}_{=\hat{B}} x + \underbrace{[C^2 + 2BCb + B^2(a^2 + b^2 - r^2)]}_{\hat{C}} = 0$$

with the solution

$$x_{1,2} = -\frac{\hat{B}}{2\hat{A}} \mp \sqrt{\left(\frac{\hat{B}}{2\hat{A}}\right)^2 - \frac{\hat{C}}{\hat{A}}}$$

$$y_{1,2} = -\frac{C + Ax_{1,2}}{B} \quad .$$

For the case $B = 0, A \neq 0$ (vertical chordal line) the two solutions

$$x_{1,2} = -\frac{C}{A}$$
$$y_{1,2} = b \mp \sqrt{r^2 - (x_{1,2} - a)^2}$$

exist.

Case I: ($|\gamma \cap C| = 0$) If there is no intersection of the circles Γ and C it can be decided by testing if one of the endpoints of γ lies in C (in this case the whole segment is contained in C) or not (in this case the empty set is returned).

Case II: ($|\gamma \cap C| = 1$) If the two endpoints A and B of the segment γ lie on different sides of the chordal line H (or alternatively one of the endpoints is inside C and the other outside C) the segment has exactly one intersection with C and it must be decided which of the two intersection points is contained in the segment. For an upper segment γ choose the intersection point (x_1, y_1) if the left endpoint A is below the chordal line ($A <_y H$) and (x_2, y_2) if A is above H. The decision is similar for lower segments. The predicate $p <_y H$ can be evaluated using the parametrization (4) and is true iff $|B|y < -(Ax + C)\,\mathrm{sign}(B)$ with $\mathrm{sign}(v) = 1$ for $v \geq 0$ and -1 otherwise (so for vertical lines the predicate evaluates true if the point p is on the left of H). This predicate has algebraic degree 3 (compare [5]).

Case III: If both endpoints of the segment γ lie on the same side of the chordal line [1] proposed predicates based on an orientation test for the decision if there are two intersection points $|\gamma \cap C| = 2$ or none (see figure 3).

Algorithm II: Split the n circles into two sets of x-monotone segments (half-circles) Γ^L (lower segments) and Γ^U (upper segments). Each lower segment restricts the result set U from the bottom and each upper segment from the top. The intersection area U can therefore be found be intersecting the upper envelope of the lower segments with the lower envelope of the upper segments.

The intersection area U is fully contained in the interval of interest $I = [x_0, x_1] = [\max_i a_i - r_i, \min_i a_i + r_i]$. If $x_0 > x_1$ no intersection is possible and $U = \emptyset$. In the following we assume $U \neq \emptyset$. The lower and upper segments are well defined continous, univariate functions $f_1 \dots f_n : I \mapsto \mathbb{R}$ and $g_1 \dots g_n : I \mapsto \mathbb{R}$ with the property, that two functions f_i and f_j intersect in at most $s = 2$ points for all $i \neq j$. The lower envelope of the upper segments $\mathcal{G} = \{g_1 \dots g_n\}$ is defined as

$$E_{\mathcal{G}}(x) = \min_{1 \leq i \leq n} g_i(x) \quad x \in I$$

and the upper envelope of the lower segments $\mathcal{F} = \{f_1 \dots f_n\}$ as

$$E^*_{\mathcal{F}}(x) = \max_{1 \leq i \leq n} f_i(x) \quad x \in I \quad .$$

The complexity of the lower/upper envelope of a set of functions, which intersect in at most s points, can be described with Davenport-Schinzel (DS) sequences. A $DS(n, s)$ sequence is defined as a sequence $< u_1 \dots u_m >$ of integers with the following properties:

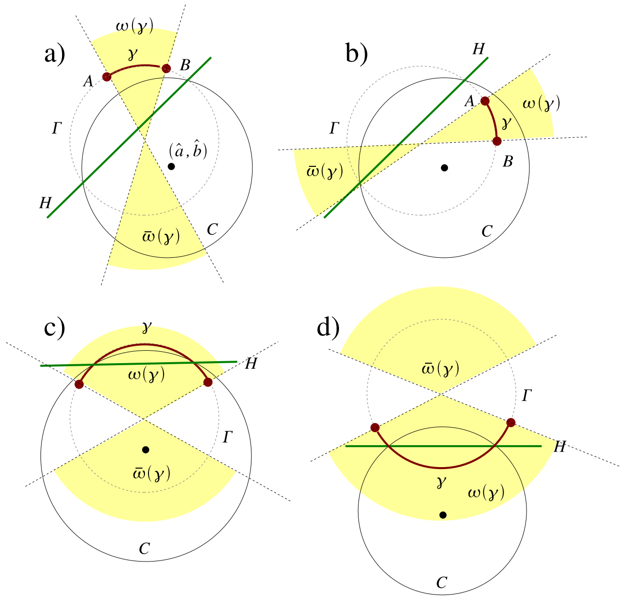

Figure 3: If C with center $(\hat{a}, \hat{b})$ and Γ build a non-empty intersection and the two endpoints of the segment γ lie on one side of the chordal line H the four cases must be distinguished: **a)** $|\gamma \cap D| = \emptyset$ if $(\hat{a}, \hat{b}) \notin \omega(\gamma) \cup \varpi(\gamma)$ and $A, B \notin D$, **b)** $\gamma \cap D = \gamma$ if $(\hat{a}, \hat{b}) \notin \omega(\gamma) \cup \varpi(\gamma)$ and $A, B \in D$, **c)** $|\gamma \cap C| = 2$ and two segments result from $\gamma \cap D$ if $A, B \in D$ and $(\hat{a}, \hat{b}) \in \varpi(\gamma)$, **d)** $|\gamma \cap C| = 2$ and one segment results from $\gamma \cap D$ if $A, B \notin D$ and $(\hat{a}, \hat{b}) \in \omega(\gamma)$. The two wedges $\omega(\gamma)$ and $\varpi(\gamma)$ divide the plane in four regions. The test $p \in \omega(\gamma)$ and $p \in \varpi(\gamma)$ can easily be solved with orientation tests.

1. $1 \leq u_i \leq n$ for each i
2. $u_i \neq u_{i+1}$ for each $i < m$
3. There do not exist $s + 2$ indices $1 \leq i_1 < i_2 < \ldots < i_{s+2} \leq m$ such that $u_{i_1} = u_{i_3} = \ldots = a$ and $u_{i_2} = u_{i_4} = \ldots = b$ for $a \neq b$

The maximum length of a $DS(n, s)$ sequence is defined as $\lambda_s(n)$.

Following [10] (chapter 6.2) the two envelopes can be computed in $O(\gamma_s(n) \log n)$ steps via a divide-and-conquer algorithm where $\gamma_s(n)$ is the maximum length of a $DS(s, n)$ sequence. For this $\mathcal{F}$ is partitioned into two sets $\mathcal{F}_\infty$, $\mathcal{F}_\in$, each of at most size $\lceil n/2 \rceil$. Upper evelopes $E^*_{\mathcal{F}_\infty}$ and $E^*_{\mathcal{F}_\in}$ are constructed recursively. $E^*_{\mathcal{F}}$ can be constructed from $E^*_{\mathcal{F}_\infty}$ and $E^*_{\mathcal{F}_\in}$ in $O(\gamma_s(n))$ steps. The same holds for $\mathcal{G}$. It is required that two functions can be intersected in $O(1)$ to achieve the given complexity. In the case of circles $s = 2$ and therefore $\gamma_s(n) = 2n - 1$. Merging the two envelopes $E_{\mathcal{G}}$ and $E^*_{\mathcal{F}}$ to get the intersection region

$$\Pi_{\mathcal{F},\mathcal{G}} = \{(x, y) \mid E^*_{\mathcal{F}}(x) \leq y \leq E_{\mathcal{G}}(x), y \in I\}$$

requires further $O(\gamma_s(n)) = O(n)$ [2] steps via an iterative search. The region $\Pi_{\mathcal{F},\mathcal{G}}$ is identical to the intersection set U of all n circular disks. The total runtime of the algorithm is $O(n \log n)$ if the intersection of two functions can be computed in $O(1)$.

2.2 Calculating the area

Let $U \subset \mathbb{R}^2$ be a connected set with the boundary $\partial U = \bigcup_{i=1}^{K} \gamma_k$ consisting of non-overlapping segments. Using the Gaussian integration theorem (see e.g. [6]) and integrating along the border ∂U with the parametrization (1) the area $A(U) = \int_U d(x,y)$ can be reformulated to

$$\frac{1}{2}\sum_{k=1}^{K}\left[2u_k b_k x + (x-a_k)\sqrt{r_k^2-(x-a_k)^2} + r_k^2 \arcsin\left(\frac{x-a_k}{r_k}\right)\right]_{x=x_{k0}}^{x=x_{k1}} . \quad (5)$$

3 Lower bounds

In this section we discuss the minimum number of operations needed to compute the boundary ∂U of the intersection $D_1 \cap \cdots \cap D_n$ of n disks D_i. Actually our lower bounds already apply to the special case where all disks have the same radius. We demonstrate a linear time reduction from sorting and provide an $\Omega(n \log n)$ lower bound in the algebraic computation tree model (which is introduced in Section 3.2).

Notice that Algorithm II above produces the segments in a certain order, such that consecutive segments on ∂U are listed consecutively. This is however not necessary for the computation of the area with Formula (5). Algorithm I just guaranties the maximility of the segments. It is therefore sensible to consider two problems taking into account these differences.

1. Computing the boundary of the intersection of discs (BID): The input is a list of n centers $(a_j, b_j) \in \mathbb{R}^2$, $1 \le j \le n$ and a radius $r > 0$, representing n discs $D_1, \ldots, D_n$ of radius r. The output is a list $(\gamma_1, \ldots, \gamma_K)$ with $\gamma_i = (a_{j_i}, b_{j_i}, x_i, x'_i, u_{j_i})$, $1 \le i \le K$, representing x-monotone circle segments of the circles ∂D_i that yield the boundary of $U = D_1 \cap \cdots \cap D_n$ without overlap.

   For convinience, we further assume that lower and upper segments are listed separately: for some J with $1 \le J < K$ we have $u_i = +1$ for $1 \le i \le J$ and $u_i = -1$ for $J+1 \le i \le K$.

2. Computing the ordered boundary of the intersection of discs (OBID): Additionally to the specification of BID it holds here that consecutive segments are listed consecutively in the output, i. e. $x'_{j_i} = x_{j_{i+1}}$ for $i \ne J$, $1 \le i \le K$.

Notice that we do not require here that the output segments are maximal (which is the case if $(a_{j_i}, b_{j_i}, u_{j_i}) = (a_{j_l}, b_{j_l}, u_l)$ implies $x'_i \ne x_l$ for $i, l \in \{1, \ldots, J\}$). In contrast, both algorithms above provide maximality since the endpoints of the circle segments are determined by an intersection with one of the other circles or as endpoint of the input half circles. In fact, for OBID we can obtain maximality easily in linear time by joining all non-maximal consecutive segments $(a_{j_i}, b_{j_i}, x_i, x'_i, u_{j_i})$, $(a_{j_{i+1}}, b_{j_{i+1}}, x_{i+1}, x'_{i+1}, u_{j_{i+1}})$ with $x'_i = x_{i+1}$ and $(a_{j_i}, b_{j_i}) = (a_{j_{i+1}}, b_{j_{i+1}})$ together to $(a_{j_i}, b_{j_i}, x_i, x'_{i+1}, u_{j_i})$.

Clearly OBID, as a special case of BID, reduces to BID in linear time. For a reduction from BID to OBID notice that that any segment $\gamma = (a, b, x, x', u)$ in some sense links the segment before and after it. I.e., let $\gamma' = (a', b', z, z', u')$ and $u = u'$ then γ' is connected with γ iff $x' = z$ or $z' = x$. One can use this observation for ordering the segments bounding ∂U in linear time by storing segments at adresses determined by their endpoints. This is possible on computational models that allow indirect addressing with real numbers like some of the RAM-models in [3] or just by using hashing (with high probability). However since the known lower bounds for sorting do not allow the use of such techniques, and also the algebraic computation tree model does not allow to use computed values as adresses, we have to distinguish the two problems.

We will now first consider OBID and show that it is at least as hard as sorting. Moreover, OBID reduces to the convex hull problem in linear time. Then we consider BID and provide lower bounds in the algebraic computation tree model.

3.1 Computing the ordered list of boundary segments

We now provide a linear time reduction of the sorting problem to OBID.

Theorem 1 *The task to sort n distinct real numbers can be reduced in $O(n)$ time to the task to compute the ordered boundary of the intersection of n disks.*

Proof. The idea is to place the numbers which we want to sort on the upper half of a circle as centers of the disks. The ordered list of the lower segments bounding the intersection then gives the centers of the disks in counterclockwise order which respresents the reversed sorting of the input. See Figure 4 for an illustration.

Formally the reduction is done as follows: Let $z_1, \ldots, z_n \in \mathbb{R}$ the numbers to be sorted. First in $O(n)$ steps determine $c = \max_{1 \le i \le n} |z_i|$. For $i = 1, \ldots, n$ let D_i be the disk with center (z_i, b_i) on the upper halfcircle C with center 0 and radius c, i.e. $b_i := \sqrt{c^2 - z_i^2}$ and let $r > c$ be arbitrary.

Let now $(\gamma'_1, \ldots, \gamma'_K)$ be an orderd list of segments bounding $D_1 \cap \cdots \cap D_n$. First in $O(K)$ steps join successive segments belonging to the same circle. We obtain an ordered list of maximal segments $(\gamma_1, \gamma_2, \gamma_3, \ldots, \gamma_{n+2})$ with $\gamma_i = (a'_i, b'_i, x_i, x'_i, u_i)$ for $1 \le i \le n+2$. Moreover γ_1, γ_2 are upper bounds ($u_i = 1$) and γ_i for $2 \le i \le n+2$ is a lower bound ($u_i = -1$). The sorted ordering of $z_1, \ldots, z_n$ is now given by $a'_{n+2}, \ldots, a'_3$.

To see that the reduction is valid the essential observation is that here the clockwise ordering of the boundary of ∂U yields a clockwise ordering of the centers of the discs on C. This follows from the results in Lemma 2 below. Notice that here all centers are vertices of the convex hull of the centers, and that all the input circles contribute to the boundary, i.e., using notation from Lemma 2 H_r consists of all centers. $\square$

Due to the $\Omega(n \log n)$ lower bound for sorting, we obtain an $\Omega(n \log n)$ lower bound for the problem to determine the ordered boundary of the intersection of n disks in the same computational models, when the operations used in the proof (jumps on comparisons, multiplication and square root) are possible in that model.

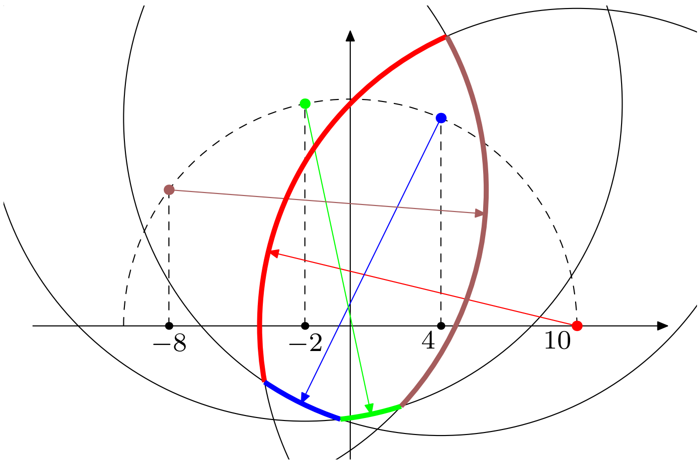

Figure 4: Illustration for the reduction in the proof of Theorem 1. One obtains the sorted permutation of $\{-8, -2, 4, 10\}$ by a clockwise traversal of ∂U, starting at the point with maximum x-value.

The reduction in the proof of the above theorem is quite similar to Shamos' reduction from sorting to the ordered convex hull problem of a point set (cf. [9]). This is no surprise, since, as we will see in the following Lemma, the convex hull problem and the problem to determine the boundary of intersecting disks of equal radius are closely related. Let in the following $d(p_1, p_2)$ denote the Euclidian distance.

Lemma 2 *Let S be a point set and let $H \subseteq S$ be the set of vertices of the convex hull of S. Let $r > 0$ be a radius, and let D_p be the disk with radius r and center p with corresponding circle ∂D_p. Let $U = \bigcap_{p \in S} D_p$ with boundary ∂U.*

Let $h^+_{pp'}$ for $p, p' \in S$, $p \neq p'$, be the halfspace of points $q \in \mathbb{R}^2$ with $d(p, q) \leq d(p', q)$, and let $h_{pp'} = \partial h^+_{pp'} = \{p \mid d(p, q) = d(p', q)\}$. Let $F_p = \bigcap_{p' \in S \setminus \{p\}} h^+_{p'p}$ be the set of points q for which p is a farthest point in S. Let $r_p = \min\{d(p, q) \mid q \in F_p\}$, and let $H_r = \{p \in H \mid r_p \leq r\}$.

Then the following holds:

1. *$\partial D_p \cap \partial U = \partial U \cap F_p$ for $p \in S$.*
2. *$\partial U = \bigcup_{p \in S} F_p \cap \partial D_p = \bigcup_{p \in H} F_p \cap \partial D_p = \bigcup_{p \in H_r} F_p \cap \partial D_p$*
3. *$U = \bigcap_{p \in H} D_p = \bigcap_{p \in H_r} D_p$.*
4. *Let $p_0, \ldots, p_{K-1}$ be a clockwise ordering of H_r. Then $\partial D_{p_0} \cap F_{p_0}, \ldots, \partial D_{p_{K-1}} \cap F_{p_{K-1}}$ is ∂U in clockwise ordering.*

Proof.

1. Assume $q \in \partial D_p \cap \partial U$ then $d(q, p) = r$ and $d(q, p') \leq r$ for $p' \in S$, since otherwise $q \notin U$. Hence $q \in F_p$.

   Assume now $q \in \partial U \cap F_p$. By $q \in U \cap F_p$ we have $d(q, p') \leq d(q, p) \leq r$. Hence, since q is on the boundary of U, $d(q, p) = r$ and $q \in \partial D_p$.

2. $\partial U \subseteq \bigcup_{p\in S} F_p \cap \partial D_p$ since every $q \in \partial U$ must be on ∂D_p for some $p \in S$. Hence $d(p,q) = r$ and $d(p',q) \leq r$ for $p' \in S$, which shows $q \in F_p$.

   On the other hand $F_p \cap \partial D_p \subseteq \partial U$ for $p \in S$ since $q \in F_p \cap \partial D_p$ implies $d(q,p) = r$ and $d(q,p') \leq r$ for $p' \in S$. Hence $q \in \partial U$.

   $\bigcup_{p\in S} F_p \cap \partial D_p = \bigcup_{p\in H} F_p \cap \partial D_p$ since $F_p = \emptyset$ for $p \in S \setminus H$.

   $\bigcup_{p\in H} F_p \cap \partial D_p = \bigcup_{p\in H_r} F_p \cap \partial D_p$ since $F_p \cap D_p = \emptyset$ for $p \in H \setminus H_r$.

3. Clearly, $U \subseteq \bigcap_{p\in H} D_p \subseteq \bigcap_{p\in H_r} D_p$.

   $\bigcap_{p\in H_r} D_p \subseteq U$: Observe that the boundary of $\bigcap_{p\in H_r} D_p$ is obviously equal to $\bigcup_{p\in H_r} F_p \cap \partial D_p$. By *2*, we have that the boundary of $\bigcap_{p\in H_r} D_p$ is equal to ∂U. This implies also that $\bigcap_{p\in H_r} D_p = U$

4. Consider three consecutive segments $\gamma_a, \gamma_b, \gamma_c$ which are in clockwise order on ∂U, and let P_a, P_b, P_c denote the centers of the according circles D_a, D_b, D_c with radius r. We will show that $0 \leq \angle P_c P_b P_a \leq \pi$, so that there is right turn when one passes in straight lines from P_a to P_b to P_c. The statment then follows by induction. See Figure 5 for an illustration of the notions used in the proof.

   Let I_{ab} (I_{bc}) be the common point of γ_a with γ_b (γ_b with γ_c, respectively), and let D_{ab} (D_{bc}) be a circle of radius r with center I_{ab} (resp. I_{bc}). Since I_{ab} is in distance r of both points P_a and P_b, P_a and P_b are on ∂D_{ab}. Further we know that P_a is in D_{bc}, since I_{bc} is in $U \subseteq D_a$, and we know that P_b is an intersection point of ∂D_{ab} with ∂D_{bc}. Let γ_{ab} denote the segment of ∂D_{ab} that passes from P_a to P_b in clockwise direction. We now show that γ_{ab} is in D_{bc}, i.e. $\gamma_{ab} \subseteq \partial D_{ab} \cap D_{bc}$. First observe that the angle $\angle P_b I_{ab} P_a$ is less than π since U is convex. Further it is clear that the segment $\partial D_{ab} \setminus D_{bc}$ stretches more than a halfcircle. Since $P_b, P_a \in D_{bc}$ it follows $\gamma_{ab} \subseteq D_{bc}$. Similarly, let γ_{bc} denote the segment of ∂D_{bc} that passes from P_b to P_c in clockwise direction. Here we obtain similarly $\gamma_{bc} \subseteq D_{ab}$. So if one traverses $\partial(D_{ab} \cap D_{bc})$ from P_c to P_a counterclockwise, one encounters P_b along the way, which shows that $0 \leq \angle P_c P_b P_a \leq \pi$ since $D_{ab} \cap D_{bc}$ is convex.

□

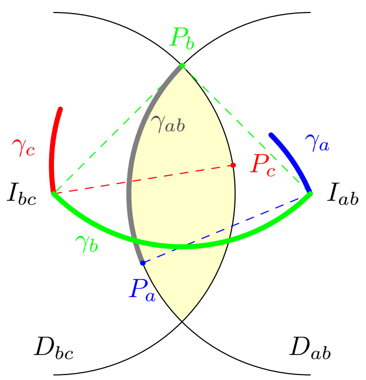

Figure 5: Illustration for the proof of Lemma 2 Item 4.

In the reduction in the proof of Theorem 1 we used the observation due to Lemma 2 that in case all points in S are on a circle of radius c then $r_p = c$ for any $p \in S$. Then choosing $r > c$ implies $H_r = S$ and that a sorted describtion of ∂U gives a clockwise ordering of S.

Since every segment of ∂U belongs to a circle whose center is a vertex of the boundary of the convex hull of S (Lemma 2.3) one can use the output sensitive algorithm from [8] for the convex hull problem to obtain an algorithm for the computation of ∂U that is faster in case the convex hull is small. Using the algorithm of [8] that produces the convex hull of a set of n points in time $O(n \log K')$ where K' is the complexity of the convex hull one obtains an algorithm for the intersection of equal sized disks with time bound $O(n \log K') + O(K' \log K') = O(n \log K')$.

In fact, from Lemma 2 Item 4 one can derive a linear time reduction that from the vertices of the convex hull (i.e. the set H) when given in clockwise order computes the ordered boundary of U. To see this, assume first that we already know $H_r = \{q_0, \ldots, q_{k-1}\}$ listed in clockwise order. Lemma 2 Item 4 then implies that $\partial D_{q_i} \cap \partial U = \partial D_{q_i} \cap h^+_{q_{i-1}q_i} \cap h^+_{q_{i+1}q_i}$ with $i \in \mathbb{Z}_k$ (i.e. the operations are modulo k). So if we know H_r the ordered boundary of U can be computed in linear time.

Let now $H = \{p_1, \ldots, p_n\}$ listed in clockwise order. For $i = 1$ to n we build a doubly linked list $H_r^{(i)}$ by considering only the intersection of the discs for $p_1, \ldots, p_i$, such that $H_r = H_r^{(n)}$. Initially, $H_r^{(0)} = \emptyset$. For the ith step let $H_r^{(i-1)} = \{q_1, \ldots, q_l\} \subseteq \{p_1, \ldots, p_{i-1}\}$ be already computed, and listed in clockwise order. Now p_i is eventually added in between q_1 and q_l. If $\partial D_{p_i} \cap h^+_{q_1 p_i} \cap h^+_{q_l p_i} = \emptyset$ then $H_r^{(i)} := H_r^{(i-1)}$ and p_i is not added. Otherwise p_i will be added but some neighbouring points may be removed from $H_r^{(i-1)}$: let $j = 1, 2, 3 \ldots$ until $\partial D_{q_j} \cap h^+_{p_i q_j} \cap h^+_{q_{j+1} q_j} \neq \emptyset$, which means that the circles around $q_1, \ldots, q_{j-1}$ do not contribute to ∂U. Similarily let $k = l, l-1, l-2, \ldots$ until $\partial D_{q_k} \cap h^+_{p_i q_k} \cap h^+_{q_{k-1} q_k} = \emptyset$. Now set $H_r^{(i)} := \{q_j, \ldots, q_k, p_i\}$.

Clearly the update in the ith step needs linear time in the number of removed or added elements when $H_r^{(i)}$ is organized as a doubly linked list. Since every point of H is added and removed at most once this give overall linear time complexity.

Theorem 3 *OBID reduces in linear time to the ordered convex hull problem.*

It is not clear whether a similar efficient reduction in the reverse direction is possible. Though if the radius r is big enough then $H_r = H$ and therefore the centers of the circles involved in the boundary of U are exactly the vertices of the convex hull of S, however the exact value when r is "big enough" will depend on the input values.

3.2 Computing the set of boundary segments

We now prove a lower bound for BID in the algebraic computation tree model using the approach of [4]. Let us briefly introduce the model. An algebraic computation tree is a tree consisting of branching nodes, computation nodes and leaf nodes. Each computation node v has only one child and is associated to a variable f_v and an operation

$$f_v := y \circ z \text{ or } f_v := c \text{ or } f_v := \sqrt{y}$$

where y, z are either input variables from $\{x_1, \ldots, x_n\}$ or variables associated to ancestors of v in the tree, the operation $\circ \in \{+, -, \times, /\}$, and $c \in \mathbb{R}$ is a constant. A branching node v is associated to a test

$$y > 0 \text{ or } y \geq 0 \text{ or } y = 0$$

where y is a variable associated to an ancestor of v. A leaf node specifies a list of output variables $(y_1, \ldots, y_l)$ which are all associated to some ancestor node. In case the computation tree is used for some decision problem it suffices to label the leaves with yes or no.

A computation proceeds along a path in the tree. So the worst case time complexity of an algorithm specified by a computation tree T is given by the height $h(T)$ of the tree. Observe that one can unroll Algorithm II to obtain an algebraic computation tree T_n of height $h(T_n) = O(n \log n)$ that on input of $(a_1, b_1, r_1, \ldots, a_n, b_n, r_n)$ computes the boundary of the intersection of the associated disks.

The results in [4] apply to decision problems so it is necessary to define an appropriate decision problem that easily reduces to the functional problem. Let us consider the following problem.

$$\begin{aligned} \text{regular-}n\text{-gon} = \{(s, a_1, b_1, \ldots, a_n, b_n) \,|& (a_i, b_i) \text{ are the vertices of a regular } n\text{-gon} \\ & \text{and } a_i^2 + b_i^2 = s^2 \text{ for } 1 \leq i \leq n\} \end{aligned}$$

So $(a_1, b_1, \ldots, a_n, b_n) \in$ regular-n-gon iff every $p_i := (a_i, b_i)$ lies on the circle centered at the origin with radius s and for some permutation σ $d(p_{\sigma(i)}, p_{\sigma(i+1)}) = 2 \sin \frac{\pi}{n}$ for $1 \leq i < n$ (σ puts the points in a clockwise or a counter-clockwise ordering).

For some $w = (r, a_1, b_1, \ldots, a_n, b_n) \in \mathbb{R}^{2n+1}$, $n \geq 2$ with $r > s > 0$ let D_i be the disk of radius r with center (a_i, b_i) and let $U_w = D_1 \cap \ldots \cap D_n$. Observe that $(s, a_1, b_1, \ldots, a_n, b_n) \in$ regular-n-gon iff $a_i^2 + b_i^2 = s^2$ for $1 \leq i \leq n$ and U_w is contained in a disk D centered at the origin $(0, 0)$ with radius

$$r' = s\sqrt{c^2 + \frac{r^2}{s^2} - 1} - c \text{ where } c = \cos\frac{\pi}{n}\,.$$

In fact, as basic geometric considerations show, $(s, p_1, \ldots, p_n)$ forms a regular n-gon iff all the vertices q of ∂U_w are at distance r' from the origin.

A circle segment $\gamma_k = (a'_k, b'_k, x_{k0}, x_{k1}, u_k)$ has the endpoints $q_{k0} = (x_{k0}, y_{k0})$ and $q_{k1} = (x'_{k1}, y'_{k1})$ with $y_{ki} = b'_k + u_k\sqrt{r^2 - (a'_k - x_{ki})^2}$ for $i = 0, 1$. Hence, if $(\gamma_1, \ldots, \gamma_K)$ is a describtion of the boundary of U_w where $\gamma_k = (a'_k, b'_k, x_{k0}, x'_{k1}, u_k)$ for $1 \leq k \leq K$ then $(s, a_1, b_1, \ldots, a_n, b_n) \in$ regular-n-gon iff $a_i^2 + b_i^2 = s^2$ for $1 \leq i \leq n$, and the endpoints q_{k0}, q_{k1} are at most at distance r' from the origin for $1 \leq k \leq K$. This consideration shows that an algebraic computation tree for the problem to determine the boundary of U_w can be transformed to an algebraic computation tree for regular-n-gon such that the depth is increased by $O(n)$.

Now we use the main result from [4] to prove a lower bound for regular-n-gon. Let us first introduce the following notation. For some set $W \in \mathbb{R}^n$ let $\#W$ denote the number of connected components W consists of.

Lemma 4 ([4]) *Let $W \in \mathbb{R}^n$ and let T be an algebraic compution tree deciding W with height $h(T)$. Then*

$$h(T) \geq \log_6 \#W - n \log_6 3 \tag{6}$$

Observe that regular-n-gon consists of $n!$ connected components: In fact, for any permutation $\sigma : \{1, \ldots n\} \to \{1, \ldots, n\}$ there is a different connected component V_σ given by all the rotations of the σ-permuted n-gons. I.e.

$$V_\sigma = \left\{ (s, a_1, b_1, \ldots, a_n, b_n) \;\middle|\; \begin{pmatrix} a_i \\ b_i \end{pmatrix} = \begin{pmatrix} s\cos(\frac{\pi}{n}\sigma(i) + \beta) \\ s\sin(\frac{\pi}{n}\sigma(i) + \beta) \end{pmatrix} \text{ for } \begin{array}{c} 1 \leq i \leq n, \\ 0 \leq \beta < \frac{2\pi}{n}, \\ s > 0. \end{array} \right\}.$$

This proves the lower bound.

Theorem 5 *Let T be an algebraic computation tree of height $h(T)$ deciding regular-n-gon. Then $h(T) \in \Omega(n \log n)$.*

And due to the above reduction:

Theorem 6 *Let T be an algebraic computation tree of height $h(T)$ that on input $w = (r, a_1, b_1, \ldots, a_n, b_n)$ computes the boundary of U_w. Then $h(T) \in \Omega(n \log n)$.*

This allows us to state the optimality of Algorithm II.

4 Summary and Discussion

To visualize functional categories of interesting genes GoMiner [11] can be used to annotate genes with functional terms. Due to the fact that one gene may belong to multiple functional categories, this analysis usually reveales a complex pattern of terms. To identify the major functional categories differentiating cell types, Euler/Venn diagramms approximated by polygons proved to be useful [7].

Here, the Euler/Venn diagram application served as a motivation for investigating the complexity of finding the intersection of multiple circles. We have described two algorithms (one trivial and one optimal) for this task. To find the intersection of n circles a running time of $O(n \log n)$ is needed. If on the other hand all combinations of circles are intersected the overall running time of computing all areas would lead to $\sum_{i=1}^{n} \binom{n}{k} O(k \log k)$ by directly applying our optimal algorithm to every circle combination. This can further be reduced to $\sum_{i=1}^{n} \binom{n}{k} O(k)$ by reusing already intersected circle segments.

References

[1] P. K. Agarwal, M. Pellegrini, and M. Sharir. Counting circular arc intersections. *SIAM Journal of Computing*, 22(4):778–793, 1993.

[2] P. K. Agarwal and M. Sharir. Davenport-schinzel sequences and their geometric applications. Technical report, Duke University, 1995.

[3] A. M. Ben-Amram and Z. Galil. Topological lower bounds on algebraic random access machines. *SIAM Journal on Computing*, 31(3):722–761, 2001.

[4] M. Ben-Or. Lower bounds for algebraic computation trees (preliminary report). In *STOC '83: Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 80–86. ACM Press, 1983.

[5] T. M. Chan. Reporting curve segment intersections using restricted predicates. *Computational Geometry*, 16(4):245–256, 2000.

[6] J. W. Harris and H. Stocker. *Handbook of Mathematics and Computational Science*. Springer, 1998.

[7] H. A. Kestler, A. Müller, T. M. Gress, and M. Buchholz. Generalized venn diagrams: a new method of visualizing complex genetic set relations. *Bioinformatics*, 21(8):1592–5, 2005.

[8] D. G. Kirkpatrick and R. Seidel. The ultimate planar convex hull algorithm. *SIAM Journal on Computing*, 15(1):287–299, 1986.

[9] F. P. Preparata and M. I. Shamos. *Computational Geometry*. Springer-Verlag, 1985.

[10] M. Sharir and P. K. Agarwal. *Davenport-Schinzel Sequences and their Geometric Applications*. Cambridge University Press, 1995.

[11] B. R. Zeeberg, W. Feng, G. Wang, M. D. Wang, A. T. Fojo, M. Sunshine, S. Narasimhan, D. W. Kane, W. C. Reinhold, S. Lababidi, K. J. Bussey, J. Riss, J. C. Barrett, and J. N. Weinstein. GoMiner: A Resource for Biological Interpretation of Genomic and Proteomic Data. *Genome Biology*, 4:R28, 2003.

Liste der bisher erschienenen Ulmer Informatik-Berichte

Einige davon sind per FTP von `ftp.informatik.uni-ulm.de` erhältlich
Die mit * markierten Berichte sind vergriffen

List of technical reports published by the University of Ulm

Some of them are available by FTP from `ftp.informatik.uni-ulm.de`
Reports marked with * are out of print

91-01 *Ker-I Ko, P. Orponen, U. Schöning, O. Watanabe*
Instance Complexity

91-02* *K. Gladitz, H. Fassbender, H. Vogler*
Compiler-Based Implementation of Syntax-Directed Functional Programming

91-03* *Alfons Geser*
Relative Termination

91-04* *J. Köbler, U. Schöning, J. Toran*
Graph Isomorphism is low for PP

91-05 *Johannes Köbler, Thomas Thierauf*
Complexity Restricted Advice Functions

91-06* *Uwe Schöning*
Recent Highlights in Structural Complexity Theory

91-07* *F. Green, J. Köbler, J. Toran*
The Power of Middle Bit

91-08* *V.Arvind, Y. Han, L. Hamachandra, J. Köbler, A. Lozano, M. Mundhenk, A. Ogiwara, U. Schöning, R. Silvestri, T. Thierauf*
Reductions for Sets of Low Information Content

92-01* *Vikraman Arvind, Johannes Köbler, Martin Mundhenk*
On Bounded Truth-Table and Conjunctive Reductions to Sparse and Tally Sets

92-02* *Thomas Noll, Heiko Vogler*
Top-down Parsing with Simulataneous Evaluation of Noncircular Attribute Grammars

92-03 *Fakultät für Informatik*
17. Workshop über Komplexitätstheorie, effiziente Algorithmen und Datenstrukturen

92-04* *V. Arvind, J. Köbler, M. Mundhenk*
Lowness and the Complexity of Sparse and Tally Descriptions

92-05* *Johannes Köbler*
Locating P/poly Optimally in the Extended Low Hierarchy

92-06* *Armin Kühnemann, Heiko Vogler*
Synthesized and inherited functions -a new computational model for syntax-directed semantics

92-07* *Heinz Fassbender, Heiko Vogler*
A Universal Unification Algorithm Based on Unification-Driven Leftmost Outermost Narrowing

92-08* *Uwe Schöning*
On Random Reductions from Sparse Sets to Tally Sets

92-09* *Hermann von Hasseln, Laura Martignon*
Consistency in Stochastic Network

92-10 *Michael Schmitt*
A Slightly Improved Upper Bound on the Size of Weights Sufficient to Represent Any Linearly Separable Boolean Function

92-11 *Johannes Köbler, Seinosuke Toda*
On the Power of Generalized MOD-Classes

92-12 *V. Arvind, J. Köbler, M. Mundhenk*
Reliable Reductions, High Sets and Low Sets

92-13 *Alfons Geser*
On a monotonic semantic path ordering

92-14* *Joost Engelfriet, Heiko Vogler*
The Translation Power of Top-Down Tree-To-Graph Transducers

93-01 *Alfred Lupper, Konrad Froitzheim*
AppleTalk Link Access Protocol basierend auf dem Abstract Personal Communications Manager

93-02 *M.H. Scholl, C. Laasch, C. Rich, H.-J. Schek, M. Tresch*
The COCOON Object Model

93-03 *Thomas Thierauf, Seinosuke Toda, Osamu Watanabe*
On Sets Bounded Truth-Table Reducible to P-selective Sets

93-04 *Jin-Yi Cai, Frederic Green, Thomas Thierauf*
On the Correlation of Symmetric Functions

93-05 *K.Kuhn, M.Reichert, M. Nathe, T. Beuter, C. Heinlein, P. Dadam*
A Conceptual Approach to an Open Hospital Information System

93-06 *Klaus Gaßner*
Rechnerunterstützung für die konzeptuelle Modellierung

93-07 *Ullrich Keßler, Peter Dadam*
Towards Customizable, Flexible Storage Structures for Complex Objects

94-01 *Michael Schmitt*
On the Complexity of Consistency Problems for Neurons with Binary Weights

94-02 *Armin Kühnemann, Heiko Vogler*
A Pumping Lemma for Output Languages of Attributed Tree Transducers

94-03 *Harry Buhrman, Jim Kadin, Thomas Thierauf*
On Functions Computable with Nonadaptive Queries to NP

94-04 *Heinz Faßbender, Heiko Vogler, Andrea Wedel*
Implementation of a Deterministic Partial E-Unification Algorithm for Macro Tree Transducers

94-05 *V. Arvind, J. Köbler, R. Schuler*
On Helping and Interactive Proof Systems

94-06 *Christian Kalus, Peter Dadam*
Incorporating record subtyping into a relational data model

94-07 *Markus Tresch, Marc H. Scholl*
A Classification of Multi-Database Languages

94-08 *Friedrich von Henke, Harald Rueß*
Arbeitstreffen Typtheorie: Zusammenfassung der Beiträge

94-09 *F.W. von Henke, A. Dold, H. Rueß, D. Schwier, M. Strecker*
Construction and Deduction Methods for the Formal Development of Software

94-10 *Axel Dold*
Formalisierung schematischer Algorithmen

94-11 *Johannes Köbler, Osamu Watanabe*
New Collapse Consequences of NP Having Small Circuits

94-12 *Rainer Schuler*
On Average Polynomial Time

94-13 *Rainer Schuler, Osamu Watanabe*
Towards Average-Case Complexity Analysis of NP Optimization Problems

94-14 *Wolfram Schulte, Ton Vullinghs*
Linking Reactive Software to the X-Window System

94-15 *Alfred Lupper*
Namensverwaltung und Adressierung in Distributed Shared Memory-Systemen

94-16 *Robert Regn*
Verteilte Unix-Betriebssysteme

94-17 *Helmuth Partsch*
Again on Recognition and Parsing of Context-Free Grammars:
Two Exercises in Transformational Programming

94-18 *Helmuth Partsch*
Transformational Development of Data-Parallel Algorithms: an Example

95-01 *Oleg Verbitsky*
On the Largest Common Subgraph Problem

95-02 *Uwe Schöning*
Complexity of Presburger Arithmetic with Fixed Quantifier Dimension

95-03 *Harry Buhrman,Thomas Thierauf*
The Complexity of Generating and Checking Proofs of Membership

95-04 *Rainer Schuler, Tomoyuki Yamakami*
Structural Average Case Complexity

95-05 *Klaus Achatz, Wolfram Schulte*
Architecture Indepentent Massive Parallelization of Divide-And-Conquer Algorithms

95-06 *Christoph Karg, Rainer Schuler*
Structure in Average Case Complexity

95-07 *P. Dadam, K. Kuhn, M. Reichert, T. Beuter, M. Nathe*
ADEPT: Ein integrierender Ansatz zur Entwicklung flexibler, zuverlässiger kooperierender Assistenzsysteme in klinischen Anwendungsumgebungen

95-08 *Jürgen Kehrer, Peter Schulthess*
Aufbereitung von gescannten Röntgenbildern zur filmlosen Diagnostik

95-09 *Hans-Jörg Burtschick, Wolfgang Lindner*
On Sets Turing Reducible to P-Selective Sets

95-10 *Boris Hartmann*
Berücksichtigung lokaler Randbedingung bei globaler Zieloptimierung mit neuronalen Netzen am Beispiel Truck Backer-Upper

95-12 *Klaus Achatz, Wolfram Schulte*
Massive Parallelization of Divide-and-Conquer Algorithms over Powerlists

95-13 *Andrea Mößle, Heiko Vogler*
Efficient Call-by-value Evaluation Strategy of Primitive Recursive Program Schemes

95-14 *Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
A Generic Specification for Verifying Peephole Optimizations

96-01 *Ercüment Canver, Jan-Tecker Gayen, Adam Moik*
Formale Entwicklung der Steuerungssoftware für eine elektrisch ortsbediente Weiche mit VSE

96-02 *Bernhard Nebel*
Solving Hard Qualitative Temporal Reasoning Problems: Evaluating the Efficiency of Using the ORD-Horn Class

96-03 *Ton Vullinghs, Wolfram Schulte, Thilo Schwinn*
An Introduction to TkGofer

96-04 *Thomas Beuter, Peter Dadam*
Anwendungsspezifische Anforderungen an Workflow-Mangement-Systeme am Beispiel der Domäne Concurrent-Engineering

96-05 *Gerhard Schellhorn, Wolfgang Ahrendt*
Verification of a Prolog Compiler - First Steps with KIV

96-06 *Manindra Agrawal, Thomas Thierauf*
Satisfiability Problems

96-07 *Vikraman Arvind, Jacobo Torán*
A nonadaptive NC Checker for Permutation Group Intersection

96-08 *David Cyrluk, Oliver Möller, Harald Rueß*
An Efficient Decision Procedure for a Theory of Fix-Sized Bitvectors with Composition and Extraction

96-09 *Bernd Biechele, Dietmar Ernst, Frank Houdek, Joachim Schmid, Wolfram Schulte*
Erfahrungen bei der Modellierung eingebetteter Systeme mit verschiedenen SA/RT–Ansätzen

96-10 *Falk Bartels, Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
Formalizing Fixed-Point Theory in PVS

96-11 *Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
Mechanized Semantics of Simple Imperative Programming Constructs

96-12 *Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
Generic Compilation Schemes for Simple Programming Constructs

96-13 *Klaus Achatz, Helmuth Partsch*
From Descriptive Specifications to Operational ones: A Powerful Transformation Rule, its Applications and Variants

97-01 *Jochen Messner*
Pattern Matching in Trace Monoids

97-02 *Wolfgang Lindner, Rainer Schuler*
A Small Span Theorem within P

97-03 *Thomas Bauer, Peter Dadam*
A Distributed Execution Environment for Large-Scale Workflow Management Systems with Subnets and Server Migration

97-04 *Christian Heinlein, Peter Dadam*
Interaction Expressions - A Powerful Formalism for Describing Inter-Workflow Dependencies

97-05 *Vikraman Arvind, Johannes Köbler*
On Pseudorandomness and Resource-Bounded Measure

97-06 *Gerhard Partsch*
Punkt-zu-Punkt- und Mehrpunkt-basierende LAN-Integrationsstrategien für den digitalen Mobilfunkstandard DECT

97-07 *Manfred Reichert, Peter Dadam*
$ADEPT_{flex}$ - Supporting Dynamic Changes of Workflows Without Loosing Control

97-08 *Hans Braxmeier, Dietmar Ernst, Andrea Mößle, Heiko Vogler*
The Project NoName - A functional programming language with its development environment

97-09 *Christian Heinlein*
Grundlagen von Interaktionsausdrücken

97-10 *Christian Heinlein*
Graphische Repräsentation von Interaktionsausdrücken

97-11 *Christian Heinlein*
Sprachtheoretische Semantik von Interaktionsausdrücken

97-12 *Gerhard Schellhorn, Wolfgang Reif*
Proving Properties of Finite Enumerations: A Problem Set for Automated Theorem Provers

97-13 *Dietmar Ernst, Frank Houdek, Wolfram Schulte, Thilo Schwinn*
Experimenteller Vergleich statischer und dynamischer Softwareprüfung für eingebettete Systeme

97-14 *Wolfgang Reif, Gerhard Schellhorn*
Theorem Proving in Large Theories

97-15 *Thomas Wennekers*
Asymptotik rekurrenter neuronaler Netze mit zufälligen Kopplungen

97-16 *Peter Dadam, Klaus Kuhn, Manfred Reichert*
Clinical Workflows - The Killer Application for Process-oriented Information Systems?

97-17 *Mohammad Ali Livani, Jörg Kaiser*
EDF Consensus on CAN Bus Access in Dynamic Real-Time Applications

97-18 *Johannes Köbler,Rainer Schuler*
Using Efficient Average-Case Algorithms to Collapse Worst-Case Complexity Classes

98-01 *Daniela Damm, Lutz Claes, Friedrich W. von Henke, Alexander Seitz, Adelinde Uhrmacher, Steffen Wolf*
Ein fallbasiertes System für die Interpretation von Literatur zur Knochenheilung

98-02 *Thomas Bauer, Peter Dadam*
Architekturen für skalierbare Workflow-Management-Systeme - Klassifikation und Analyse

98-03 *Marko Luther, Martin Strecker*
A guided tour through *Typelab*

98-04 *Heiko Neumann, Luiz Pessoa*
Visual Filling-in and Surface Property Reconstruction

98-05 *Ercüment Canver*
Formal Verification of a Coordinated Atomic Action Based Design

98-06 *Andreas Küchler*
On the Correspondence between Neural Folding Architectures and Tree Automata

98-07 *Heiko Neumann, Thorsten Hansen, Luiz Pessoa*
Interaction of ON and OFF Pathways for Visual Contrast Measurement

98-08 *Thomas Wennekers*
Synfire Graphs: From Spike Patterns to Automata of Spiking Neurons

98-09 *Thomas Bauer, Peter Dadam*
Variable Migration von Workflows in *ADEPT*

98-10 *Heiko Neumann, Wolfgang Sepp*
Recurrent V1 – V2 Interaction in Early Visual Boundary Processing

98-11 *Frank Houdek, Dietmar Ernst, Thilo Schwinn*
Prüfen von C–Code und Statmate/Matlab–Spezifikationen: Ein Experiment

98-12 *Gerhard Schellhorn*
Proving Properties of Directed Graphs: A Problem Set for Automated Theorem Provers

98-13 *Gerhard Schellhorn, Wolfgang Reif*
Theorems from Compiler Verification: A Problem Set for Automated Theorem Provers

98-14 *Mohammad Ali Livani*
SHARE: A Transparent Mechanism for Reliable Broadcast Delivery in CAN

98-15 *Mohammad Ali Livani, Jörg Kaiser*
Predictable Atomic Multicast in the Controller Area Network (CAN)

99-01 *Susanne Boll, Wolfgang Klas, Utz Westermann*
A Comparison of Multimedia Document Models Concerning Advanced Requirements

99-02 *Thomas Bauer, Peter Dadam*
Verteilungsmodelle für Workflow-Management-Systeme - Klassifikation und Simulation

99-03 *Uwe Schöning*
On the Complexity of Constraint Satisfaction

99-04 *Ercument Canver*
Model-Checking zur Analyse von Message Sequence Charts über Statecharts

99-05 *Johannes Köbler, Wolfgang Lindner, Rainer Schuler*
Derandomizing RP if Boolean Circuits are not Learnable

99-06 *Utz Westermann, Wolfgang Klas*
Architecture of a DataBlade Module for the Integrated Management of Multimedia Assets

99-07 *Peter Dadam, Manfred Reichert*
Enterprise-wide and Cross-enterprise Workflow Management: Concepts, Systems, Applications. Paderborn, Germany, October 6, 1999, GI–Workshop Proceedings, Informatik ’99

99-08 *Vikraman Arvind, Johannes Köbler*
Graph Isomorphism is Low for ZPP^{NP} and other Lowness results

99-09 *Thomas Bauer, Peter Dadam*
Efficient Distributed Workflow Management Based on Variable Server Assignments

2000-02 *Thomas Bauer, Peter Dadam*
Variable Serverzuordnungen und komplexe Bearbeiterzuordnungen im Workflow-Management-System ADEPT

2000-03 *Gregory Baratoff, Christian Toepfer, Heiko Neumann*
Combined space-variant maps for optical flow based navigation

2000-04 *Wolfgang Gehring*
Ein Rahmenwerk zur Einführung von Leistungspunktsystemen

2000-05 *Susanne Boll, Christian Heinlein, Wolfgang Klas, Jochen Wandel*

Intelligent Prefetching and Buffering for Interactive Streaming of MPEG Videos

2000-06 *Wolfgang Reif, Gerhard Schellhorn, Andreas Thums*
Fehlersuche in Formalen Spezifikationen

2000-07 *Gerhard Schellhorn, Wolfgang Reif (eds.)*
FM-Tools 2000: The 4th Workshop on Tools for System Design and Verification

2000-08 *Thomas Bauer, Manfred Reichert, Peter Dadam*
Effiziente Durchführung von Prozessmigrationen in verteilten Workflow-Management-Systemen

2000-09 *Thomas Bauer, Peter Dadam*
Vermeidung von Überlastsituationen durch Replikation von Workflow-Servern in ADEPT

2000-10 *Thomas Bauer, Manfred Reichert, Peter Dadam*
Adaptives und verteiltes Workflow-Management

2000-11 *Christian Heinlein*
Workflow and Process Synchronization with Interaction Expressions and Graphs

2001-01 *Hubert Hug, Rainer Schuler*
DNA-based parallel computation of simple arithmetic

2001-02 *Friedhelm Schwenker, Hans A. Kestler, Günther Palm*
3-D Visual Object Classification with Hierarchical Radial Basis Function Networks

2001-03 *Hans A. Kestler, Friedhelm Schwenker, Günther Palm*
RBF network classification of ECGs as a potential marker for sudden cardiac death

2001-04 *Christian Dietrich, Friedhelm Schwenker, Klaus Riede, Günther Palm*
Classification of Bioacoustic Time Series Utilizing Pulse Detection, Time and Frequency Features and Data Fusion

2002-01 *Stefanie Rinderle, Manfred Reichert, Peter Dadam*
Effiziente Verträglichkeitsprüfung und automatische Migration von Workflow-Instanzen bei der Evolution von Workflow-Schemata

2002-02 *Walter Guttmann*
Deriving an Applicative Heapsort Algorithm

2002-03 *Axel Dold, Friedrich W. von Henke, Vincent Vialard, Wolfgang Goerigk*
A Mechanically Verified Compiling Specification for a Realistic Compiler

2003-01 *Manfred Reichert, Stefanie Rinderle, Peter Dadam*
A Formal Framework for Workflow Type and Instance Changes Under Correctness Checks

2003-02 *Stefanie Rinderle, Manfred Reichert, Peter Dadam*
Supporting Workflow Schema Evolution By Efficient Compliance Checks

2003-03 *Christian Heinlein*
Safely Extending Procedure Types to Allow Nested Procedures as Values

2003-04 *Stefanie Rinderle, Manfred Reichert, Peter Dadam*
On Dealing With Semantically Conflicting Business Process Changes.

2003-05 *Christian Heinlein*
Dynamic Class Methods in Java

2003-06 *Christian Heinlein*
Vertical, Horizontal, and Behavioural Extensibility of Software Systems

2003-07 *Christian Heinlein*
Safely Extending Procedure Types to Allow Nested Procedures as Values
(Corrected Version)

2003-08 *Changling Liu, Jörg Kaiser*
Survey of Mobile Ad Hoc Network Routing Protocols)

2004-01 *Thom Frühwirth, Marc Meister (eds.)*
First Workshop on Constraint Handling Rules

2004-02 *Christian Heinlein*
Concept and Implementation of C+++, an Extension of C++ to Support User-Defined Operator Symbols and Control Structures

2004-03 *Susanne Biundo, Thom Frühwirth, Günther Palm(eds.)*
Poster Proceedings of the 27th Annual German Conference on Artificial Intelligence

2005-01 *Armin Wolf, Thom Frühwirth, Marc Meister (eds.)*
19th Workshop on (Constraint) Logic Programming

2005-02 *Wolfgang Lindner (Hg.), Universität Ulm , Christopher Wolf (Hg.) KU Leuven*
2. Krypto-Tag – Workshop über Kryptographie, Universität Ulm

2005-03 *Walter Guttmann, Markus Maucher*
Constrained Ordering

2006-01 *Stefan Sarstedt*
Model-Driven Development with ACTIVECHARTS, Tutorial

2006-02 *Alexander Raschke, Ramin Tavakoli Kolagari*
Ein experimenteller Vergleich zwischen einer plan-getriebenen und einer leichtgewichtigen Entwicklungsmethode zur Spezifikation von eingebetteten Systemen

2006-03 *Jens Kohlmeyer, Alexander Raschke, Ramin Tavakoli Kolagari*
Eine qualitative Untersuchung zur Produktlinien-Integration über Organisationsgrenzen hinweg

2006-04 *Thorsten Liebig*
Reasoning with OWL - System Support and Insights –

2008-01 *H.A. Kestler, J. Messner, A. Müller, R. Schuler*
On the complexity of intersecting multiple circles for graphical display

Ulmer Informatik-Berichte
ISSN 0939-5091

Herausgeber:
Universität Ulm
Fakultät für Ingenieurwissenschaften und Informatik
89069 Ulm