



Clarifying the Structure of Complex Venn Diagrams

Susmita Ray Peter Peelman * Jan Paredaens

Dept. of Mathematics and Computer Science[†]
University of Antwerp (UIA)
Antwerp, Belgium

1. Introduction

Venn diagrams serve as the basis of visual representation systems, because of their ability to express set relationships between groups of information. Traditionally, the emptiness of a data set is indicated by *shading* the region of the diagram that represents it. In this paper, we describe a method for transforming such a shaded Venn diagram into an equivalent one without shadings, or with a minimal number of shadings. This technique greatly simplifies the complexity of large Venn diagrams while keeping their visual representation properties intact.

We show that it is not always possible to remove all shaded regions of a Venn diagram — so the best one can do is to remove as much of them as possible. The algorithm for reducing the shadings of a Venn diagram is presented in two parts. First, we describe a method that, given a Venn diagram in which all shaded regions lie inside one of the curves of the diagram, tests whether or not all these regions can be removed. Afterwards, we use this method in a heuristic algorithm that solves the general problem of reducing as many as possible shadings of a Venn diagram. This short paper only states the problem, specifies the algorithm and gives an example. The full paper is available as Technical Report 94-08, Dept. Math. & Computer Sci., University of Antwerp (UIA).

*Supported by grant of “Programmatie van het Wetenschapsbeleid, IT/IF/13.”

[†]Address: Universiteitsplein 1, B-2610 Antwerpen, Belgium. E-mail: {sray,peelman,pareda}@wins.uia.ac.be.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association of Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

AVI 94-6/94 Bari Italy

© 1994 ACM 0-89791-733-2/94/0010..\$3.50

2. Venn Diagrams in Visual Representation System

A Venn diagram for n sets A_1, \dots, A_n is a family of n curves C_1, \dots, C_n , that divide the plane in exactly 2^n regions, each one representing a set of the form $X_1 \cap \dots \cap X_n$, where each X_i is either A_i or $\neg A_i$. Venn diagrams are a useful tool in set-theoretic applications of discrete mathematics. They were introduced by John Venn in 1880, and give rise to various interesting geometric problems.

In this paper we investigate how a given Venn diagram can be visualized as simply as possible. In this paper we look at Venn diagrams from the viewpoint of expressing information. Since in a standard Venn diagram of n sets all possible relationships between any combination of these sets are visualized, many regions of the diagram will be empty. Traditionally, the emptiness of such regions is indicated by shading them. However, the visual attractiveness of a Venn diagram can be augmented by drawing the diagram in such a way that the empty regions are left out.

Consider the following example. Suppose we want to draw a Venn diagram about 4 sets of people. Set A contains all people, B is the set of all married people, C the set of all divorced people, and D is the set of all retired people. We know that B , C and D are subsets of A , so in a standard Venn diagram the regions belonging to $B \setminus A$, $C \setminus A$, and $D \setminus A$ are shaded. Also since one cannot at the same time be married and divorced, sets B and C have an empty intersection. The traditional way to represent this situation with a Venn diagram is given in top diagram of Figure 1.

However, the same situation can also be represented by the diagram at the bottom of Figure 1. In this diagram the facts that B , C and D are subsets

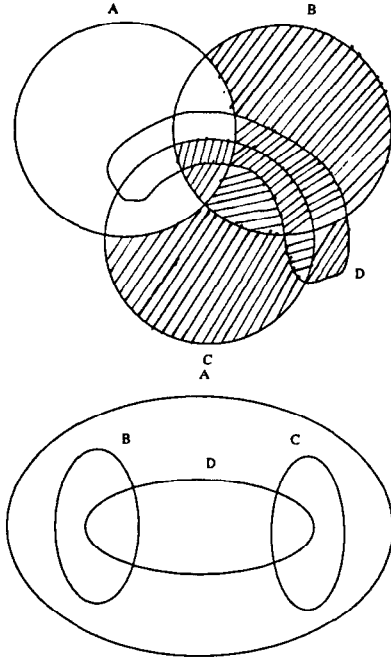


Figure 1: A Venn diagram with shadings and an equivalent one

of A , and that B and C are disjoint, are visualized much better than in the original Venn diagram.

In a practical application of a visual representation system, a user will ask for a Venn diagram for n sets, and which will then be generated by the application. Then the user will indicate the shaded regions (Figure 1,top), and the system will redraw the original Venn diagram in such a way that the shaded regions are left out (Figure 1,bottom). Hence, an algorithm is needed which transforms a Venn diagram with shadings to an aesthetically pleasant looking one without shadings. The process of transforming a Venn diagram to one with fewer (or without) shadings is called *shading reduction* and the diagram obtained after it is referred to as the *reduced diagram* which we define as *Loose Venn Diagram*. A *Loose Venn Diagram* is allowed to have fewer curves and regions compared to the source diagram with shadings from which it has been derived. Also, a *Loose Venn Diagram* of n curves is not restricted to contain 2^n regions. It is allowed to have more than two curves passing through the same point provided that every curve has a different direction coefficient at that point

of intersection. Every constituent curve of a *Loose Venn Diagram* should be *closed*, *differentiable* and *non-self-intersecting*.

3. Reduction of Shaded Regions inside a Curve

The problem of removing all shaded regions of a Venn diagram is quite complicated. For this reason we first solve following subproblem of it. "Given a shaded Venn diagram V and a curve a of V , is it possible to reduce V in such a way that a does not contain any shaded regions?"

Example 3.1

Let us take an example situation as shown in the *Loose Venn diagram* V (the topleft picture) of Figure 2 where curve a has 2 shadings inside it. We intend to reduce all shadings inside curve a . At the first step algorithm **CurveSubtract** draws the diagram V' as shown in (topright picture) of Figure 2 by removing a from V and derives : $IR = \{2, 3, 4, 5, 8, 6\}$ and $ER = \{\}$, where IR denotes the set of regions of V' that were intersected by a and as such were composed of 2 unshaded regions in V and ER denotes the set of regions of V' that were totally included in a .

At the second step the *boolean* function **Can-CurveRedraw()** tests the possibility of redrawing a as a *closed*, *differentiable*, *non-self-intersecting* curve in V' , taking into account the specified IR and ER for it. It further produces the list LPP of *Passpoints* and the set SEP of *enclosed points* of the curve when drawing is possible as : $LPP : \langle p_1, p_4 \rangle$ and $SEP : \{p_5\}$.

At the third step using the LPP and SEP the algorithm **CurveAdd** builds the diagram V_{new} (as shown in the picture at the bottom of Figure 2) by drawing the modified a in V' .

4. Strategy of Shading Reduction

Given a Venn diagram V with a number of shaded regions we explain the method of removing shadings from it so as to obtain a diagram with as few shadings as possible.

For a Venn diagram V of n curves and m shadings we find out the number of shaded regions n_c inside each curve c_i $\{i = 1, 2, \dots, n\}$. We then pick

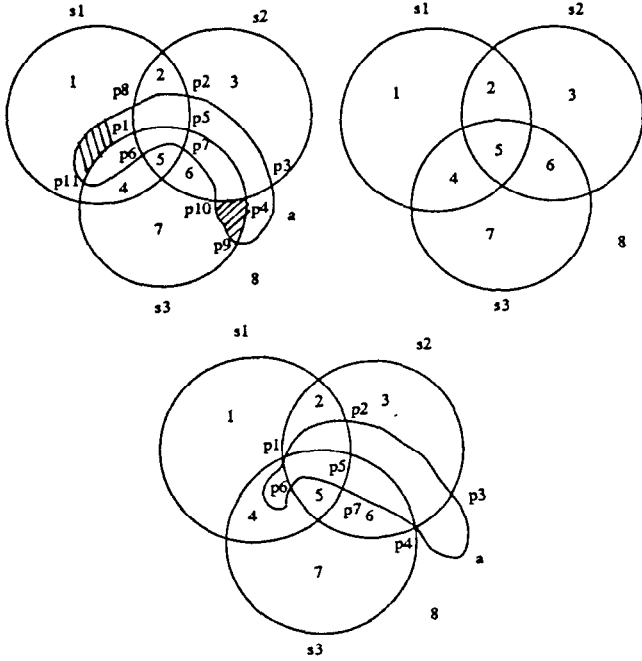


Figure 2: Loose Venn diagrams V , V' and V_{new}

out the curve (say, c_k) having the maximum number m_x of shaded regions (inside it) and eliminate the curve from V and obtain V' using **CurveSubtract**. In V' we now try to redraw a modified c_k using **CanCurveRedraw()** such that there will be no shaded regions inside it. If redrawing is possible we then redraw the modified c_k in V' using **CurveAdd** and thereby obtain a new diagram V_{new} and modify shadings in the context of V_{new} . We then rename V_{new} as V and go back to recalculate n_c for each curve and follow as before all the steps. However if redrawing of c_k is found to be impossible then everytime we reduce m_x by 1 we check all curves and try to find out the one for which it is possible to reduce all the m_x shadings inside it and the moment we get such a curve we redraw it as before and again go back to first step as before (in recalculating n_c). Thus the process is repeated till no shadings are left or no further shading reduction is possible.

Example 4.1

Let us try to remove the shadings from the top-most Venn diagram of Figure 3. We find that, $n_A = 3$, $n_B = 1$, $n_C = 2$, $n_D = 3$. So, according to our strategy we may try to reduce shadings

inside either A or D . Let us choose D and reduce shadings inside it to get the second(from top) diagram of Figure 3. Now in this diagram we find that, $n_A = 1$, $n_B = 0$, $n_C = 1$, $n_D = 0$. Now we may choose to reduce shadings inside A or C . We choose A and removing shadings inside A we obtain the lowermost diagram of Figure 3 in which $n_c = 1$ and $n_A = n_B = n_D = 0$. But it is no more possible to remove shadings inside c and hence the final reduced diagram (the lowermost one) of Figure 3 still contains one shading.

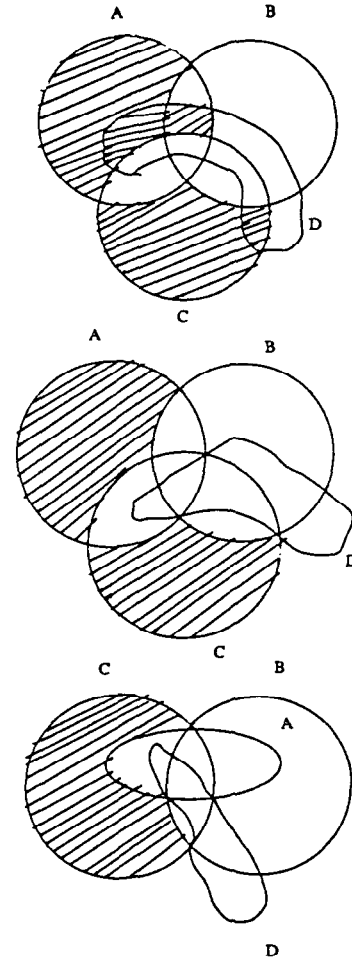


Figure 3: A Venn diagram with shadings and the ones after removing shading from D and A