finding the correct results for each operation. Furthermore, computing the result should be robust and efficient.

The solution we choose uses a planar graph as the underlying data structure, which allows an easy traversing of the result. This data structure contains vertices, edges, faces, and holes, and each of these element types has a color. The result needs neither necessarily be connected nor is it guaranteed that there are no dangling elements.

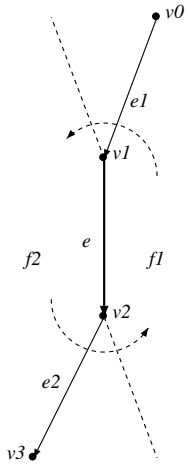
**The Algorithm:**

In order to perform boolean operations on simple polygons we use the following strategy:

1. Find all intersections by doing a sweep-line over the edges of the polygons ([MS88], [MN94], [BO79], [PS85]).
2. Construct a planar graph by inserting all edges and all intersections.
3. Traverse this planar graph in order to perform an intersection, union, or difference.

In this paper we will concentrate on the second and third step, constructing a planar graph from two simple polygons, and traversing it in order to get the respective result. For that reason we have to choose a suitable representation or data structure of a planar graph: the doubly connected edge list (DCEL).

This data structure contains the following elements: faces, edges, and vertices. Each element is able to store additional information like a color. This data structure also allows an easy and fast access of all elements and the insertion of polygons with associated colors.




| E | V1 | V2 | F1 | F2 | P1 | P2 |
|---|---|---|---|---|---|---|
| | | | | | | |
| e | v1 | v2 | f1 | f2 | e1 | e2 |
| | | | | | | |
| e1 | v0 | v1 | f1 | f2 | | e |
| | | | | | | |
| e2 | v2 | v3 | f1 | f2 | e | |
| | | | | | | |

Figure 1: Illustration of the DCEL

## 1.1 A DCEL Represents a Planar Graph

Let $V = \{v_1, v_2, \ldots, v_n\}$ be a set of vertices and $E = \{e_1, e_2, \ldots, e_m\}$ be a set of edges. A planar graph $G = (V, E)$, not necessarily connected, can be represented in different ways. One straightforward representation would be to use a list of vertices $V$ and a list of edges $E$, whereas each element $V$ contains the vertex coordinates and each element of $E$ represents an edge, which contains its source-vertex and its destination-vertex. The main disadvantage of this representation is that an efficient traversal of the graph is not possible without additional structures.