SPATIAL COGNITION AND COMPUTATION, *vol*(issue), 1–end

On Embedding a Qualitative Representation in a Two-Dimensional Plane

Kazuko TAKAHASHI
Kwansei Gakuin University
Takao SUMITOMO†
Kwansei Gakuin University
Izumi TAKEUTI
National Institute of Advanced Industrial Science and Technology

This paper discusses embedding in a two-dimensional plane a symbolic representation for spatial data using the simple objects, $points(P), lines(L), circuits(C)$ and $areas(A)$. We have proposed PLCA as a new framework for a qualitative spatial reasoning. In a PLCA expression, the entire figure is represented in a form in which all the objects are related. We investigate the conditions for two-dimensional realizability of a PLCA expression, and derive the relation that the numbers of objects in a PLCA expression should have. In this process, we use the well-known Euler's formula. We also give an algorithm for drawing the figure of the PLCA expression that satisfies this condition in a two-dimensional plane and prove its correctness. The algorithm generates a quantitative expression from qualitative expression.
Keywords: qualitative spatial reasoning, planar graph, graph theory, spatial database

1 Introduction

Compared to textual data, spatial data contains more abundant information and provides a representation that is easy for various users to understand, but these

† Currently, Asahi Intelligence Service Co.,LTD.
Correspondence concerning this article should be addressed to Kazuko TAKAHASHI, School of Science and Technology, Kwansei Gakuin University, 2-1, Gakuen, Sanda, Hyogo, 669-1337, JAPAN; email ktaka@kwansei.ac.jp

data require significant memory and space. However, highly-detailed data are not always neccessary depending on the user's purpose. It is sometimes sufficient to know the number of objects or the positional relationships of the objects in a figure. Qualitative Spatial Reasoning (QSR) is a method that treats spatial data symbolically, not numerically, by extracting only the information necessary for a user's purpose. It is useful for the recognition and analysis of physical phenomena, explanation of a causality, diagnosis, and so on (Cohn & Hazarika, 2001).

In QSR, a space is usually represented as a set of relative positional relationships between objects such as regions or lines. So far, various QSR systems have been proposed (Borgo, Guarino, & Masolo, 1996)(Cohn & Varzi, 1998)(Egenhofer & Franzosa, 1995)(Randell, Cui, & Cohn, 1992)(Renz, 2002)(Stock(ed.), 1997). In most systems, the spatial relationships of objects are represented using predicates, and axioms on these predicates are introduced. For example, consider several relationships between regions α and β as shown in Figure 1. In most systems, these relationships are represented by the binary relation of α and β. This means that many predicates and axioms are required to distinguish these figures, making the system not feasible and hard to implement. On the other hand, the system is inapplicable to real problems if the classification is too coarse, such as if we regard, (a),(b) and (c) as the same relation of "connected."

We have proposed PLCA to solve these problems (Sumitomo & Takahashi, 2004)(Takahashi & Sumitomo, 2005). PLCA provides a symbolic representation for spatial data using the simple objects: $points(P), lines(L), circuits(C)$ and $areas(A)$. No pair of areas has a part in common[1]. The entire space is covered with the areas. The four kinds of the objects are used to represent "a region." The figures in Figure 1 are distinguished not by the relationships between regions but by the objects that are used to represent these regions. If two regions are connected, then they have the same points and/or lines in common, and these points and/or lines are different depending on the connection patterns. Therefore, we can distinguish the connection patterns by checking the common elements without introducing more predicates nor axioms. Symbolic representation allows rapid processing and enables the compaction of information to a level that is suitable for the user's purpose.

For a given figure in a two-dimensional plane, there exists a PLCA expression, that is unique in its pattern of connections among regions. Generating a PLCA expression from a figure transforms a quantitative representation into a qualitative one. We have implemented a prototype system (Takahashi & Sumitomo, 2007).

However, it is difficult to generate a figure from a PLCA expression, since we have to supplement the missing part of the data. A representation for spatial data using objects related to each other is frequently used in computational geometry (de Berg, van Kreveld, & Overmars, de BergETALqsr:BKOS97) and Geo-

[1]We use the term $area$ instead of $region$, since $area$ used in this paper is a different entity from the $region$ generally used in qualitative spatial reasoning.

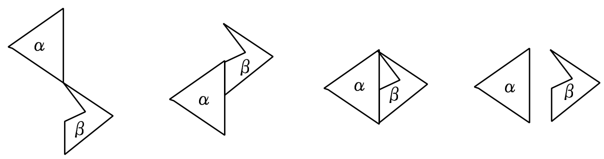


Figure 1.: Relationships between regions

graphic Information Systems (Thurston, Poiker, & Moore, 2003). Different from these representations which have a one-to-one relation with a figure, a quantitative representation has to be generated from a qualitative one in PLCA. Moreover, it has not been clarified whether a figure can be drawn in a two-dimensional plane for any PLCA expression.

In this paper, we investigate the conditions for the two-dimensional realizability of a PLCA expression and show that we can determine the two-dimensional realizability by examining the number of the objects contained in the expression. We prove this condition using Euler's formula. We also give an algorithm for drawing the figure in a two-dimensional plane for a PLCA expression that satisfies this condition. Well-known concepts and existing algorithms in graph theory are used to construct the algorithms and proofs.

This paper is organized as follows. In section 2, we describe the PLCA expression. In section 3, we discuss two-dimensional realizability of a PLCA expression. In section 4, we show the algorithm for drawing the figure in a two-dimensional plane corresponding to a PLCA expression. In section 5, we compare drawing for PLCA with those for other QSR systems. Finally, in section 6, we show the conclusion.

2 PLCA Expressions

2.1 Definition of Classes

PLCA has four basic components: $points(P)$, $lines(L)$, $circuits(C)$ and $areas(A)$.

Point is defined as a primitive class.

Line is defined as a class that satisfies the following condition: for an arbitrary instance l of $Line$, $l.points$ is a pair $[p_1, p_2]$ where $p_1, p_2 \in Point$. A line has an inherent orientation. When $l = [p_1, p_2]$, l^+ and l^- mean $[p_1, p_2]$ and $[p_2, p_1]$, respectively. l^* denotes either l^+ or l^-. Intuitively, a line is the edge connecting two (not always different) points. No two lines are allowed to cross.

Circuit is defined as a class that satisfies the following condition: for an arbitrary instance c of $Circuit$, $c.lines$ is a sequence $[l_0^*, \ldots, l_n^*]$ where $l_0, \ldots, l_n \in$

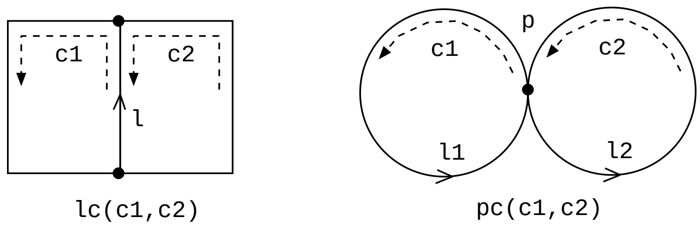


Figure 2.: Line-connected and point-connected

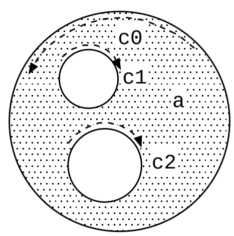


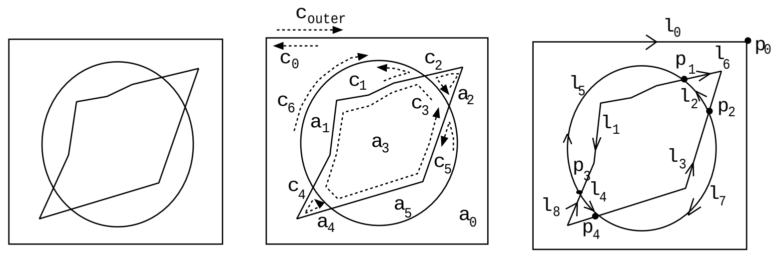
Figure 3.: An area

$Line(n \geq 0)$, $l_i.points = [p_i, p_{i+1}](0 \leq i \leq n)$ and $p_{n+1} = p_0$. $[l_0^*, \ldots, l_n^*]$ and $[l_j^*, \ldots, l_n^*, l_0^*, \ldots, l_{j-1}^*]$ denote the same circuit for any j $(0 \leq j \leq n)$.

For $c_1, c_2 \in Circuit$, we introduce two new predicates lc and pc to denote that two circuits share line(s) and point(s), respectively. $lc(c_1, c_2)$ is $true$ iff there exists $l \in Line$ such that $(l^+ \in c_1.lines) \wedge (l^- \in c_2.lines)$. $pc(c_1, c_2)$ is $true$ iff there exists $p \in Point$ such that $(p \in l_1.points) \wedge (p \in l_2.points) \wedge (l_1^* \in c_1.lines) \wedge (l_2^* \in c_2.lines)$. For example, for the circuits c_1 and c_2 in Figure 2, $lc(c_1, c_2)$ holds in (a) while $pc(c_1, c_2)$ holds in (b). A circuit is the boundary between an area and its adjacent areas viewed from the side of that area. We say that p *is on* c if there exists l such that $p \in l.points \wedge l^* \in c.lines$.

Area is defined as a class that satisfies the following condition: for an arbitrary instance a of $Area$, $a.circuits$ is a set $\{c_0, \ldots, c_n\}$ where $c_0, \ldots, c_n \in Circuit(n \geq 0)$, and $\forall c_i, c_j \in a.circuits; (i \neq j) \rightarrow (\neg pc(c_i, c_j) \wedge \neg lc(c_i, c_j))$. Intuitively, an area is a connected region which consists of exactly one piece that may have a hole. No two areas are allowed to cross. The final condition means that any pair of circuits that belong to the same area cannot share a point or a line. For example, in Figure 3, the hatched area a has three circuits: $a.circuits = \{c_0, c_1, c_2\}$, all of which are $\neg pc$ and $\neg lc$ with each other.

The *PLCA expression* e is defined as a five tuple $e = \langle P, L, C, A, outermost \rangle$



(a) target figure (b) areas and circuits (c) lines and points

Figure 4.: Example of a target figure

where P, L, C and A are a set of points, lines, circuits and areas, respectively, and $outermost \in C$. An element of $P \cup L \cup C \cup A$ is called *a component of* e.

We assume that there exists a circuit in the outermost side of the figure that is called $outermost$. It means that the target figure is drawn in a finite space, and the space can be divided into a number of areas which do not overlap with each other.

In Figure 4, (a) shows an example of a target figure, (b) and (c) show the names of objects. In Figure 5, we show a PLCA expression corresponding to Figure 4.

$Example$ 1.

$e.points = \{p_0, p_1, p_2, p_3, p_4\}$
$e.lines = \{l_0, l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8\}$
$e.circuits = \{c_{outer}, c_0, c_1, c_2, c_3, c_4, c_5, c_6\}$
$e.areas = \{a_0, a_1, a_2, a_3, a_4, a_5\}$
$e.outermost = c_{outer}$
$l_0.points = [p_0, p_0]$
$l_1.points = [p_1, p_3]$
$l_2.points = [p_2, p_1]$
$l_3.points = [p_4, p_2]$
$l_4.points = [p_3, p_4]$
$l_5.points = [p_3, p_1]$
$l_6.points = [p_1, p_2]$
$l_7.points = [p_2, p_4]$
$l_8.points = [p_4, p_3]$

$c_{outer}.lines = [l_0^+]$
$c_0.lines = [l_0^-]$
$c_1.lines = [l_1^-, l_5^-]$
$c_2.lines = [l_2^-, l_6^-]$
$c_3.lines = [l_4^+, l_3^+, l_2^+, l_1^+]$
$c_4.lines = [l_4^-, l_8^-]$
$c_5.lines = [l_3^-, l_7^-]$
$c_6.lines = [l_5^+, l_8^+, l_7^+, l_6^+]$
$a_0.circuits = \{c_6, c_0\}$
$a_1.circuits = \{c_1\}$
$a_2.circuits = \{c_2\}$
$a_3.circuits = \{c_3\}$
$a_4.circuits = \{c_4\}$
$a_5.circuits = \{c_5\}$

Figure 5.: PLCA expression for Figure 4

Definition 2.1 *(consistency) For PLCA expression* $e = \langle P, L, C, A, outermost\rangle$*, if the following three constraints are satisfied, then it is said to be consistent.*

1. **constraint on P-L:** *For any* $p \in Point$ *there exists at least one line* l *such that* $p \in l.points$*.*
2. **constraint on L-C:** *For any* $l \in Line$*, there exist exactly two distinct circuits* c_1, c_2 *such that* $l^+ \in c_1.lines, l^- \in c_2.lines$*.*
3. **constraint on C-A:** *For any* $c \in Circuit$ *other than* $outermost$*, there exists exactly one area* a *such that* $c \in a.circuits$*. The* $outermost$ *is not included in any area.*

Due to these constraints, neither isolated lines nor points are allowed.

2.2 PLCA Equivalence

For consistent PLCA expressions $e_1 = \langle P_1, L_1, C_1, A_1, outermost_1\rangle$ and $e_2 = \langle P_2, L_2, C_2, A_2, outermost_2\rangle$, if there exists a bijective mapping f from e_1 to e_2, that satisfies the following conditions, then e_1 and e_2 are said to be *PLCA-equivalent*.

For $\forall p \in P_1,\ f(p) \in P_2$
For $\forall l \in L_1,\ f(l) \in L_2$
For $\forall c \in C_1,\ f(c) \in C_2$
For $\forall a \in A_1,\ f(a) \in A_2$
For $\forall l \in L_1,\ f(l.points) = f(l).points$
For $\forall c \in C_1,\ f(c.lines) = f(c).lines$
For $\forall a \in A_1,\ f(a.circuits) = f(a).circuits$

2.3 Operations on Lines

Definition 2.2 *(redundant expression, minimum expression)*
Let $e = \langle P, L, C, A, outermost\rangle$ *be a consistent PLCA expression. For a point* $p \in P$*, if the number of lines such that* $p \in l.points$ *is two, then* p *is said to be* a redundant point*. If* e *contains a redundant point, it is said to be a redundant expression; otherwise, it is said to be the minimum expression.*

Several operations are defined on a PLCA expression, including line division and line combination (Figure 6).

Line division divides the designated line. It is defined as follows:

1. Select a line $l.points = [p_1, p_2]$ to be divided.
2. Assume that $c_1.lines = \{l_0^*, \ldots, l_{i-1}^*, l^+, l_i^*, \ldots, l_n^*\}$ and $c_2.lines = \{m_0^*, \ldots, m_{j-1}^*, l^-, m_j^*, \ldots, m_k^*\}$.

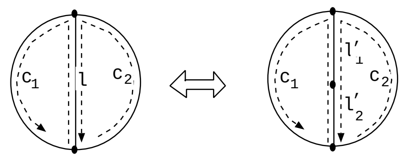


Figure 6.: Line division and line combination

3. Set $l'_1.points = [p_1, p]$ and $l'_2.points = [p, p_2]$.
4. Set $L = L - \{l\} \cup \{l'_1, l'_2\}$ and set $P = P \cup \{p\}$.
5. Replace $c_1.lines$ and $c_2.lines$ by $\{l^*_0, \ldots, l^*_{i-1}, l'^+_1, l'^+_2, l^*_i, \ldots, l^*_n\}$ and $\{m^*_0, \ldots, m^*_{j-1}, l'^-_2, l'^-_1 m^*_j, \ldots, m^*_k\}$, respectively.

Line combination combines two adjacent lines that share a redundant point. The operation is the opposite of line division.

1. Select two lines $\{l'_1, l'_2\}$ where $l'_1.points = [p_1, p]$ and $l'_2.points = [p, p_2]$.
2. Assume that $c_1.lines = \{l^*_0, \ldots, l^*_{i-1}, l'^+_1, l'^+_2, l^*_i, \ldots, l^*_n\}$ and $c_2.lines = \{m^*_0, \ldots, m^*_{j-1}, l'^-_2, l'^-_1, m^*_j, \ldots, m^*_k\}$.
3. Set $l.points = [p_1, p_2]$.
4. Set $L = L - \{l'_1, l'_2\} \cup \{l\}$ and set $P = P - \{p\}$.
5. Replace $c_1.lines$ and $c_2.lines$ by $\{l^*_0, \ldots, l^*_{i-1}, l^+, l^*_i, \ldots, l^*_n\}$ and $\{m^*_0, \ldots, m^*_{j-1}, l^-, m^*_j, \ldots, m^*_k\}$, respectively.

A minimal expression can be converted to the redundant expression by line division, and the redundant expression can be converted to a minimum one by line combination.

3 Two-Dimensional Realizability

We investigate a condition in which a PLCA expression can be realized in a two-dimensional plane.

3.1 Concepts from Graph Theory

As a preparation, we introduce several concepts from graph theory.

A (non-directed) graph is defined to be $G = (V, E)$, where V is a set of vertices and E is a set of edges. An edge of E is defined as a pair of vertices of V. For graphs $G = (V, E)$ and $G' = (V', E')$, if $V' \subset V$ and $E' \subset E$, G' is said to be *a subgraph* of G; if $V \cap V' = \phi$ and $E \cap E' = \phi$, it is said that G and G' are *disjoint*. Here, when we consider more than one subgraph of G, we assume that they are disjoint. If there is no edge (v, v), and for any pair of different vertices, if there exists a unique edge that connects them, the graph is said to be *simple*. A graph that can be embedded in a plane so that no edges intersect is said to be *a planar graph*. If it is possible to move between any pair of vertices by moving along the edges of the graph, the graph is said to be *connected*; otherwise, it is said to be *disconnected*. A connected planar graph divides a plane into a number of regions, which are called *faces*. Note that faces include the outer infinitely large region. A sequence $(v_0, \ldots, v_n)$ where (v_i, v_{i+1}) for each i ($0 \leq i \leq n-1$) is an edge is said to be *a path*, and if $v_0 = v_n$, it is said to be *a cycle*. A cycle that is a border from the graph and the outer infinitely large region is said to be *an outer boundary cycle*.

For a connected planar graph, the following theorem holds (Chartland & Lesniak, 1996).

Theorem 3.1 *(Euler's formula) Let G be a connected planar graph. Let V_G, E_G, F_G be the numbers of the vertices, edges and faces of G. The following relation holds:*

$$V_G - E_G + F_G = 2$$

3.2 Mapping to Graph Expression

Let $e = \langle P, L, C, A, outermost\rangle$ be a consistent PLCA expression. We can define a non-directed graph $m(e) = (V, E)$ by relating P and L to V and E, respectively.

For $p \in P$, $m(p)$ denotes the corresponding vertex, and for $l \in L$, $m(l)$ denotes the corresponding edge. We extend m so that c is mapped to $m(c)$. For each $l_i (i = 0, \ldots, n)$ such that $l_i^* \in c.lines$, if $m(l_i)$ is contained in a graph G, then we say that $m(c)$ *is contained in* G.

3.3 PLCA Connectedness

We introduce the connectedness of the components of a PLCA expression.

Definition 3.1 *(d-pcon) Let $e = \langle P, L, C, A, outermost\rangle$ be a PLCA expression. For a pair of components of e, the predicate $d\text{-}pcon$ is defined as follows.*

1. $d\text{-}pcon(p, l)$ *iff* $p \in l.points$.
2. $d\text{-}pcon(l, c)$ *iff* $l^* \in c.lines$.
3. $d\text{-}pcon(c, a)$ *iff* $c \in a.circuits$.

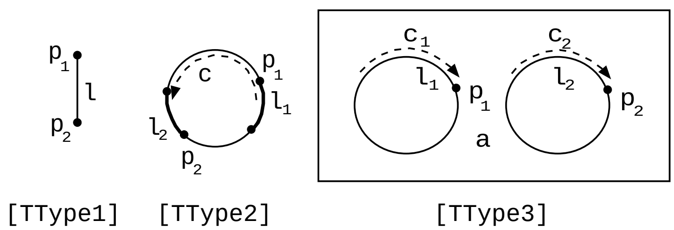


Figure 7.: A trail between points in PLCA

Definition 3.2 *(PLCA trail) A sequence* $(\alpha_0 \ldots, \alpha_n)$ *where* $d\text{-}pcon(\alpha_i\, \alpha_{i+1})$ *or* $d\text{-}pcon(\alpha_{i+1}, \alpha_i)$ *holds for each* i $(0 \leq i \leq n-1)$*, and* $\alpha_i \neq \alpha_j$ *for each* i, j $(0 \leq i < j \leq n)$*, is said to be a PLCA trail from* α_0 *to* α_n*.*

Definition 3.3 *(pcon) Let* α, β *and* γ *be components of a PLCA expression. The predicate* $pcon$ *is defined as a relation that satisfies the following properties.*

1. *If* $d\text{-}pcon(\alpha, \beta)$*, then* $pcon(\alpha, \beta)$*.*
2. *If* $pcon(\alpha, \beta)$*, then* $pcon(\beta, \alpha)$*.*
3. *If* $pcon(\alpha, \beta)$ *and* $pcon(\beta, \gamma)$*, then* $pcon(\alpha, \gamma)$*.*

Definition 3.4 *(PLCA connected) A PLCA expression* e *is said to be PLCA connected iff* $pcon(\alpha, \beta)$ *holds for any pair of components* α *and* β *of* e*.*

Intuitively, PLCA connectedness guarantees that all the components including the $outermost$ are connected. That is, for any pair of components, there is a trail that can go from one component to the other by tracing components.

Let $e = \langle P, L, C, A, outermost \rangle$ be a consistent connected PLCA expression. For any pair of points $p_1, p_2 \in P$ ($p_1 \neq p_2$), let $(\alpha_0 \ldots, \alpha_n)$ be a trail from p_1 to p_2. If $\alpha_1, \ldots, \alpha_{n-1}$ are not points, then this trail is one of the following three types (Figure 7).

[Trail Types between Points]

[TType1] (p_1, l, p_2)

[TType2] $(p_1, l_0, c, l_1, c, l_2, \ldots, l_k, p_2)$

[TType3] $(p_1, l_1, c_0, a, c_1, a, c_2, \ldots, c_k, l_2, p_2)$

where $l, l_0, \ldots, l_k$ are lines, $l_i \neq l_j$ for each i, j ($0 \leq i < j \leq k$), $c, c_0, \ldots, c_k$ are circuits, $c_i \neq c_j$ for each i, j ($0 \leq i < j \leq k$), and a is an area.

Note that an area and a circuit do not appear in [TType1], and an area does not appear in [TType2]. If there exists a pair of points p_1 and p_2 such that the trail from p_1 to p_2 is [TType3], $m(e)$ is disconnected.

3.4 Planar PLCA Expression

Lemma 3.1 *Let* $e = \langle P, L, C, A, outermost \rangle$ *be a consistent connected PLCA expression that satisfies* $|a.circuits| = 1$ *for any area* $a \in A$*, then* $m(e)$ *is a connected graph.*

Proof)

For any pair of circuits c_1 and c_2 ($c_1 \neq c_2$), there is no area a that satisfies $c_1, c_2 \in a.circuits$, since $|a.circuits| = 1$. That is, there is no area a that satisfies $d\text{-}pcon(c_1, a) \wedge d\text{-}pcon(c_2, a)$. Therefore, for any pair of p_1 and p_2 ($p_1 \neq p_2$), trail type from p_1 to p_2 is either [TType1] or [TType2]. $m(e)$ is a connected graph in both cases.

Theorem 3.2 *For a consistent connected PLCA expression* $e = \langle P, L, C, A, outermost \rangle$*,* e *can be realized in a two-dimensional plane iff* $|P| - |L| - |C| + 2|A| = 0$ *holds.*

Proof)

For a consistent connected PLCA expression e, if there exists an area a that satisfies $|a.circuits| \geq 2$, then we transform e by making lines between circuits so that any area contains only one circuit.

For an area a that satisfies $a.circuits = \{c_1, c_2\}$, let p_1 and p_2 be arbitrary points on c_1 and c_2, respectively. That is, for $c_1.lines = \{l_0^*, \ldots, l_n^*\}$, there exists i ($0 \leq i \leq n$) such that $(p_1 \in l_{i-1}.points) \wedge (p_1 \in l_i.points)$, and for $c_2.lines = \{m_0^*, \ldots, m_k^*\}$, there exists j ($0 \leq j \leq k$) such that $(p_2 \in m_{j-1}.points) \wedge (p_2 \in m_j.points)$.

Delete c_1, c_2, a and add $l'_1, l'_2, c'_1, c'_2, a'_1, a'_2$ that satisfy the followings (Figure 8):

$l'_1.points = [p_1, p_2]$
$l'_2.points = [p_2, p_1]$
$c'_1.lines = \{l_0^*, \ldots, l_{i-1}^*, l_1'^{+}, m_j^*, \ldots, m_k^*, m_0^*, \ldots, m_{j-1}^*, l_2'^{+}, l_i^*, \ldots, l_n^*\}$
$c'_2.lines = \{l_1'^{-}, l_2'^{-}\}$
$a'_1.circuits = \{c'_1\}$
$a'_2.circuits = \{c'_2\}$

This operation is performed by an operation of area division by creating two lines between two existing points. The consistency is preserved by this operation (Takahashi & Sumitomo, 2007). The number of lines is increased by 2, the number of areas is increased by 1, and that of the others is not changed in this operation.

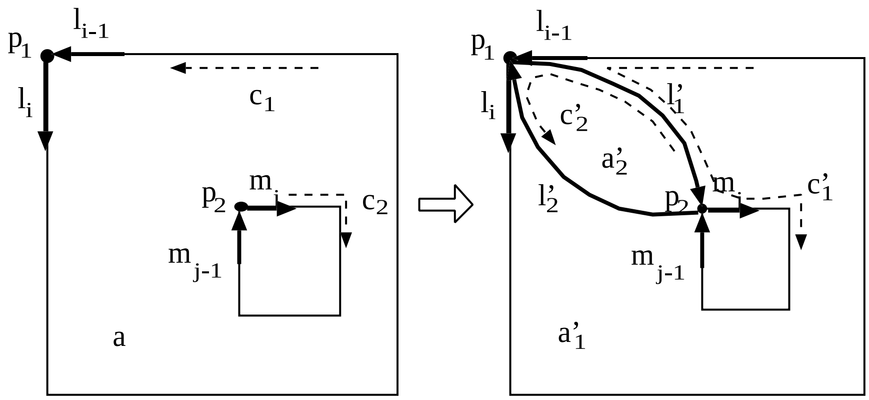


Figure 8.: Area division

For an area such that $|a.circuits| = K \geq 2$, we repeat the operation $K - 1$ times. If we repeat this operation for each area $a \in A$ such that $|a.circuits| \geq 2$, then we can obtain a consistent connected PLCA expression $e' = \langle P', L', C', A', outermost' \rangle$ that satisfies $|a'.circuits| = 1$ for any $a' \in A'$. Let N be the total repeated times of the operation. Then, $|P'| = |P|, |L'| = |L| + 2N, |C'| = |C|$ and $|A'| = |A| + N$.

$m(e')$ is a connected graph by Lemma 3.1. Therefore, $|P'| - |L'| + |A'| = 1$ holds by Theorem 3.1, since $|A'|$ corresponds to the number of faces that does not include the outer infinitely large region. Therefore, $|P| - |L| + |A| = N + 1$ holds.

N is equivalent to $\Sigma_{a \in A}(|a.circuits| - 1) - 1$, since each circuit but for the $outermost$ is contained in only one area. Therefore, $N = |C| - |A| - 1$. Hence, $|P| - |L| - |C| + 2|A| = 0$ holds.

On the other hand, if $|P| - |L| - |C| + 2|A| = 0$ holds, e can be realized in a two-dimensional plane. It can easily be proved by induction on the structure of an expression.

Hereafter, for a PLCA expression $e = \langle P, L, C, A, outermost \rangle$, we denote the value of $|P| - |L| - |C| + 2|A|$ by $\varepsilon(e)$.

Definition 3.5 *(planar PLCA expression) A consistent connected PLCA expression* e *that satisfies* $\varepsilon(e) = 0$ *is said to be planar.*

The PLCA expression in Example 2 is consistent and PLCA connected, but not realizable in a two-dimensional plane (Figure 9). In this case, $\varepsilon(e) = -2$.

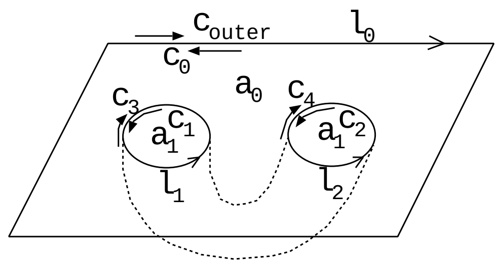


Figure 9.: PLCA expression unrealizable in a two-dimensional plane

$Example$ 2.

$e.points = \{p_0, p_1, p_2\}$
$e.lines = \{l_0, l_1, l_2\}$
$e.circuits = \{c_{outer}, c_0, c_1, c_2, c_3, c_4\}$
$e.areas = \{a_0, a_1\}$
$l_0.points = [p_0, p_0]$
$l_1.points = [p_1, p_1]$
$l_2.points = [p_2, p_2]$
$c_{outer}.lines = [l_0^+]$
$c_0.lines = [l_0^-]$
$c_1.lines = [l_1^+]$
$c_2.lines = [l_2^+]$
$c_3.lines = [l_1^-]$
$c_4.lines = [l_2^-]$
$a_0.circuits = \{c_0, c_3, c_4\}$
$a_1.circuits = \{c_1, c_2\}$

A consistent PLCA expression e that satisfies $\varepsilon(e) = 0$ is not always PLCA connected. The PLCA expression in Example 3 is consistent and $\varepsilon(e) = 0$, but not PLCA connected. It can be divided into a plane part which consists of $p_0, l_0, c_{outer}, c_0, a_0$ and a floating part which consists of the other components. A component of the former is not $pcon$ with that of the latter. (For example, $pcon(c_{outer}, c_1)$ does not hold.) This expression is not realizable in a two-dimensional plane.

Example 3.

$e.points = \{p_0, p_1\}$
$e.lines = \{l_0, l_1, l_2\}$
$e.circuits = \{c_{outer}, c_0, c_1, c_2, c_3\}$
$e.areas = \{a_0, a_1, a_2\}$
$l_0.points = [p_0, p_0]$
$l_1.points = [p_1, p_1]$
$l_2.points = [p_1, p_1]$
$c_{outer}.lines = \{l_0^+\}$
$c_0.lines = \{l_0^-\}$
$c_1.lines = \{l_1^+, l_2^+\}$
$c_2.lines = \{l_2^-\}$
$c_3.lines = \{l_1^-\}$
$a_0.circuits = \{c_0\}$
$a_1.circuits = \{c_1\}$
$a_2.circuits = \{c_2, c_3\}$

3.5 Orientation of a Circuit

Each circuit of a planar PLCA expression e has an orientation of $inner$ or $outer$. If $m(e)$ is a non-connected graph, then it can be decomposed into subgraphs.

We determine the orientation of each circuit using the diagram called "DCO diagram." In this diagram, the nodes correspond to subgraphs, areas and circuits of e, and edges represent their relationships.

[**Algorithm: DCO(determine circuit's orientation)**]

Let e be a consistent connected PLCA expression.

1. Set $c = outermost$.
2. Make a node N_c.
3. Set the orientation of c to be $outer$.
4. For a subgraph g such that $m(c) \in g$, make a node N_g and draw an edge from N_c to N_g.
5. For each $m(c') \in g$ such that $c' \neq c$, do the following:
   (a) Set the orientation of c' to be $inner$.
   (b) Make a node $N_{c'}$ and draw an edge from N_g to $N_{c'}$.
   (c) For an area a such that $c' \in a.circuits$, make a node N_a and draw an edge from $N_{c'}$ to N_a.

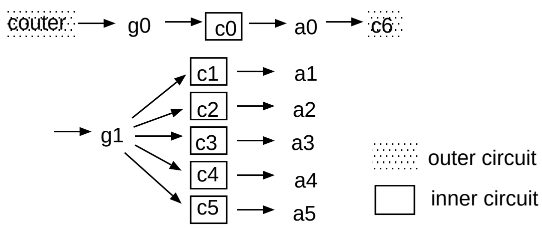


Figure 10.: DCO diagram for Example 1

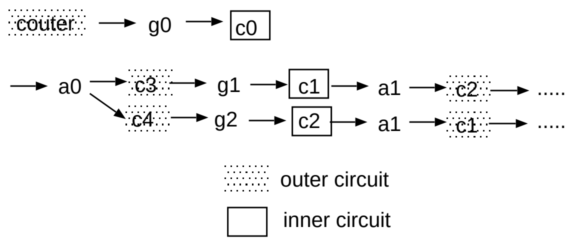


Figure 11.: DCO diagram for Example 2

(d) If there is no circuit $c'' \in a.circuits$ such that $c'' \neq c'$, then terminate. Otherwise, for each c'', do the following:

i. Make a node $N_{c''}$ and draw an edge from N_a to $N_{c''}$.

ii. Set $c = c''$.

iii. Go to 3.

A diagram constructed in this way is called a *DCO diagram*. Each path in the diagram is a sequence of a pattern $N_{c_1} \rightarrow N_g \rightarrow N_{c_2} \rightarrow N_a$ where c_1, c_2 are circuits, a is an area of e, and g is a subgraph of $m(e)$. Figure 10 and Figure 11 show DCO diagrams for Example 1 and Example 2, respectively.

Lemma 3.2 *For a planar PLCA expression e, (i) the orientation of each circuit is decidable, (ii) there exists the unique $inner$ circuit in $a.circuit$ for each area a, and (iii) there exists an $outer$ circuit c such that $m(c) \in g$ is an outer boundary cycle of g for each subgraph g.*

Proof)

Assume that the DCO algorithm does not terminate. There exists c_1, c_2 ($c_1 \neq c_2$) and a such that both patterns $N_{c_1} \rightarrow N_a \rightarrow N_{c_2}$ and $N_{c_2} \rightarrow N_a \rightarrow N_{c_1}$ appear in the DCO diagram. It follows that there are trails $(outermost, \ldots, c_1, a, c_2)$ and $(outermost, \ldots, c_2, a, c_1)$, since e is a planar PLCA expression. This means that c_1 is closer to the $outermost$ than a and also that a is closer to the $outermost$

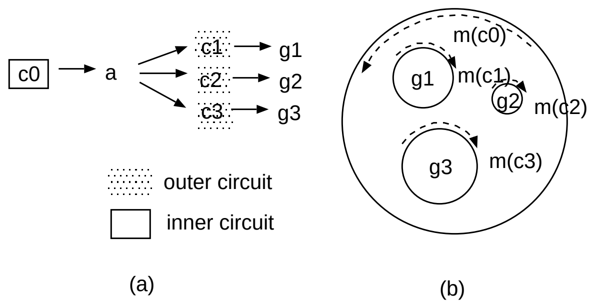


Figure 12.: Realization of a DCO diagram

than c_1, which is a contradiction. Therefore, the DCO algorithm terminates. Thus, each a, c of e and g of $m(e)$ appears at most once in the DCO diagram. On the other hand, each appears at least once from the planarity of the PLCA expression. Therefore, they appear exactly once in the DCO diagram. For each pattern $N_{c_1} \rightarrow N_g \rightarrow N_{c_2} \rightarrow N_a$, c_1 and c_2 are determined as $outer$ and $inner$, respectively. Therefore, the orientation of each circuit is decidable. There exists the unique $inner$ circuit in $a.circuit$ for each area a, since there exists only one N_a such that $N_{c_2} \rightarrow N_a$. $m(c_1)$ in g is an outer boundary cycle of g, since there exists only one N_g such that $N_{c_1} \rightarrow N_g$.

The realization of a part of the DCO diagram shown in Figure 12(a) is shown in Figure 12(b). That is, we can draw subgraphs $g_1, \ldots, g_n$ whose outer boundary cycles are $m(c_1), \ldots, m(c_n)$, respectively, in the area surrounded by $m(c_0)$ so that they do not have a common part.

4 Drawing Algorithm

We describe the algorithm for drawing a figure for a planar PLCA expression in a two-dimensional plane.

4.1 Outline of the Algorithm

First, make a redundant PLCA expression of a given planar PLCA expression using the line division operation, so that the corresponding graph expression of the result is a simple graph.

Second, decompose the graph into connected subgraphs and determine the coordinates for each connected subgraph.

Next, determine the position to be embedded for each connected subgraph by changing the size and/or the position.

Then, substitute a curve for the pair of edges which are connected to the point added in the first step.

The final step, which avoids redundancy, can be omitted.

4.2 Details of the Algorithm

[**Algorithm: DF(drawing a figure)**]

Let $e = \langle P, L, C, A, outermost\rangle$ be a planar PLCA expression.

[STEP1]

[STEP1.1] Elimination of edges that circulate the node itself.

1. Set $Lines = L, NewL = \{\}$.
2. If $Lines = \{\}$, terminate. Otherwise, extract an arbitrary $l \in Lines$ and set $Lines = Lines - \{l\}$.
3. If $l.points = [p, p]$, then divide l to create new lines l_1, l_2 and set $NewL = NewL \cup \{l_1, l_2\}$. At the same time, update the other components of e according to the line division operation. Otherwise, set $NewL = NewL \cup \{l\}$.
4. Go to 2.

[STEP1.2] Elimination of multiple edges that connect the same pair of nodes, so that only one edge remains.

1. Set $L' = \{\}$.
2. If $NewL = \{\}$, terminate. Otherwise, extract an arbitrary $l \in NewL$ and set $NewL = NewL - \{l\}$.
3. If $l.points = [p_1, p_2]$ where $p_1 \neq p_2$ and there exists $l' \in NewL$ such that $l' = [p_1, p_2]$ or $l' = [p_2, p_1]$, do the following:
   (a) Set $P = P \cup \{p'\}$.
   (b) Set $L' = L' \cup \{l_1, l_2\}$ where l_1, l_2 are new lines.
   (c) Set $l_1.points = [p_1, p'], l_2.points = [p', p_2]$.

   Otherwise, set $L' = L' \cup \{l\}$.
4. Go to 2.

As the result, $e' = \langle P', L', C, A, outermost\rangle$ is obtained where P' is the updated set of P by the line division operation.

[STEP2]

We utilize the existing graph algorithms (e.g.(Ochiai, 2004)) in this step.

Let $m(e')$ be the graph corresponding to the PLCA expression obtained in STEP1.

1. Decompose $m(e')$ into the connected subgraphs that have no common elements.

2. For each circuit, determine the orientation using the DCO algorithm provided in the previous section.

3. For each connected subgraph, determine the coordinates that are used to draw a planar graph embedded in the unit circle of the center $(0, 0)$ by a straight line.

[STEP3]

We utilize the existing polygon triangulation algorithm (e.g.(Ima, 2001)) in this step.

For each $a \in A$ where $a.circuits$ contains more than one circuit, do the following:

1. Let $a.circuits = \{c_0, c_1, \ldots, c_n\}$, where the orientation of c_0 is $inner$ and those of $c_1, \ldots, c_n$ are $outer$.

2. Decompose the polygon which has an edge $m(c_0)$ into triangles $t_1, \ldots, t_h$, where the size of t_u is larger than or equal to t_{u+1} ($u = 1, \ldots, h-1$).

3. Let $g_1, \ldots, g_n$ be the subgraphs whose outer boundary cycle are $m(c_1), \ldots, m(c_n)$, respectively.

   (a) Set $k = h$.

   (b) While $k < n$ do the following:

      i. Decompose t_1 into two triangles t_{11} and t_{12}.

      ii. Sort $t_{11}, t_{12}, t_2, \ldots, t_k$ to make a new sequence $t'_1, \ldots, t'_{k+1}$ in the descending order of power.

      iii. Set $k = k + 1$.

   (c) For each i ($i = 1, \ldots, n$), embed g_i in t_i.

[STEP4]

Substitute a curve for the pair of edges and delete the node shared by the pair which corresponds to the points and lines added in STEP1.

4.3 Correctness of the Algorithm

Theorem 4.1 *For a planar PLCA expression, a figure can be drawn in a two-dimensional plane by the algorithm DF.*

Proof)

The line division operation does not affect two-dimensional realizability, since consistency and connectedness are preserved, and the differences in the numbers of points and lines before and after the operation are equivalent. Thus, $m(e')$ which is obtained in STEP1 is a simple planar graph, and the decomposed subgraphs are also simple planar graphs.

A simple planar graph can be drawn in a two-dimensional plane using only straight lines (Chartland & Lesniak, 1996).

From Lemma 3.2, for a planar PLCA expression, the orientation of each circuit can be determined either as $inner$ or $outer$. This lemma also ensures that we can determine the outer boundary cycle for each subgraph and the positional relationships between subgraphs.

In STEP3, for an a such that $a.cycles = \{c_0, c_1, \ldots, c_n\}$, the polygon which has an edge $m(c_0)$ is decomposed into more than $n-1$ triangles and each subgraph is drawn in each triangle. Therefore, these subgraphs are embedded in the inner part of the polygon in such a manner that each pair does not intersect. Therefore, a is realized in this process.

Let $m(p)$ be a point to be deleted in STEP4. We can assume that there are exactly two lines $l_1.points = [p_1, p]$ and $l_2.points = [p, p_2]$, since p is shared only by the lines added in STEP1. Edges $m(l_1)$ and $m(l_2)$ can be replaced by an approximating curve which connects nodes $m(p_1)$ and $m(p_2)$ without changing the connection patterns of areas.

Hence, a figure corresponding to a given PLCA expression can be drawn in a two-dimensional plane.

4.4 Prototype System

We have implemented a prototype system using JAVA. The system checks whether a given PLCA expression is planar, and if it is, the corresponding figure is drawn in a two-dimensional plane (Figure 13) [2].

5 Comparison with other QSR Systems

RCC (Randell et al., 1992) and 9-intersection model (Egenhofer & Franzosa, 1995) are representative frameworks for qualitative spatial reasoning.

[2] In the prototype system, STEP4 has not yet been implemented.

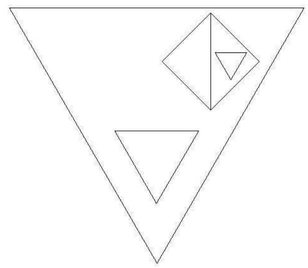


Figure 13.: A figure generated by the prototype system

Existence of a topological space and planarity for a set of RCC relationships are discussed in (Grigni, Papadias, & Papadimitriou, 1995; Renz, 2002; Wolter & Zakharyaschev, 2000). However, an algorithm for drawing has not been given yet.

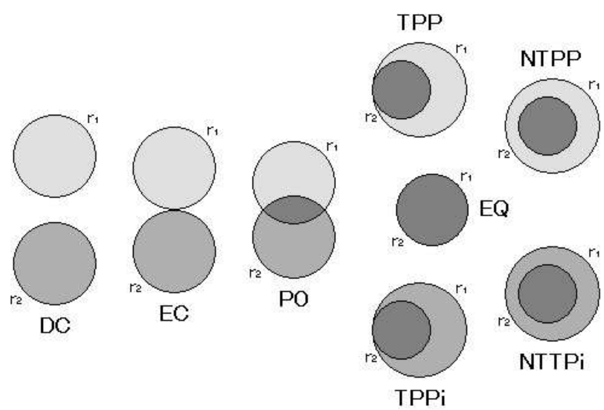


Figure 14.: Fundamental relationships of RCC

Figure 14 shows the fundamental eight relationships of RCC. Consider a set of RCC relationships $S = \{R_1, \ldots, R_n\}$ which is realizable in a two-dimensional plane. If each $R_i (i = 1, \ldots, n)$ is either NTPP, NTPPi, DC or EQ, then S is transformed uniquely into PLCA. In this case, the figure corresponding to S can be drawn using PLCA drawing algorithm. Otherwise, there exists multiple PLCA expressions. We can draw one of them, but it is an open problem whether we can determine which one can be drawn. As for 9-intersection model, in which positional relationships between regions are represented in the form of $3 * 3$ matrix, the similar discussion is available.

A significant difference between PLCA and the other QSR systems is the representation of the relations of the objects. In the other QSR systems, the entire figure is represented in the form of a set of binary relations, while we do not use binary relations. Moreover, objects in the other QSR systems may share their parts with each other which is prohibited in PLCA. PLCA adopts more refined classification on equivalent figures than the other QSR systems. It may be possible to determine the one-to-one mapping from the extension of RCC, which has an information of connection patterns of regions (Cohn & Varzi, 1998), and the extension of 9-intersection model (Egenhofer & Franzosa, 1995; Nedas & Egenhofer, 2004) to PLCA.

6 Concluding Remarks

We have investigated the condition for embedding a PLCA expression in a two-dimensional plane, and shown that we have only to check the numbers of the components. We have proved this condition by utilizing the well-known Euler's formula. We have also provided an algorithm for drawing the figure in a two-dimensional plane for a planar PLCA expression, and proved its correctness. This algorithm generates a quantitative expression from a qualitative expression.

The algorithm utilizes the existing graph decomposition algorithm, graph drawing algorithm, and polygon triangulation algorithm. This suggests that the theorems and the algorithms in graph theory and computational geometry can be useful in studying PLCA.

In the future, we will investigate further the relationship between PLCA and other theories such as graph theory, computational geometry and topology. We will also consider the extension of PLCA to three-dimensional space.

We are now improving the algorithm so that objects are placed at the most visible position and the computational complexity is reduced. For a PLCA expression, we can draw an infinite number of figures. We should select the appropriate one to show depending on the user's purpose and the environment such as a display size. It is also an interesting issue to determine the standard for the most "appropriate" figure representation.

Acknowledgments

The authors would like to thank Shuji Yamada for giving suggestions on planar graph drawing and Shou Kumokawa for implementing a prototype system. This research is supported by KAKENHI17500101.

7 References

Borgo, S., Guarino, N., & Masolo, C. (1996). A pointless theory of space based on strong connection and congruence. *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR96)* (pp. 220–229). Morgan Kaufmann Publishers.

Cohn, A., & Hazarika, S. (2001). Qualitative spatial representation and reasoning: an overview. *Fundamental Informaticae*, *46*(1-1), 1–29.

Chartland, G., & Lesniak, L. (1996). *Graphs & digraphs, 3rd edition*. Wadsworth & Brooks/Cole.

Cohn, A., & Varzi, A. (1998). Connection relations in mereotopology. *Proceedings of the 13th European Conference on Artificial Intelligence (ECAL98)*. (pp. 150–154). Wiley.

de Berg, M., van Kreveld, M., Overmars, M., & Schwarzkopf, O. (1997). *Computational geometry*. Springer-Verlag.

Egenhofer, M., & Franzosa, R. (1995). On the equivalence of topological relations. *International Journal of Geographical Information Systems*, *9*(2), 133–152.

Grigni, M., Papadias., D. & Papadimitriou, C. (1995) Topological Inference. *International Joint Conference on Artificial Intelligence (IJCAI95)*. (pp. 901–907).

Imai Labo, Chuo University (2001). `http://www.ise.chuo-u.ac.jp/ise-labs/imai-lab/program/program.html`

Nedas, K. & Egenhofer, M. (2004). Splitting ratios: Metric details of topological line-line relations. In *Seventeenth International FLorida Conference on Artificial Intelligence (FLAIRS04)*.

Ochiai, N. (2004). *Introduction to graph theory: Application for plane graph*. Nihon-Hyouron-sha (In Japanese).

Randell, D., Cui, Z., & Cohn, A. (1992). A spatial logic based on regions and connection. *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR92)* (pp. 165–176). Morgan Kaufmann Publishers.

Renz, J. (2002). *Qualitative spatial reasoning with topological information*. LNAI-2293. Springer Verlag.

Stock(ed.), O. (Ed.). (1997). *Spatial and temporal reasoning*. Kluwer Academic Press.

Sumitomo, T., & Takahashi, K. (2004). DLCS: Qualitative representation for spatial data. *Twenty-first Annual Meeting of Japan Society for Software Science and Technology (In Japanese)*.

Takahashi, K., & Sumitomo, T. (2007). The qualitative treatment of spatial data. *International Journal on Artificial Intelligent Tools*, 16(4), 661–682.

Takahashi, K., & Sumitomo, T. (2005). A framework for qualitative spatial reasoning based on the connection patterns of regions. *IJCAI-05 Workshop on Spatial and Temporal Reasoning* (pp. 57–62).

Thurston, J., Poiker, T., & Moore, J. P. (Eds.). (2003). *Integrated geospatial technologies*. Wiley.

Wolter, F. & Zakharyaschev, M. (2000). Spatio-temporal representation and reasoning based on RCC-8. *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000)* (pp. 3–14). Morgan Kaufmann Publishers.

Appendix Example of DF Algorithm

Let e be a planar PLCA expression.

$$
\begin{array}{l}
e = \langle P, L, C, A, outermost \rangle \\
e.points = \{p_0, p_1, p_2\} \\
e.lines = \{l_0, l_1, l_2, l_3\} \\
e.circuits = \{c_{outer}, c_0, c_1, c_2, c_3, c_4, c_5\} \\
e.areas = \{a_0, a_1, a_2, a_3\} \\
e.outermost = c_{outer} \\
l_0.points = [p_0, p_0] \\
l_1.points = [p_1, p_1] \\
l_2.points = [p_1, p_1] \\
l_3.points = [p_2, p_2] \\
c_{outer}.lines = \{l_0^+\} \\
c_0.lines = \{l_0^-\} \\
c_1.lines = \{l_1^-\} \\
c_2.lines = \{l_3^+\} \\
c_3.lines = \{l_3^-\} \\
c_4.lines = \{l_2^-\} \\
c_5.lines = \{l_1^+, l_2^+\} \\
a_0.circuits = \{c_0, c_5\} \\
a_1.circuits = \{c_1, c_2\} \\
a_2.circuits = \{c_4\} \\
a_3.circuits = \{c_3\}
\end{array}
$$

[STEP1] Make a simple graph.

e is converted to the following e' so that $m(e')$ is a simple graph.

$$
\begin{aligned}
&e' = \langle P', L', C, A, outermost \rangle \\
&e'.points = \{p_0, p_1, p_2, p_{00}, p_{01}, p_{10}, p_{11}, p_{12}, p_{13}, p_{20}, p_{21}\} \\
&e'.lines = \{l_{00}, l_{000}, l_{001}, l_{10}, l_{100}, l_{101}, l_{20}, l_{200}, l_{201}, l_{30}, l_{300}, l_{301}\} \\
&e'.circuits = \{c_{outer}, c_0, c_1, c_2, c_3, c_4, c_5\} \\
&e'.areas = \{a_0, a_1, a_2, a_3\} \\
&e'.outermost = c_{outer} \\
&l_{000}.points = [p_0, p_{00}] \\
&l_{001}.points = [p_{00}, p_{01}] \\
&l_{00}.points = [p_{01}, p_0] \\
&l_{100}.points = [p_1, p_{10}] \\
&l_{101}.points = [p_{10}, p_{11}] \\
&l_{10}.points = [p_{11}, p_1] \\
&l_{200}.points = [p_1, p_{12}] \\
&l_{201}.points = [p_{12}, p_{13}] \\
&l_{20}.points = [p_{13}, p_1] \\
&l_{300}.points = [p_2, p_{20}] \\
&l_{301}.points = [p_{20}, p_{21}] \\
&l_{30}.points = [p_{21}, p_2] \\
&c_{outer}.lines = \{l^+_{000}, l^+_{001}, l^+_{00}\} \\
&c_0.lines = \{l^-_{00}, l^-_{001}, l^-_{000}\} \\
&c_1.lines = \{l^-_{10}, l^-_{101}, l^-_{100}\} \\
&c_2.lines = \{l^+_{300}, l^+_{301}, l^+_{30}\} \\
&c_3.lines = \{l^-_{30}, l^-_{301}, l^-_{300}\} \\
&c_4.lines = \{l^-_{20}, l^-_{201}, l^-_{200}\} \\
&c_5.lines = \{l^+_{100}, l^+_{101}, l^+_{10}, l^+_{200}, l^+_{201}, l^+_{20}\} \\
&a_0.circuits = \{c_0, c_5\} \\
&a_1.circuits = \{c_1, c_2\} \\
&a_2.circuits = \{c_4\} \\
&a_3.circuits = \{c_3\}
\end{aligned}
$$

[STEP2]

Decompose $m(e')$ into connected subgraphs g_0, g_1 and g_2.

g_0: $\{p_0, p_{00}, p_{01}\}$ $\{l_{000}, l_{001}, l_{00}\}$ $\{c_{outer}, c_0\}$

g_1: $\{p_1, p_{10}, p_{11}, p_{12}, p_{13}\}$ $\{l_{100}, l_{101}, l_{10}, l_{200}, l_{201}, l_{20}\}$ $\{c_1, c_4, c_5\}$

g_2: $\{p_2, p_{20}, p_{21}\}$ $\{l_{300}, l_{301}, l_{30}\}$ $\{c_2, c_3\}$

Determine the coordinates for each subgraph.

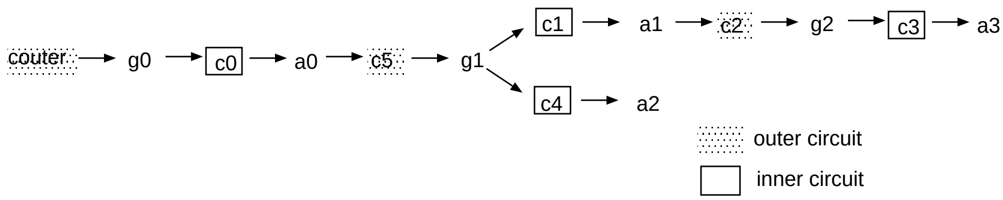


Figure 15.: Example of drawing - DCO diagram

[STEP3]

Determine the orientation of curcuits using the DCO algorithm (Figure 15), and draw subgraphs in the proper part (Figure 16).

[STEP4]

Substitute a curve for a pair of edges.

Finally, we have obtained the figure shown in Figure 17 that corresponds to the PLCA expression e.

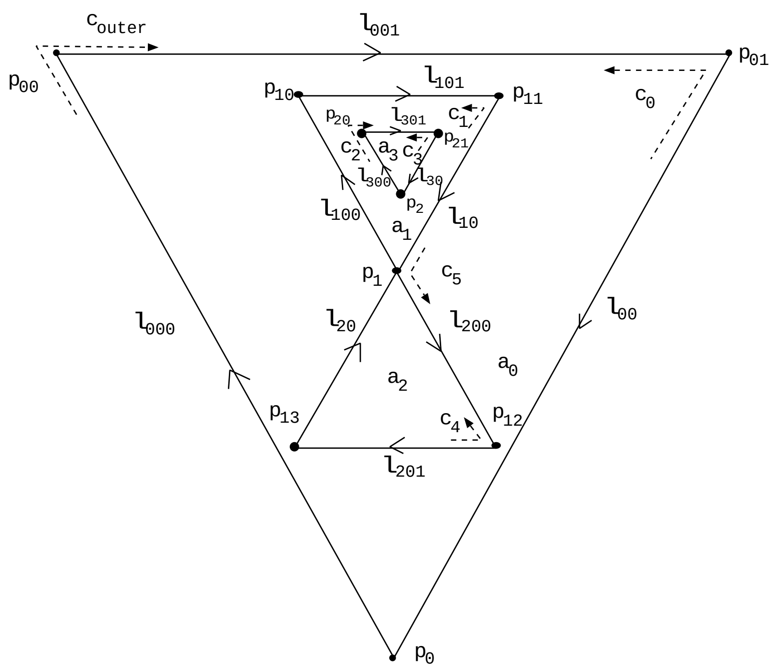


Figure 16.: Example of drawing - location of each subgraph

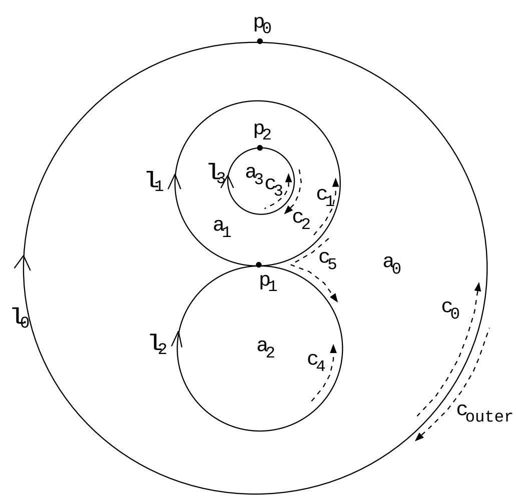


Figure 17.: The obtained figure