

形式言語による結び目の実現問題 に関する幾つかの予想

日本大学大学院総合基礎科学研究科地球情報数理科学専攻
日本大学文理学部情報システム解析学科

上谷雄一
岡田蓉子
堀口 俊
前川達也
鈴木 理

Abstract

ライデマイスター I 型の移動には整合括弧列が対応することに注目してライデマイスター II, III 型の移動に対応する形式言語を構成する。以上で構成された形式言語が実際に結び目から導かれるかどうかの判定に関する予想をのべる (実現問題)。ライデマイスター III 型に対応する文字列の変形を定め、これを用いた自明な結び目への *reduction* を考える。

§ 1 結び目の基礎

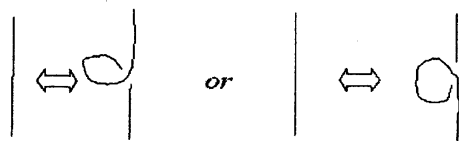
(1) 結び目の定義

3 次元か球面上で自分自身と交わらない閉曲線を用意する。平面を定め、その平面への正射影を結び目のダイアグラムという。最も簡単な結び目は交点のない円周であり、これを自明な結び目のダイアグラムとよぶ。交点を増やさない変形は自由に行ってもよい。

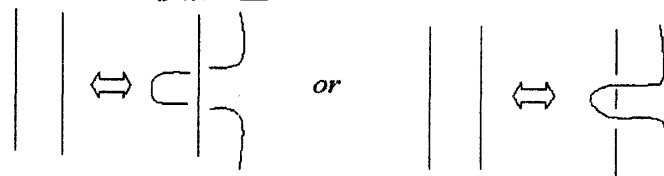
(2) Reidemeister 移動

結び目のダイアグラムの変形を考える。次の 3 種類の変形を考える。

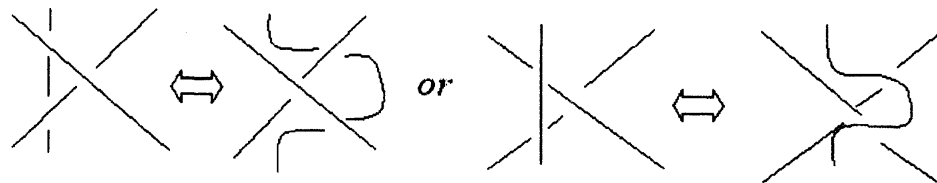
(i) Reidemeister 移動 I 型



(ii) Reidemeister 移動 II 型



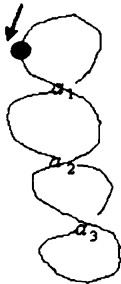
(iii) Reidemeister 移動 III 型



定義([1])

2つの結び目の一方が他方に、Reidemeister 移動の列で移り変わるときに、これらの2つの結び目は同値であるという。自明な結び目と同値になる結び目をほどける結び目という。従ってすべてのほどける結び目は自明な結び目から Reidemeister I, II, III型移動を行って作られる。

(3) 結び目の定める文字列



結び目のダイアグラムをひとつとりこの上に一点をとりこれを始点として結び目のダイアグラムが定める曲線をたどる。交点 $(a_1 a_2 \dots a_i \dots a_n \dots)$ と定める) を通過するたびに上を通れば

a_i^+ , 下を通れば a_i^- と置いて文字列 X を定める。

左図においては、矢印を始点とすると次のようになる；

$$a_1^+ a_2^+ a_3^+ a_3^- a_2^- a_1^-$$

同一の結び目のダイアグラムであっても始点を変えたり、たどる向きをかえると異なる文字列がえられる。本論文ではこれらは異なる列とみなすことにする。また円周となる結び目のダイアグラムに対応する文字列は空語 λ と定める。

次に結び目に関する基本用語をまとめておく：

- (1) 自明な結び目のダイアグラムを λ とあらわす。
- (2) 結び目のダイアグラムが定める文字列は X, Y 等であらわす。文字列 X を二つの文字列 X_1, X_2 に分割するときこれを $X = X_1 X_2$ とかく。
- (3) 文字の $+$ (或いは $-$) の個数を文章 X の長さ (length) といい、 $|X|$ と書く。
長さ n の文字列の全体のつくる集合を L_n とかく。
 $+$ ($-$) から始まる文字列の集合を $L_n^{(+)} (L_n^{(-)})$ とかく。 $L_n = L_n^{(+)} \cup L_n^{(-)}$ となる。

§ 2 形式言語の基礎(整合括弧列)([2])

整合括弧列とは、次の生成規則によって作られる開き括弧と閉じ括弧の列である。たとえば、 $()(), (())$ などは、整合した括弧である。次の仕様は整合括弧列を帰納的に記述している。

- (1) 記号列 $()$ は整合した括弧である。
- (2) 記号列 A が整合した括弧ならば、 (A) も対応した括弧である。
- (3) 記号列 A と B が整合した括弧ならば、 AB も整合した括弧である。

長さ n の括弧列の全体のつくる集合を K_n とかくと次のようになる：

$$K_0^{(+)} = \{\lambda\}, \quad K_1^{(+)} = \{()\}, \quad K_2^{(+)} = \{()(), (())\}, \dots$$

§ 3 R—I型移動およびR—II型移動から動機づけられる形式言語

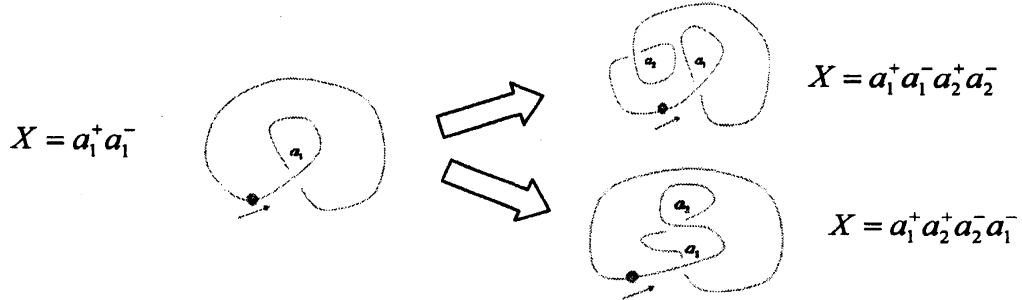
(1) R—I型の形式言語

自明な結び目に R—I 移動繰り返し行なって得られる結び目の文字列 $\hat{X}$ は帰納的に次のように得られる：

$$\hat{X} = \begin{cases} Xa_n^+a_n^+ \\ X_1a_n^+a_n^+X_2 (X = X_1X_2) \end{cases}$$

例

$X = a_1^+a_1^-$ とおくとき, $\hat{X} = a_1^+a_1^-a_2^+a_2^-$, $\hat{X} = a_1^+a_2^+a_2^-a_1^-$ は次のようにして作られる。



このような生成規則で定められた文字列全体の作る形式言語を R-I 型の言語という。この規則を使い L_n を構成する。

$L_n^{(+)} = \{X_n\}$ とするとき $L_{n+1}^{(+)}$ は次の様にして作られる。

$$L_{n+1}^{(+)} = \{a_{n+1}^+X_na_{n+1}^-, a_{n+1}^+a_{n+1}^-X_n : X_n \in L_n\}$$

同様に一から出発する言語の作る集合を $L_n^{(-)}$ とかくと R-I 型の文章が得られる。

$$R^{(1)} = \bigcup_{n=0}^{\infty} R_n^{(1)}, \quad R_n^{(1)} = \left(\bigcup_{i=0}^n L_i^{(+)} \right) \cup \left(\bigcup_{i=0}^n L_i^{(-)} \right)$$

定理 1

$$a_i^+ \Leftrightarrow (\quad \quad \quad a_i^- \Leftrightarrow (\quad \quad \quad)$$

の対応により $L_n^{(+)}$ と K_n とは 1 対 1 に対応する。

(2) R-II 型の形式言語

R-I 型の文章(X とする)から次のように定義される文章を $\hat{X}$ とかき $\hat{X}$ 全体を R-II 型の形式言語という。

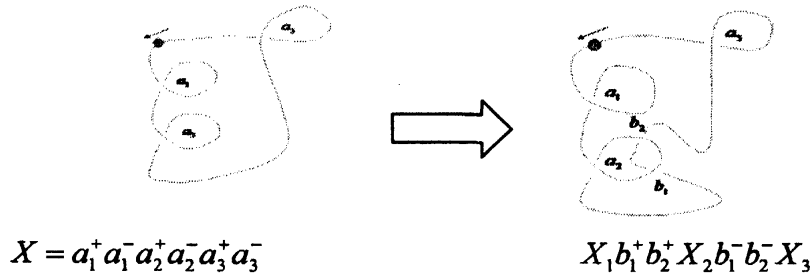
$$\hat{X} = \begin{cases} X_1b_1^+b_2^+X_2b_1^+b_2^+X_3, |X_2| = 2l-1 \\ X_1b_1^+b_2^+X_2b_2^+b_1^+X_3, |X_2| = 2l \end{cases} (X = X_1X_2X_3)$$

ただし複号同順とする。これを偶奇性の規則という。

例

$X = a_1^+a_1^-a_2^+a_2^-a_3^+a_3^-$ と置くとき, $X_1b_1^+b_2^+X_2b_1^+b_2^+X_3$ は次のようにして作られる。

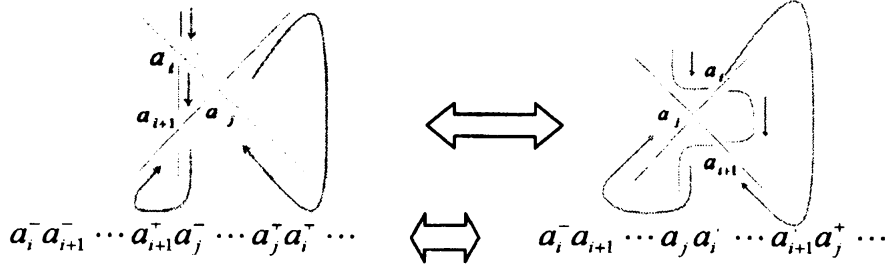
ここでは $X_1 = a_1^+a_1^-a_2^+$, $X_2 = a_2^-$, $X_3 = a_3^+a_3^-$ とする。



§ 4 R-III 型移動の定める形式言語

ここでは R-III 型移動の定める形式言語を構成する。この言語は R-II 言語から構成され R-II 言語には含まれない元が存在することがしめされる。まず、R-III 移動がどのような文章の変形を行うかを述べる。

R-III 型移動は下図のような変形を定める。交点を a_i, a_{i+1}, a_j と置くと



交点の上下と始める位置を変えると文字列の変形 $\delta_i (i=1,2,\dots,16)$ は 16 通りになる。

$$\begin{array}{ll}
 \delta_1 & \dots a_i^\pm a_{i+1}^\pm \dots a_i^\mp a_j^\pm \dots a_{i+1}^\mp a_j^\mp \dots \Leftrightarrow \dots a_i^\pm a_{i+1}^\pm \dots a_j^\mp a_{i+1}^\mp \dots a_j^\mp a_i^\mp \dots \\
 \delta_2 & \dots a_i^\pm a_{i+1}^\pm \dots a_i^\mp a_j^\mp \dots a_{i+1}^\mp a_j^\pm \dots \Leftrightarrow \dots a_i^\pm a_{i+1}^\pm \dots a_j^\mp a_{i+1}^\mp \dots a_j^\pm a_i^\mp \dots \\
 \delta_3 & \dots a_i^\pm a_{i+1}^\pm \dots a_{i+1}^\mp a_j^\pm \dots a_i^\mp a_j^\mp \dots \Leftrightarrow \dots a_i^\pm a_{i+1}^\pm \dots a_j^\mp a_i^\mp \dots a_j^\mp a_{i+1}^\mp \dots \\
 \delta_4 & \dots a_i^\pm a_{i+1}^\pm \dots a_{i+1}^\mp a_j^\mp \dots a_i^\mp a_j^\pm \dots \Leftrightarrow \dots a_i^\pm a_{i+1}^\pm \dots a_j^\mp a_i^\mp \dots a_j^\pm a_{i+1}^\mp \dots \\
 \delta_5 & \dots a_i^\pm a_{i+1}^\pm \dots a_i^\mp a_j^\pm \dots a_j^\mp a_{i+1}^\mp \dots \Leftrightarrow \dots a_i^\pm a_{i+1}^\pm \dots a_j^\mp a_{i+1}^\mp \dots a_i^\mp a_j^\mp \dots \\
 \delta_6 & \dots a_i^\pm a_{i+1}^\pm \dots a_i^\mp a_j^\mp \dots a_j^\pm a_{i+1}^\mp \dots \Leftrightarrow \dots a_i^\pm a_{i+1}^\pm \dots a_j^\mp a_{i+1}^\mp \dots a_i^\mp a_j^\pm \dots \\
 \delta_7 & \dots a_i^\pm a_{i+1}^\pm \dots a_{i+1}^\mp a_j^\pm \dots a_j^\mp a_i^\mp \dots \Leftrightarrow \dots a_i^\pm a_{i+1}^\pm \dots a_j^\mp a_i^\mp \dots a_{i+1}^\mp a_j^\mp \dots \\
 \delta_8 & \dots a_i^\pm a_{i+1}^\pm \dots a_{i+1}^\mp a_j^\mp \dots a_j^\pm a_i^\mp \dots \Leftrightarrow \dots a_i^\pm a_{i+1}^\pm \dots a_j^\mp a_i^\mp \dots a_{i+1}^\mp a_j^\pm \dots \\
 \delta_9 & \dots a_i^\pm a_{i+1}^\mp \dots a_{i+1}^\pm a_j^\pm \dots a_j^\mp a_i^\mp \dots \Leftrightarrow \dots a_i^\mp a_{i+1}^\pm \dots a_j^\pm a_i^\pm \dots a_{i+1}^\mp a_j^\mp \dots \\
 \delta_{10} & \dots a_i^\pm a_{i+1}^\mp \dots a_j^\mp a_{i+1}^\pm \dots a_j^\mp a_i^\mp \dots \Leftrightarrow \dots a_i^\mp a_{i+1}^\pm \dots a_i^\pm a_j^\pm \dots a_{i+1}^\mp a_j^\mp \dots \\
 \delta_{11} & \dots a_i^\pm a_{i+1}^\mp \dots a_j^\mp a_i^\mp \dots a_{i+1}^\pm a_j^\pm \dots \Leftrightarrow \dots a_i^\mp a_{i+1}^\pm \dots a_{i+1}^\mp a_j^\mp \dots a_i^\pm a_j^\pm \dots \\
 \delta_{12} & \dots a_i^\pm a_{i+1}^\mp \dots a_i^\mp a_j^\mp \dots a_{i+1}^\pm a_j^\pm \dots \Leftrightarrow \dots a_i^\mp a_{i+1}^\pm \dots a_j^\mp a_{i+1}^\mp \dots a_i^\pm a_j^\pm \dots \\
 \delta_{13} & \dots a_i^\pm a_{i+1}^\mp \dots a_{i+1}^\pm a_j^\pm \dots a_i^\mp a_j^\mp \dots \Leftrightarrow \dots a_i^\mp a_{i+1}^\pm \dots a_j^\pm a_i^\pm \dots a_j^\mp a_{i+1}^\mp \dots \\
 \delta_{14} & \dots a_i^\pm a_{i+1}^\mp \dots a_j^\mp a_{i+1}^\pm \dots a_i^\mp a_j^\mp \dots \Leftrightarrow \dots a_i^\mp a_{i+1}^\pm \dots a_i^\pm a_j^\pm \dots a_j^\mp a_{i+1}^\mp \dots \\
 \delta_{15} & \dots a_i^\pm a_{i+1}^\mp \dots a_j^\mp a_i^\mp \dots a_j^\pm a_{i+1}^\mp \dots \Leftrightarrow \dots a_i^\mp a_{i+1}^\pm \dots a_{i+1}^\mp a_j^\mp \dots a_j^\pm a_i^\pm \dots \\
 \delta_{16} & \dots a_i^\pm a_{i+1}^\mp \dots a_i^\mp a_j^\mp \dots a_j^\pm a_{i+1}^\mp \dots \Leftrightarrow \dots a_i^\mp a_{i+1}^\pm \dots a_j^\mp a_{i+1}^\mp \dots a_j^\pm a_i^\pm \dots
 \end{array}$$

文字列にこの条件を満たすパターンがあらわれるとき、これを R-III configuration という。

つぎに R-III 言語を定義する。R-II 型の文章 (X とする) から次のように定義される文章を $\hat{X}$ とかき $\hat{X}$ 全体を R-III 型の言語と定める。

$$X \in R - \Pi$$

$$X \xrightarrow{\sigma_i} \hat{X} \in R - \text{III}$$

R-III 言語のなかには R-II 言語からは得られない言語が存在する。

例 $a_1^+ a_2^+ a_3^+ a_4^- a_5^- a_1^- a_6^- a_3^+ a_7^+ a_8^- a_5^+ a_2^- a_6^+ a_7^- a_8^+$

§ 4 近接条件とその判定法

文字列が結び目のダイアグラムから定義される文字列となるときの実現可能という。実現可能かどうかの判定法を与える。これは文字列から結び目の接続関係を導き出すアルゴリズムを提供する。

実現可能な文字列 X が与えられたとき、新たに Π 型の生成規則により得られる $\hat{X}$ が結び目のダイアグラムが実現できるかどうかを次ぎの手続きで判定する。

(step1) X の文字の間に番号(①, ②, ...)を置く。

(step2) 2箇所の番号(①と①とする)の一方、①の両側の文字(a^+, b^+ とする)に着目する。 a^+, b^+ の反対符号の文字 a^-, b^- と a^-, b^- の両側の文字を c^+, d^+, e^+, f^+ とし、この3文字の2つのペアの並び $c^+ a^- d^+, e^+ b^- f^+$ のみを用いて、①の近くで結び目のダイアグラムが実現できるかどうか判定できる(近接判定法という)。

(step3) (step2)の図から①と①が接続できるかどうか調べる。

(step4) (step2)で予想した図では、接続箇所が定まらないなら、 c^+, d^+, e^+, f^+ の反対符号の文字 c^-, d^-, e^-, f^- と c^-, d^-, e^-, f^- のそれぞれの両側の文字から図を予想する。

(step5) (step4)を接続箇所が定まるまで繰り返す。

このように判定法が実行可能であるとき結び目は近接条件を満たすという。

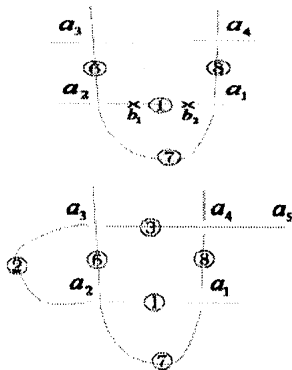
Example(近接判定法の実行例)

$$X = a_1^+ a_2^- a_3^- a_4^+ a_5^+ a_3^+ a_2^- a_1^- a_4^- a_5^-$$

(Step1) この文字列の文字の間に番号を置く。

$$a_1^+ \textcircled{1} a_2^- \textcircled{2} a_3^- \textcircled{3} a_4^+ \textcircled{4} a_5^+ \textcircled{5} a_3^+ \textcircled{6} a_2^- \textcircled{7} a_1^- \textcircled{8} a_4^- \textcircled{9} a_5^- \textcircled{10}$$

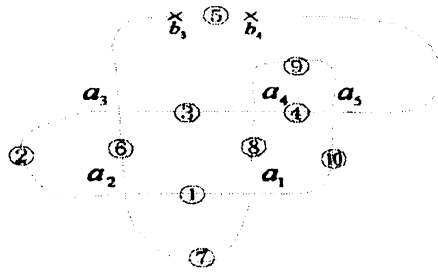
(step2) $b_1^+ b_2^+$ を①に挿入するとき、①の両側の文字 a_1^+, a_2^- の反対符号の文字とその両側に着目する。 $a_2^+ a_1^- a_4^-$ $a_3^+ a_2^- a_1^-$ り①の近くの結び目のダイアグラムが理解できる。



(step3) ①と接続できるのは⑥、⑦、⑧であることが判定できる。①が⑥、⑦、⑧のいずれかであるとき、終了する。

(step4) ①がこれ以外のときには、 a_4^-, a_3^+ の反対符号の文字の両側に着目する。この場合は、 $a_3^- a_4^+ a_5^-, a_2^- a_3^- a_4^+$ に注目する。 $b_1^+ b_2^+$ を①に文字列として挿入したとき、ダイアグラムとして成立するのは新たに③に接続可能であることがわかる。②、④、⑤、⑨、⑩に挿入すると結び目のダイアグラムが成り立たないこともわかる。

(Step5) 以下同様の step を繰り返す。



ここでは⑤の近接条件を調べる。 $b_3^+ b_4^+$ を⑤に挿入するとき、⑤の両側の文字 a_3^+ , a_4^+ の反対符号の文字とその両側に着目する。 $a_4^- a_5^- a_1^+$, $a_2^- a_3^- a_4^+$ に注目する。

以上により①と⑤の近接条件を調べることで左図が予想される。全ての番号の接続箇所が定まった。

§ 6 R-I, R-II型の形式言語の実現問題

与えた文字列の実現可能かどうかを考える。

(i) 予想1：偶奇性の実現条件（必要条件）

任意の a_i に対して、 $X = X_1 a_i^+ X_2 a_i^- X_3$ とおくとき

X が実現可能であるための必要条件は $|X_2| = 2l$ であることである。

(ii) 予想2：近接条件（十分条件）

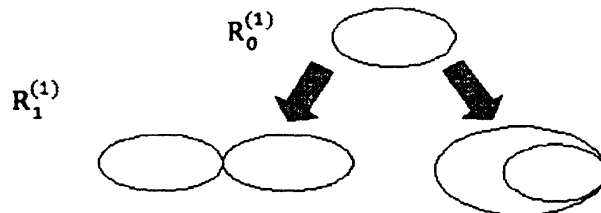
文字列を空語から生成するとき、各生成ステップで近接条件をみたす。

(i) R-I型の形式言語の実現問題

定理2

R-I型の文章はR-I型移動でえられる結び目を実現可能である。

(証明)R-Iの文字列に結び目を対応させる対応は生成規則に下図のように結び目を対応させることによりなされる：



(ii) R-II型の実現問題

実際に実現可能な文字列と相でない文字列の例を述べる：

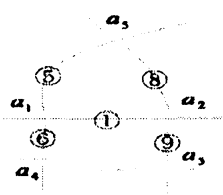
例1 (実現不可能な例)

$$a_1^+ b_1^+ b_2^+ b_3^+ b_4^+ a_1^- b_3^- b_4^- b_1^- b_2^-$$

(Step1) この文字列の文字の間に番号を置く。

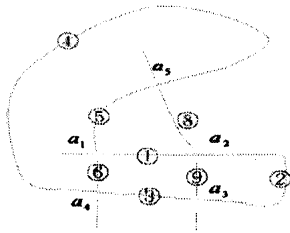
$$a_1^+ ① a_2^+ ② a_3^+ ③ a_4^+ ④ a_5^+ ⑤ a_1^- ⑥ a_4^- ⑦ a_3^- ⑧ a_2^- ⑨ a_1^- ⑩$$

(Step2) $b_1^+ b_2^+$ を①に挿入するとき、①の両側の文字 a_1^+ , a_2^+ の反対符号の文字とその両側に着目する。 $a_5^+ a_1^- a_4^-$, $a_5^- a_2^- a_3^-$ より①の近くの結び目のダイアグラムが理解できる。



(step3)①と接続できるのは①、⑤、⑥、⑧、⑨であることが判定できる。

(step4) a_4^- , a_3^- の反対符号の文字の両側に着目する。この場合は $a_3^+ a_4^+ a_5^+$, $a_2^+ a_3^+ a_4^+$ に注目する。



$b_1^+ b_2^+$ を①に文字列として挿入したとき、新たに③に接続可能であることがわかる。②、④、⑦、⑨、⑩に挿入すると結び目のダイアグラムが成り立たないこともわかる。

a_5^- の両側に着目すると、 $a_4^- a_5^- a_2^-$ である。交点を増やさずに a_4^- の次に a_5^- を通らなければならないが、④をどうしてもまたがなければいけない。よって、与えられた文字列は実際に結び目のダイアグラムを実現できない。

例 2

$a_1^+ a_2^- a_3^- a_7^- a_6^- a_4^+ a_5^+ a_6^+ a_7^+ a_3^+ a_2^+ a_1^- a_4^- a_5^-$

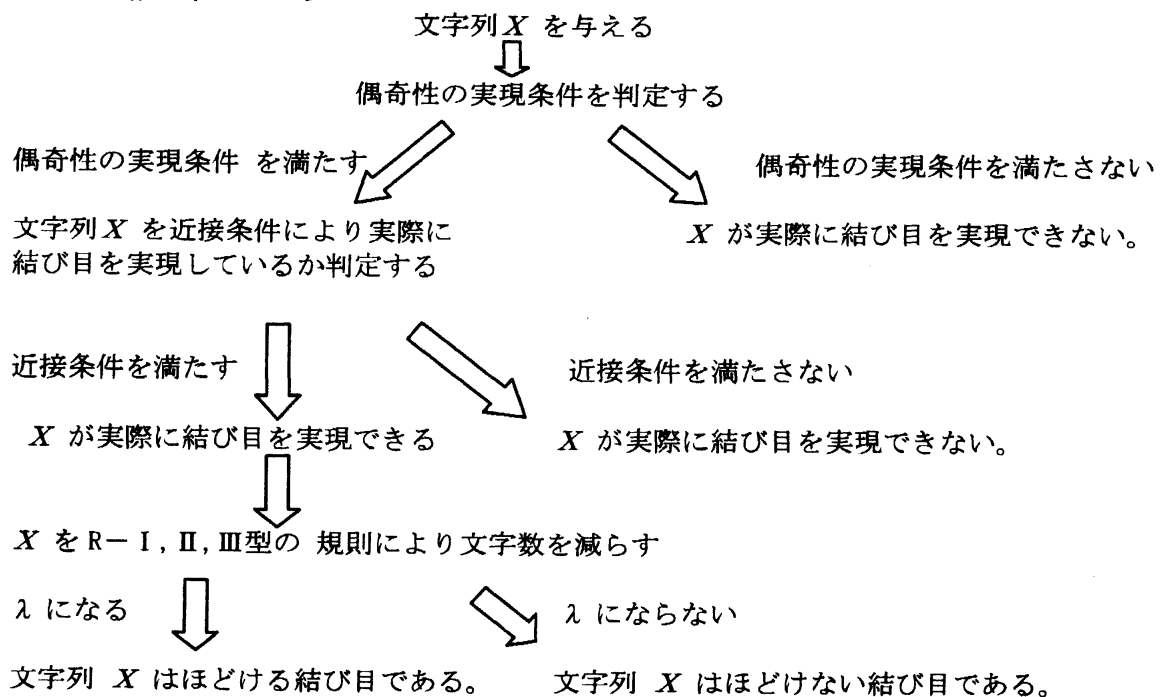
$a_1^+ a_2^- a_3^- a_4^+ a_5^+ a_6^+ a_7^+ a_3^+ a_2^+ a_1^- a_4^- a_5^- a_6^- a_7^-$

例 2 の文章は結び目のダイアグラムを実現できる。なぜなら § 4 の **Example** の結び目のダイアグラムに $a_6^+ a_7^-$ を⑤に文字列として挿入し、③に $a_7^- a_6^-$ を、⑩に $a_6^- b_7^-$ を挿入した文字列であるからである。

§ 7 Reduction 問題

与えられた文字列から文字数を減らすための条件を求める。

文字数を減らすアルゴリズム



例

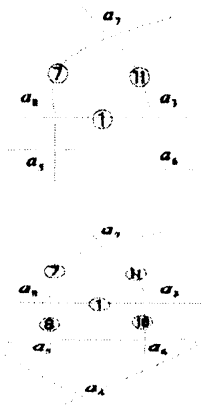
$a_1^+ a_2^+ a_3^+ a_4^- a_5^- a_1^- a_6^- a_3^- a_4^+ a_7^+ a_8^+ a_5^+ a_2^- a_6^+ a_7^- a_8^+$ の reduction を与える。。

(i) この文字列は偶奇性を満たすことがわかる。

(ii) 近接条件を確かめる。右の図は結び目のダイアグラムからライデマイスター I II III 型移動を行って自明な結び目のダイアグラムに変形したものである。

$$\begin{aligned}
 & a_1^+ a_2^+ a_3^+ a_4^- a_5^- a_1^- a_6^- a_3^- a_4^+ a_7^- a_8^+ a_5^- a_2^+ a_6^+ a_7^- a_8^+ \\
 & \quad \downarrow \sigma_6 \\
 & a_1^+ a_2^+ a_3^+ a_4^- a_5^- a_1^- a_2^- a_6^- a_4^+ a_7^- a_8^+ a_5^- a_6^+ a_3^- a_7^- a_8^+ \\
 & \quad \downarrow R-II(a_1 a_2 \text{ を除く}) \\
 & a_3^+ a_4^- a_5^- a_6^- a_4^+ a_7^- a_8^+ a_5^+ a_6^+ a_3^- a_7^- a_8^+ \\
 & \textcircled{1} a_3^+ \textcircled{2} a_4^- \textcircled{3} a_5^- \textcircled{4} a_6^- \textcircled{5} a_4^+ \textcircled{6} a_7^- \textcircled{7} a_8^+ \textcircled{8} a_5^+ \textcircled{9} a_6^+ \textcircled{10} a_3^- \textcircled{11} a_7^- \textcircled{12} a_8^+
 \end{aligned}$$

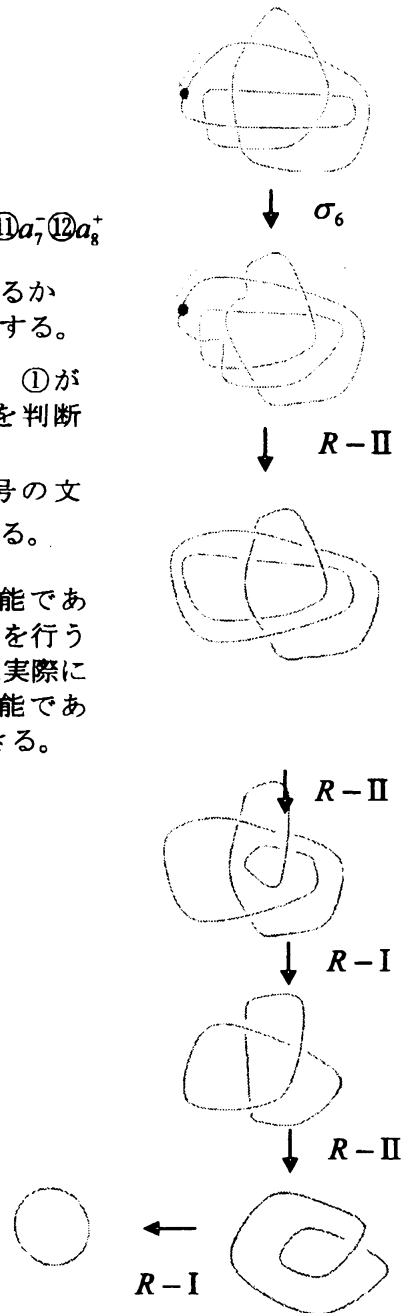
近接条件により元の文が結び目を実現しているか判定する。①の反対符号の文字の両側に着目する。



この図だけでは、①が接続できる箇所を判断できないので a_5^+, a_6^+ の反対符号の文字の両側に着目する。

①と④は接続可能である。以下 Step5 を行うと、この文字列は実際に結び目を実現可能であることが判定できる。

$$\begin{aligned}
 & a_3^+ a_4^- a_5^- a_6^- a_4^+ a_7^- a_8^+ a_5^+ a_6^+ a_3^- a_7^- a_8^+ \\
 & \quad \downarrow R-II(a_5 a_6 \text{ を除く}) \\
 & a_3^+ a_4^- a_4^+ a_7^- a_8^+ a_3^- a_7^- a_8^+ \\
 & \quad \downarrow R-I(a_4 \text{ を除く}) \\
 & a_3^+ a_7^- a_8^+ a_3^- a_7^- a_8^+ \\
 & \quad \downarrow R-II(a_3 a_7 \text{ を除く}) \\
 & a_8^- a_8^+ \\
 & \quad \downarrow R-I(a_8 \text{ を除く}) \\
 & \lambda
 \end{aligned}$$



参考文献

- [1] 結び目の数学, C.C. アダムス, 培風館
- [2] オートマトン・言語理論の基礎, 米田政明, 近代科学社