Coloring for Pattern Detection

Yamada, Yasuhiro
Research Institute for Information Technology, Kyushu University

Hirokawa, Sachio
Research Institute for Information Technology, Kyushu University

出版情報：DOI Technical Report. 233, 2008-03. Department of Informatics, Kyushu University
バージョン：
権利関係：

Coloring for Pattern Detection

Yasuhiro Yamada and Sachio Hirokawa

Research Institute for Information Technology, Kyushu University,
Fukuoka 812-8581, Japan
{yamada,hirokawa}@cc.kyushu-u.ac.jp

Abstract. This paper studies pattern detection of Web documents with the same type of contents using pattern languages. The pattern detection problem of the documents is to find a descriptive regular pattern of input strings such that successive variables do not appear in the pattern. Our pattern detection approach is to find a set of substrings of input strings instead of to detect a pattern of the strings directly. The set is called a component set. It divides each input string into colored regions and non-colored regions. This paper proposes an algorithm to generate a pattern from the two regions. Under this approach, the pattern detection problem is replaced as the problem that is to find a component set.

1 Introduction

The number of Web documents has been increased rapidly. A large amount of information is published in the form of documents in which the structure of the information is not explicit, as in the case of a database. Lack of explicit structure is a cause of difficulty in using the documents as a database.

In a Web site, documents with the same type of contents look the same although the contents are different. We say that a format of the documents is the same. Examples of such documents are news articles and search result pages. The contents of news articles are the headline, the date and the body text. In most cases, HTML or XML tags are used to describe a format. In some cases, however, simple strings and special characters are also used to describe it.

If only the contents are extracted from the documents, they can be treated as a database system. In order to achieve that, we need to detect a format of Web documents. The format detection also applies to information retrieval [3], discovering important content blocks [11], eliminating noisy information from Web pages [8, 14], Web page clustering and classification [14], and information extraction [2, 4–7, 10, 13].

We proposed format detection algorithms in [8, 9] and confirmed the effectiveness of the algorithms using actual Web documents. We utilized the algorithms to make wrappers for information extraction automatically [13]. However, we did not discuss about a formulation of a format of Web documents.

We use pattern languages [1] to define a format of Web documents with the same type of contents. A pattern is the concatenation of constant symbols and variable symbols. A pattern is regular if it has at most one occurrence for each

variable [12]. Web documents with the same type of contents consist of their contents and the common strings for describing them. Therefore, the pattern detection problem of Web documents is to find a descriptive regular pattern, such that successive variables do not appear in the pattern, of a set of strings.

In order to detect a pattern of Web documents, our proposed approach [8, 9] divides each input string into colored regions and non-colored regions. The colored regions correspond to constants of a pattern and the non-colored regions correspond to variables of the pattern. This paper proposes an algorithm to generate a pattern from the two regions. Consequently, the pattern detection problem is replaced as the coloring problem which is to obtain the two regions from a set of strings.

In order to solve the coloring problem, our approach finds a set of substrings of input strings. We call the set a component set. Intuitively speaking, all the occurrences of the strings in the set color each input string. Under the approach, the coloring problem is replaced as the component set discovery problem which is to find a component set of input strings.

2 Preliminaries

The set Σ is a finite alphabet. When $a_i \in \Sigma$ for all i $(1 \leq i \leq n)$, $x = a_1 \cdots a_n$ is a *string* over Σ. The set of all finite strings of symbols from Σ is denoted by Σ^*. We denote the empty string by ε. The set of all finite non-null strings of symbols from Σ is denoted by Σ^+. A *sample* is a finite non-empty subset of Σ^+. The *length* of x is denoted by $|x|$. For an integer $1 \leq i \leq |x|$, the i-th character of x is denoted by $x[i]$. Let x and y be two strings. The concatenation of x and y is denoted by $x \cdot y$ or simply by xy. If $|x| = |y|$ and $x[i] = y[i]$ for each $1 \leq i \leq |x|$, the relationship between x and y is denoted by $x = y$.

For a string x, if there exist strings $u, v, w \in \Sigma^*$ such that $x = uvw$, u is called a *prefix* of x, v is a *substring*, and w is a *suffix*. A substring from the i-th character to the j-th one on x is denoted by $x[i : j]$. If there exists a positive integer i such that $x[i : i + |v| - 1] = v$, we say that v appears at the position i on x.

2.1 Pattern Language

This section introduces pattern languages according to [1]. Let V be an infinite set of symbols disjoint from Σ. An element in Σ is called a *constant* and an element in V is called a *variable*. A *pattern* is any non-empty string over $\Sigma \cup V$. The set of all patterns is denoted by P. The *length* of a pattern p, denoted by $|p|$, is the number of symbols composing it. The concatenation of two patterns p and q is denoted by pq. A pattern is *regular* if it has at most one occurrence for each variable [12]. In the sequel of this paper, we consider only regular patterns.

Let f be a non-erasing homomorphism from P to P. If $f(a) = a$ for any constant a, then f is called a *substitution*. If f is a substitution, $f(x)$ is in V, and $f(x) = f(y)$ implies $x = y$ for any variables x and y, then f is called a

renaming of variables. Let p and q be patterns, then we define a binary relation: (1) $p \equiv q$ iff $p = f(q)$ for some renaming of variables f, (2) $p \leq q$ iff $p = f(q)$ for some substitution f.

Let p be a pattern, then the *language* of p, denoted by $L(p)$, is the set $\{s \in \Sigma^+ \mid s \leq p\}$. If $s \in L(p)$, we say that p generates s.

Let S be a sample and p be a pattern, then p is a pattern of S iff $S \subseteq L(p)$. A pattern p is *descriptive* with respect to S iff (1) $S \subseteq L(p)$ and (2) there is no pattern q such that $S \subseteq L(q) \subset L(p)$.

3 Pattern Detection Problem

We consider information extraction from Web documents with the same type of contents. Fig. 1 shows two HTML documents of personal information with the name, the job title and the e-mail address. Information extraction from the documents is to extract only the three fields from the documents.

Sachio Hirokawa	Yasuhiro Yamada
Professor	Researcher
hirokawa@cc.kyushu-u.ac.jp	yamada@gmail.com

Fig. 1. HTML documents of personal information

The instances of each field are different between the documents. On the other hand, the other strings are the same between the documents. Therefore, we can consider that the documents are generated by a pattern. The name, the job title and the e-mail address correspond to variables and the other strings correspond to constants. The pattern is "
x_1
\n
x_2 \n
x_3
" where x_1, x_2 and x_3 are variables and \n stands for a new line.

As this example, a field is surrounded by two strings which describe the structure of it or the design of it on Web browsers. The two strings are common to Web documents with the same type of contents. We can consider that a variable is surrounded by constants in a pattern of the documents. A pattern of the documents is expressed by a regular pattern such that successive variables do not appear in the pattern.

The learning problem of the pattern language is to find a descriptive pattern of a given sample [1]. Therefore, the pattern detection problem for Web documents with the same type of contents is defined as follows.

Definition 1 (Pattern Detection Problem). *The pattern detection problem is to find a descriptive regular pattern, such that successive variables do not appear in the pattern, of a given sample.*

4 Coloring Algorithm I

In order to solve the pattern detection problem in the previous section, our approach divides each string in a given sample into two regions which are colored and non-colored regions. This section proposes an algorithm to generate a pattern of a sample from the regions.

First, a *coloring* of a string is defined as follows.

Definition 2. *Let s be a non-null string. A pair (s, c) of two strings is a coloring of s iff $c \in \{0,1\}^+$, $|s| = |c|$.*

For instance, let $s = abcbcac$ be a string, then a pair $(abcbcac, 0101010)$ is a coloring of s. Other coloring of s is $(abcbcac, 0111110)$.

An order relation of colorings of a string is defined as follows.

Definition 3. *Let (s, c) and (s, c') be colorings of a string s. Then, $(s, c) \leq (s, c')$ iff $c[i] \leq c'[i]$ for all i.*

In the above example, $(abcbcac, 0101010) \leq (abcbcac, 0111110)$.

The positions of 0 on c are defined as follows.

Definition 4. *Let (s, c) be a coloring of a string s. The l-th position of 0 on c, denoted by $z(c, l)$, is $min\{1 \leq k \leq |c| \mid \#\{1 \leq k' \leq k \mid c[k'] = 0\} = l\}$.*

For instance, let $(s, c) = (abcbcac, 0111010)$ be a coloring of a string s, then $z(c, 1) = 1$, $z(c, 2) = 5$ and $z(c, 3) = 7$.

Definition 5. *Let (s, c) be a coloring of a string s. The number of 0 on c, denoted by $sum_0(c)$, is $\#\{1 \leq k \leq |c| \mid c[k] = 0\}$. The number of successive 1's regions on c, denoted by $rsum_1(c)$, is $\#\{1 \leq k \leq |c'|-1 \mid c'[k] = 1 \wedge c'[k+1] = 0\}$ where $c' = c0$.*

For instance, let $(s, c) = (abcbcac, 0111010)$ be a coloring of a string s, then $sum_0(c) = 3$ and $rsum_1(c) = 2$.

The above definitions are with respect to a coloring of a single string. We extend them to a coloring of a set of strings.

Definition 6. *Let $S = \{s_1, \ldots, s_n\}$ be a sample. A set $C = \{(s_1, c_1), \ldots, (s_n, c_n)\}$ of pairs of two strings is a coloring of S iff $(s_i, c_i) \in C$ is a coloring of s_i for all i.*

For instance, let $S = \{abc, aabba\}$ be a sample, then $\{(abc, 011), (aabba, 00101)\}$ is a coloring of S.

Definition 7. *Let $S = \{s_1, \ldots, s_n\}$ be a sample and $C = \{(s_1, c_1), \ldots, (s_n, c_n)\}$ be a coloring of S. The maximum of $sum_0(c_i)$ for all i, denoted by $dzero(C)$, is $max\{sum_0(c_i) \mid 1 \leq i \leq n\}$. The maximum of $rsum_1(c_i)$ for all i, denoted by $done(C)$, is $max\{rsum_1(c_i) \mid 1 \leq i \leq n\}$.*

For instance, in the above example, $dzero(C) = 3$ and $done(C) = 2$.

The order of two colorings of a sample is defined as follows.

Definition 8. *Let $S = \{s_1, \ldots, s_n\}$ be a sample, $C = \{(s_1, c_1), \ldots, (s_n, c_n)\}$ and $C' = \{(s_1, c'_1), \ldots, (s_n, c'_n)\}$ be colorings of S. Then, $C \leq C'$ iff $(s_i, c_i) \leq (s_i, c'_i)$ for all i.*

Let $S = \{bcbcac, abbcba\}$ be a sample, $C = \{(bcbcac, 110000), (abbcba, 010100)\}$ and $C' = \{(bcbcac, 110110), (abbcba, 111110)\}$ be colorings of S. Then, $C \leq C'$.

A *common coloring* of a sample is defined as follows.

Definition 9. *Let $S = \{s_1, \ldots, s_n\}$ be a sample and $C = \{(s_1, c_1), \ldots, (s_n, c_n)\}$ be a coloring of S. Then, C is a common coloring of S iff (1) $\forall i, j$ $sum_0(c_i) = sum_0(c_j)$, (2) $\forall i, j, l$ $s_i[z(c_i, l)] = s_j[z(c_j, l)]$ $(1 \leq l \leq sum_0(c_1))$, (3) $\forall i, j, l$ $z(c_i, l+1) = z(c_i, l) + 1 \Leftrightarrow z(c_j, l+1) = z(c_j, l) + 1$ $(1 \leq l \leq sum_0(c_1))$, (4) $\forall i, j$ $c_i[1] = c_j[1]$ and $c_i[|c_i|] = c_j[|c_j|]$.*

The first condition indicates that the number of 0 on c_i for all i is the same. The second condition indicates that the character of s_i at the l-th position of 0 on c_i for all i is the same. The third condition indicates that the l-th and $(l+1)$-th positions of 0 on c_i for all i are successive if there exists an integer j such that those on c_j are successive. The last condition indicates that the first and the last characters on c_i for all i are the same. For instance, let $S = \{babcb, baacbb\}$ be a sample, $C = \{(babcb, 00110), (baacbb, 001110)\}$ and $C' = \{(babcb, 10101), (baacbb, 010101)\}$ be colorings of S. Then, C is a common coloring of S, but C' is not.

Next, we propose an algorithm C2P which outputs a pattern from a coloring $C = \{(s_1, c_1), \ldots, (s_n, c_n)\}$ of a sample $S = \{s_1, \ldots, s_n\}$. Fig. 2 is the pseudo code of the algorithm. The algorithm calls the function make_pattern(), and the function calls itself recursively. The algorithm receives a pattern generated by the function. If the last character of the pattern is the special character "\$" which is not in Σ, the algorithm outputs failure. Otherwise, it outputs the generated pattern.

The function make_pattern() generates a pattern by comparing the first characters of (s_i, c_i) for all i. If (s_i, c_i) for all i is $(null, null)$ (i.e. headnull(C) is true), the function finishes calling itself and returns the empty string.

If $c_i[1] = 1$ for all i (i.e. headone(C) is true), the pattern is the concatenation of a new variable x and make_pattern(shiftone(C)). Note that the variable x introduced in the case headone(C) is a new variable. The function shiftone(C) deletes the first 1's region on c_i for all i. Let $shiftone(C) = \{(s'_1, c'_1), \ldots, (s'_n, c'_n)\}$. If the character 0 does not appear on c_i for some i, then $(s'_i, c'_i) = (null, null)$. Otherwise, $(s'_i, c'_i) = (s_i[k : |s_i|], c_i[k : |c_i|])$ where $k = min\{1 \leq l \leq |c_i| \mid c[l] = 0\}$.

If $c_i[1] = 0$ for all i (i.e. headzero(C) is true) and $s_i[1]$ for all i is the same (i.e. samehead(C) is true), the pattern is the concatenation of $s_1[1]$ as a constant (head(C)) and make_pattern(shiftzero(C)). The function shiftzero(C) deletes the first characters on s_i and c_i for all i. Let $shiftzero(C) = \{(s'_1, c'_1), \ldots, (s'_n, c'_n)\}$. If $|s_i| = 1$ for some i, then $(s'_i, c'_i) = (null, null)$. Otherwise, $(s'_i, c'_i) = (s_i[2 : |s_i|], c_i[2 : |c_i|])$.

The function make_pattern() returns the special character "\$" which is not in Σ if (1) there exist integers i, j such that $c_i[1] \neq c_j[1]$, or (2) $c_i[1] = 0$ for all i

```
Algorithm C2P
  Input: C = {(s_i, c_i) | 1 ≤ i ≤ n, s_i ∈ Σ^+, c_i ∈ {0,1}^+, |s_i| = |c_i|};
  Output: pattern or failure;
  var p : pattern;
  begin
    p = make_pattern(C);
    if (the last character of p is $){
      output failure;
    } else {
      output p;
    }
  end

function make_pattern
  Input: C = {(s_i, c_i) | 1 ≤ i ≤ n, s_i ∈ Σ^*, c_i ∈ {0,1}^*, |s_i| = |c_i|};
  Output: pattern;
  begin
    if headnull(C){
      return ε;
    } elsif headone(C){
      return x · make_pattern(shiftone(C));
    } elsif (headzero(C)&&samehead(C)) {
      return head(C) · make_pattern(shiftzero(C));
    } else {
      return $;
    }
  end
```

Fig. 2. An algorithm to generate a pattern from a coloring of a sample

but there exist integers i, j such that $s_i[1] \neq s_j[1]$, or (3) there exists an integer i such that (s_i, c_i) is (null, null) but there exists an integer j such that (s_j, c_j) is not (null, null).

Theorem 1. *Let $S = \{s_1, \ldots, s_n\}$ be a sample, and $C = \{(s_1, c_1), \ldots, (s_n, c_n)\}$ be a coloring of S. If C2P(C) outputs a pattern p, then p is a pattern of S.*

Proof. We prove the theorem by double induction on $(dzero(C), done(C))$.

(1) Base Step: $dzero(C) = 0$. Since $sum_0(c_i) = 0$ for all i, $c_i \in \{1\}^+$. Under the situation, headone(C) is true and make_pattern(shiftone(C)) returns the empty string. Therefore, the output pattern p that C2P(C) returns is a pattern $p = x$ where x is a variable. Then, p is a pattern of S.

(2) Induction Step: Since $dzero(C) > 0$, there is some (s_i, c_i) such that s_i and c_i are not null. Since C2P(C) outputs a pattern p, we have (2-1) headone(C) or (2-2) headzero(C) & samehead(C).

(2-1) headone(C). We define the prefixes h_i of s_i (i.e. $s_i = h_i s'_i$) and d_i of c_i (i.e. $c_i = d_i c'_i$) $(1 \leq i \leq n)$, and consider $C' = \{(s'_1, c'_1), \ldots, (s'_n, c'_n)\}$ such that

$h_i = s_i[1:k_i]$ and $d_i = c_i[1:k_i]$ where $k_i = max\{1 \leq y \leq |c_i| \mid \forall x\ 1 \leq x \leq y, c_i[x] = 1\}$. Since $dzero(C) > 0$, we have (2-1-1) $\exists i\ |s'_i| = 0$ or (2-1-2) $\forall i\ |s'_i| \neq 0$.

(2-1-1) $\exists i\ |s'_i| = 0$. Then, make_pattern(C') returns "\$" and make_pattern(C) returns "x\$" where x is a variable. Therefore, C2P(C) returns failure. This contradicts the assumption.

(2-1-2) $\forall i\ |s'_i| \neq 0$. We can easily confirm that C' is a coloring of $S' = \{s'_1, \ldots, s'_n\}$ because $\forall i\ c'_i \in \{0,1\}^+$ and $|s'_i| = |c'_i|$. Since $dzero(C') = dzero(C)$ and $done(C') = done(C) - 1$, we can apply induction hypothesis for C' and $q = C2P(C')$ is a pattern of S'. In the situation, we have $p = xq$ where x is a variable because headone(C) is true and $C' = shiftone(C)$. Therefore, p is a pattern of S.

(2-2) headzero(C) & samehead(C). We define the prefixes h_i of s_i (i.e. $s_i = h_i s'_i$) and d_i of c_i (i.e. $c_i = d_i c'_i$) $(1 \leq i \leq n)$, and $C' = \{(s'_1, c'_1), \ldots, (s'_n, c'_n)\}$ such that $h_i = s_i[1]$ and $d_i = c_i[1]$. Since $dzero(C) > 0$, we have (2-2-1) $\forall i\ |s'_i| = 0$ or (2-2-2) $\exists i\ |s'_i| \neq 0$.

(2-2-1) $\forall i\ |s'_i| = 0$. In this situation, $|s_i| = 1$ for all i and $s_1 = \cdots = s_n$. C2P(C) outputs a pattern $p = s_1$. Therefore, p is a pattern of S.

(2-2-2) $\exists i\ |s'_i| \neq 0$. We have (2-2-2-1) $\exists j\ |s'_j| = 0$ or (2-2-2-2) $\forall j\ |s'_j| \neq 0$.

(2-2-2-1) $\exists j\ |s'_j| = 0$. Then, the function make_pattern(C') returns "\$" and make_pattern(C) returns "$s_1[1]$\$". Therefore, C2P(C) returns failure. This contradicts the assumption.

(2-2-2-2) $\forall j\ |s'_j| \neq 0$. We can easily confirm that C' is a coloring of $S' = \{s'_1, \ldots, s'_n\}$ because $\forall i\ c'_i \in \{0,1\}^+$ and $|s'_i| = |c'_i|$. Since $dzero(C') = dzero(C) - 1$ and $done(C') = done(C)$, we can apply induction hypothesis for C' and $q = C2P(C')$ is a pattern of S'. In the situation, we have $p = s_1[1]q$ because headzero(C) & samehead(C) is true and $C' = shiftzero(C)$. Therefore, p is a pattern of S.

We have now fulfilled both conditions of the principle of mathematical induction. □

The time complexity of the algorithm C2P is $O(N)$ where N is the total length of strings in an input sample because the algorithm reads all characters on s_i and c_i for all i at most once.

We give the following lemma about a common coloring of a sample.

Lemma 1. *Let $S = \{s_1, \ldots, s_n\}$ be a sample, and $C = \{(s_1, c_1), \ldots, (s_n, c_n)\}$ be a coloring of S. Then, C is a common coloring of S iff C2P(C) outputs a pattern.*

Proof. First, we prove that C2P(C) outputs a pattern if C is a common coloring of S by double induction on $(dzero(C), done(C))$.

(1) Base Step: $dzero(C) = 0$. Since $sum_0(c_i) = 0$ for all i, $c_i \in \{1\}^+$. Under the situation, the function headone(C) is true and make_pattern(shiftone(C)) returns the empty string. Therefore, C2P(C) outputs a pattern $p = x$ where x is a variable.

(2) Induction Step: Since C is a common coloring of S, we have (2-1) $\forall i\ c_i[1] = 1$ or (2-2) $\forall i\ c_i[1] = 0$.

(2-1) $\forall i\ c_i[1] = 1$. We define the prefixes h_i of s_i (i.e. $s_i = h_i s'_i$) and d_i of c_i (i.e. $c_i = d_i c'_i$) $(1 \leq i \leq n)$, and consider $C' = \{(s'_1, c'_1), \ldots, (s'_n, c'_n)\}$ such that $h_i = s_i[1 : k_i]$ and $d_i = c_i[1 : k_i]$ where $k_i = max\{1 \leq y \leq |c_i| \mid \forall x\ 1 \leq x \leq y, c_i[x] = 1\}$. Since $dzero(C) > 0$, we have (2-1-1) $\exists i\ |s'_i| = 0$ or (2-1-2) $\forall i\ |s'_i| \neq 0$.

(2-1-1) $\exists i\ |s'_i| = 0$. Then, $c_i \in \{1\}^+$ for such i. Since $dzero(C) > 0$, there is some (s'_j, c'_j) such that $c'_j[1] = 0$. Thus, we have $sum_0(c_i) = 0$ and $sum_0(c_j) = sum_0(c'_j) \geq 1$ which contradicts the first condition of common coloring.

(2-1-2) $\forall i\ |s'_i| \neq 0$. We can confirm that C' is a common coloring of $S' = \{s'_1, \ldots, s'_n\}$ because (1) $\forall i\ sum_0(c'_i) = sum_0(c_i)$, therefore $\forall i, j\ sum_0(c'_i) = sum_0(c'_j)$; (2) $\forall i, l\ \ s_i[z(c_i, l)] = s'_i[z(c'_i, l)]$ $(1 \leq l \leq sum_0(c_i))$, thus $\forall i, j, l$ $s'_i[z(c'_i, l)] = s'_j[z(c'_j, l)]$ and (3) $\forall i, j, l\ z(c'_i, l+1) = z(c'_i, l) + 1 \Leftrightarrow z(c'_j, l+1) = z(c'_j, l) + 1$; (4) $\forall i\ c_i[|c_i|] = c'_i[|c'_i|]$, therefore $\forall i, j\ c'_i[|c'_i|] = c'_j[|c'_j|]$ and $c'_i[1] = c'_j[1] = 0$. Since $dzero(C') = dzero(C)$ and $done(C') = done(C) - 1$, we can apply induction hypothesis for C'. C2P(C) outputs a pattern $p = xq$ where x is a variable and $q = C2P(C')$ because headone(C) is true and $C' = shiftone(C)$.

(2-2) $\forall i\ c_i[1] = 0$. Since C is a common coloring of S, we have $\forall i, j\ s_i[1] = s_j[1]$. We define the prefixes h_i of s_i (i.e. $s_i = h_i s'_i$) and d_i of c_i (i.e. $c_i = d_i c'_i$) $(1 \leq i \leq n)$, and consider $C' = \{(s'_1, c'_1), \ldots, (s'_n, c'_n)\}$ such that $h_i = s_i[1]$ and $d_i = c_i[1]$. Since $dzero(C) > 0$, we have (2-2-1) $\forall i\ |s'_i| = 0$ or (2-2-2) $\exists i\ |s'_i| \neq 0$.

(2-2-1) $\forall i\ |s'_i| = 0$. In this situation, $|s_i| = 1$ for all i and $s_1 = \cdots = s_n$. Then, we have headzero(C) & samehead(C). Therefore, C2P(C) outputs $s_1 \cdot make\_pattern(C') = s_1$.

(2-2-2) $\exists i\ |s'_i| \neq 0$. We have (2-2-2-1) $\exists j\ |s'_j| = 0$ or (2-2-2-2) $\forall j\ |s'_j| \neq 0$.

(2-2-2-1) $\exists j\ |s'_j| = 0$. If $c'_i[|c'_i|] = 0$ for such i, $sum_0(c_i) > 1 = sum_0(c_j)$. If $c'_i[|c'_i|] = 1$ for such i, $c_i[|c_i|] \neq c_j[|c_j|] = 0$. Therefore, C is not a common coloring of S. This contradicts the assumption.

(2-2-2-2) $\forall j\ |s'_j| \neq 0$. The first character of c_i for all i is 0, and s'_i (resp. c'_i) is a string which is removed the first character from s_i (resp. c_i). Therefore, C' is a common coloring of S' because C' holds the four conditions of common coloring. Since $dzero(C') = dzero(C) - 1$ and $done(C') = done(C)$, we can apply induction hypothesis for C'. C2P(C) outputs a pattern $p = s_1[1]q$ where $q = C2P(C')$ because headzero(C) & samehead(C) is true and $C' = shiftzero(C)$.

We have now fulfilled both conditions of the principle of mathematical induction.

Next, we prove that C is a common coloring of S if C2P(C) outputs a pattern p by double induction on $(dzero(C), done(C))$.

(1) Base Step: $dzero(C) = 0$. Since $sum_0(c_i) = 0$ for all i, $c_i \in \{1\}^+$. Under the situation, C is a common coloring of S because $\forall i, j\ c_i[1] = c_j[1]$ and $c_i[|c_i|] = c_j[|c_j|]$.

(2) Induction Step: Since $dzero(C) > 0$, there is some (s_i, c_i) such that c_i is not null. Since C2P(C) outputs a pattern p, we have (2-1) headone(C) or (2-2) headzero(C) & samehead(C).

(2-1) headone(C). We define the prefixes h_i of s_i (i.e. $s_i = h_i s'_i$) and d_i of c_i (i.e. $c_i = d_i c'_i$) $(1 \leq i \leq n)$, and consider $C' = \{(s'_1, c'_1), \ldots, (s'_n, c'_n)\}$ such that

$h_i = s_i[1:k_i]$ and $d_i = c_i[1:k_i]$ where $k_i = max\{1 \leq y \leq |c_i| \mid \forall x\ 1 \leq x \leq y, c_i[x] = 1\}$. Since $dzero(C) > 0$, we have (2-1-1) $\exists i\ |s'_i| = 0$ or (2-1-2) $\forall i\ |s'_i| \neq 0$.

(2-1-1) $\exists i\ |s'_i| = 0$. Then, make_pattern(C') returns "\$" and make_pattern(C) returns "x\$" where x is a variable. Therefore, C2P(C) returns failure. This contradicts the assumption.

(2-1-2) $\forall i\ |s'_i| \neq 0$. We can easily confirm that C' is a coloring of $S' = \{s'_1, \ldots, s'_n\}$ because $\forall i\ c'_i \in \{0,1\}^+$ and $|s'_i| = |c'_i|$. Since $dzero(C') = dzero(C)$ and $done(C') = done(C) - 1$, we can apply induction hypothesis for C'. We have $C2P(C) = xq$ where x is a variable and $q = C2P(C')$ because headone(C) is true and $C' = shiftone(C)$. In this situation, C is a common coloring of S because (1) $\forall i\ sum_0(c_i) = sum_0(c'_i)$, therefore $\forall i, j\ sum_0(c_i) = sum_0(c_j)$; (2) $\forall i, l\ s_i[z(c_i, l)] = s'_i[z(c'_i, l)]$ $(1 \leq l \leq sum_0(c_i))$, therefore $\forall i, j, l\ s_i[z(c_i, l)] = s_j[z(c_j, l)]$ $(1 \leq l \leq sum_0(c_1))$ and (3) $\forall i, j, l\ z(c_i, l+1) = z(c_i, l) + 1 \Leftrightarrow z(c_j, l+1) = z(c_j, l) + 1$; (4) $\forall i\ c_i[|c_i|] = c'_i[|c'_i|]$, therefore $\forall i, j\ c_i[|c_i|] = c_j[|c_j|]$ and $c_i[1] = c_j[1] = 1$.

(2-2) headzero(C) & samehead(C). We define the prefixes h_i of s_i (i.e. $s_i = h_i s'_i$) and d_i of c_i (i.e. $c_i = d_i c'_i$) $(1 \leq i \leq n)$, and consider $C' = \{(s'_1, c'_1), \ldots, (s'_n, c'_n)\}$ such that $h_i = s_i[1]$ and $d_i = c_i[1]$. Since $dzero(C) > 0$, we have (2-2-1) $\forall i\ |s'_i| = 0$ or (2-2-2) $\exists i\ |s'_i| \neq 0$.

(2-2-1) $\forall i\ |s'_i| = 0$. In this situation, $\forall i\ |s_i| = 1$ and $s_1 = \cdots = s_n$. C2P(C) outputs s_1. C is a common coloring of S because C holds the four conditions of common coloring.

(2-2-2) $\exists i\ |s'_i| \neq 0$. We have (2-2-2-1) $\exists j\ |s'_j| = 0$ or (2-2-2-2) $\forall j\ |s'_j| \neq 0$.

(2-2-2-1) $\exists j\ |s'_j| = 0$. Then, the function make_pattern(C') returns "\$" and make_pattern(C) returns "$s_1[1]$\$". Therefore, C2P(C) returns failure. This contradicts the assumption.

(2-2-2-2) $\forall j\ |s'_j| \neq 0$. We can easily confirm that C' is a coloring of $S' = \{s'_1, \ldots, s'_n\}$ because $\forall i\ c'_i \in \{0,1\}^+$ and $|s'_i| = |c'_i|$. Since $dzero(C') = dzero(C) - 1$ and $done(C') = done(C)$, we can apply induction hypothesis for C'. We have $C2P(C) = s_1[1]q$ where $q = C2P(C')$ because headzero(C) & samehead(C) is true and $C' = shiftzero(C)$. The first character of c_i for all i is 0, and s'_i (resp. c'_i) is a string which is removed the the first character from s_i (resp. c_i). Therefore, C is a common coloring of S because C holds the four conditions of common coloring.

We have now fulfilled both conditions of the principle of mathematical induction. □

We consider a pattern generated from each pair of two strings in a common coloring of a sample.

Lemma 2. *Let $S = \{s_1, s_2\}$ be a sample, and $C = \{(s_1, c_1), (s_2, c_2)\}$ be a common coloring of S. Let $C_1 = \{(s_1, c_1)\}$ and $C_2 = \{(s_2, c_2)\}$. Let $p_1 = C2P(C_1)$ and $p_2 = C2P(C_2)$. Then, $p_1 \equiv p_2$.*

Proof. We prove the lemma by double induction on $(dzero(C_1), done(C_1))$.

(1) Base Step: $dzero(C_1) = 0$, then $c_1 \in \{1\}^+$. Since C is a common coloring of S, $c_2 \in \{1\}^+$. We have $p_1 = C2P(C_1) = x$ and $p_2 = C2P(C_2) = y$ where x and y are variables. Therefore, $p_1 \equiv p_2$.

(2) Induction Step

(2-1) The last character of c_1 is 1. Since C is a common coloring of S, the last character of c_2 is also 1.

Since $dzero(C_1) > 0$, c_1 contains an occurrence of 0. Thus we have $c_1 = c'_1 0 z_1$ and $c_2 = c'_2 0 z_2$ for some $z_1, z_2 \in \{1\}^+$. Let $s'_1 = s_1[1 : |s_1| - |z_1|]$ and $s'_2 = s_2[1 : |s_2| - |z_2|]$ be strings. Let $C'_1 = \{(s'_1, c'_1 0)\}$ and $C'_2 = \{(s'_2, c'_2 0)\}$. Since $dzero(C'_1) = dzero(C_1)$ and $done(C'_1) = done(C_1) - 1$, we can apply induction hypothesis for C'_1 and C'_2. We have $C2P(C'_1) \equiv C2P(C'_2)$.

We have $p_1 = C2P(C_1) = C2P(C'_1)x$ and $p_2 = C2P(C_2) = C2P(C'_2)y$ where x and y are variables. Therefore, $p_1 \equiv p_2$.

(2-2) The last character of c_1 is 0. Since C is a common coloring of S, the last character of c_2 is also 0.

Therefore, we have $c_1 = c'_1 0$ and $c_2 = c'_2 0$ for some c'_1, c'_2. Let $s'_1 = s_1[1 : |s_1| - 1]$ and $s'_2 = s_2[1 : |s_2| - 1]$ be strings. Let $C'_1 = \{(s'_1, c'_1)\}$ and $C'_2 = \{(s'_2, c'_2)\}$. Since $dzero(C'_1) = dzero(C_1) - 1$ and $done(C'_1) = done(C_1)$, we can apply induction hypothesis for C'_1 and C'_2. We have $C2P(C'_1) \equiv C2P(C'_2)$.

We have $p_1 = C2P(C_1) = C2P(C'_1)d$ and $p_2 = C2P(C_2) = C2P(C'_2)d$ where $d = s_1[|s_1|]$. Therefore, $p_1 \equiv p_2$.

We have now fulfilled both conditions of the principle of mathematical induction. □

We consider the relation between the order of two colorings and the order of two patterns.

Lemma 3. *Let $S = \{s_1, \ldots, s_n\}$ be a sample, $C_1 = \{(s_1, c_{1,1}), \ldots, (s_n, c_{1,n})\}$ and $C_2 = \{(s_1, c_{2,1}), \ldots, (s_n, c_{2,n})\}$ be common colorings of S. Let $p_1 = C2P(C_1)$ and $p_2 = C2P(C_2)$ be patterns of S. Then, $C_1 \leq C_2 \Rightarrow p_1 \leq p_2$.*

Proof. Let $C'_1 = \{(s_1, c_{1,1})\}$ and $C'_2 = \{(s_1, c_{2,1})\}$. Note that $C2P(C_1) \equiv C2P(C'_1)$ and $C2P(C_2) \equiv C2P(C'_2)$. It suffices to show that $C2P(C_1) \leq C2P(C'_1)$. We prove this by induction on $(dzero(C'_2), done(C'_2))$.

(1) Base Step: $dzero(C'_2) = 0$, then we have $c_{2,1} \in \{1\}^+$. The output pattern that C2P(C'_2) returns is $p'_2 = x$ where x is a variable. Therefore, $C2P(C'_1) = p'_1 \leq p'_2$.

(2) Induction Step

(2-1) The last character of $c_{2,1}$ is 1.

Since $dzero(C'_2) > 0$, we have $c_{2,1} = c'_{2,1} 0 z_2$ where $z_2 \in \{1\}^+$. Since $C'_1 \leq C'_2$, the $(|c'_{2,1}| + 1)$-th character on $c_{1,1}$ is 0. Therefore, we have $c_{1,1} = c'_{1,1} 0 z_1$ where $z_1 \in \{0, 1\}^+$ such that $|c'_{1,1}| = |c'_{2,1}|$ and $|z_1| = |z_2|$.

Let $s'_1 = s_1[1 : |s_1| - |z_2|]$ be a string. Let $C''_1 = \{(s'_1, c'_{1,1} 0)\}$ and $C''_2 = \{(s'_1, c'_{2,1} 0)\}$ be common colorings of $S' = \{s'_1\}$. Since $dzero(C''_2) = dzero(C'_2)$ and $done(C''_2) = done(C'_2) - 1$, we can apply induction hypothesis for C''_1 and C''_2.

We have $p'_1 = C2P(C'_1) = C2P(C''_1)p''_1$ and $p'_2 = C2P(C'_2) = C2P(C''_2)x$ where p''_1 is a pattern and x is a variable. Therefore, $p'_1 \leq p'_2$.

(2-2) The last character of $c_{2,1}$ is 0. Since $C'_1 \leq C'_2$, the last character of $c_{1,1}$ is also 0.

We have $c_{1,1} = c'_{1,1}0$ and $c_{2,1} = c'_{2,1}0$. Let $s'_1 = s_1[1 : |s_1| - 1]$ be a string. Let $C''_1 = \{(s'_1, c'_{1,1})\}$ and $C''_2 = \{(s'_1, c'_{2,1})\}$ be common colorings of $S' = \{s'_1\}$. Since $dzero(C''_2) = dzero(C'_2) - 1$ and $done(C''_2) = done(C'_2)$, we can apply induction hypothesis for C''_1 and C''_2.

We have $p'_1 = C2P(C'_1) = C2P(C''_1)d$ and $p'_2 = C2P(C'_2) = C2P(C''_2)d$ where $d = s_1[|s_1|]$. Therefore, $p'_1 \leq p'_2$.

We have now fulfilled both conditions of the principle of mathematical induction. □

A minimum common coloring is defined as follows.

Definition 10. *A common coloring C of a sample S is a minimum common coloring of S iff there exists no common coloring C' of S such that $C' < C$.*

For instance, let $S = \{abcba, abaaba, abbbba\}$ be a sample, then a minimum common coloring of S is $\{(abcba, 00100), (abaaba, 001100), (abbbba, 001100)\}$.

Next, we consider a relation between a minimum common coloring and a descriptive pattern.

Theorem 2. *If C is a minimum common coloring of a sample S, then the pattern $p = C2P(C)$ is descriptive with respect to S.*

Proof. It suffices to show that $p \leq q$ for any pattern q of S. Let $q = e_1 \cdots e_l$ where $e_r \in \Sigma \cup V$ $(1 \leq r \leq l)$. Since q is a pattern of $S = \{s_1, \ldots s_n\}$, we have substitutions f^i $(i = 1, \ldots, n)$ such that $s_i = f^i(e_1) \cdots f^i(e_l)$. We define a coloring of s_i depending on whether e_r is a variable $(e_r \in V)$ or e_r is a constant $(e_r \in \Sigma)$ as follows:

$$c^i[j] = \begin{cases} 0 \text{ if } e_{pos(i,j)} \in \Sigma \\ 1 \text{ if } e_{pos(i,j)} \in V \end{cases}$$

where $pos(i, j) = min\{m \mid j \leq \sum_{k=1}^{m} |f^i(e_k)|\}$. Intuitively, $pos(i, j) = r$ means that the character at the j-th position on s_i comes from the r-th character e_r of the pattern $q = e_1 \cdots e_r \cdots e_l$.

We can see that $C' = \{(s_1, c^i_1), \ldots (s_n, c^i_n)\}$ is a common coloring and that $q = C2P(C')$. Since C is a minimum common coloring, we have $p \leq q$ by Lemma 3. □

This theorem shows that the algorithm C2P(C) outputs a descriptive pattern of S if C is a minimum common coloring of S. Therefore, the pattern detection problem in Section 3 is replaced as the following problem.

Definition 11 (Coloring Problem). *The coloring problem is, given a sample, to obtain a minimum common coloring of the sample.*

5 Coloring Algorithm II

We propose an another algorithm C2P' to generate a pattern from a coloring of a sample. Fig. 3 is the pseudo code of the algorithm. The algorithm C2P'

uses the function make_pattern() in the algorithm C2P. The algorithm generates a pattern from (s_i, c_i) for each i. If all generated patterns $p_1, \ldots, p_n$ are the same without renaming (i.e. same() is true), the algorithm outputs the pattern. Otherwise, it outputs failure.

Algorithm C2P'

```
Input: C = {(s_i, c_i) | 1 ≤ i ≤ n, s_i ∈ Σ^+, c_i ∈ {0,1}^+, |s_i| = |c_i|};
Output: pattern or failure;
var: p_1, ..., p_n : pattern;
       i : integer;
begin
  for (i = 1; i ≤ n; i++){
      p_i = make_pattern({(s_i, c_i)})
  }
  if same(p_1, ..., p_n){
    output p_1;
  } else {
    output failure;
  }
end;
```

Fig. 3. An another algorithm to generate a pattern from a coloring of a sample

Corollary 1. *Let $S = \{s_1, \ldots, s_n\}$ be a sample, and $C = \{(s_1, c_1), \ldots, (s_n, c_n)\}$ be a coloring of S. Let $p = C2P(C)$ and $p' = C2P'(C)$. Then, $p \equiv p'$.*

Proof. We prove the corollary by double induction on $(dzero(C), done(C))$.

(1) Base Step: $dzero(C) = 0$. Since $sum_0(c_i) = 0$ for all i, $c_i \in \{1\}^+$. First, we consider the algorithm C2P(C). Under the situation, headone(C) is true and make_pattern(shiftone(C)) returns the empty string. Therefore, C2P(C) outputs a pattern $p = x$ where x is a variable.

Next, we consider the algorithm C2P'(C). The function make_pattern($\{(s_i, c_i)\}$) for each i returns a pattern $p'_i = y_i$ where y_i is a variable. Since $\forall i, j\ p'_i \equiv p'_j$, C2P'($C$) outputs a pattern $p' = y_1$. Therefore, $p \equiv p'$.

(2) Induction Step:

(2-1) headone(C). We define the prefixes h_i of s_i (i.e. $s_i = h_i s'_i$) and d_i of c_i (i.e. $c_i = d_i c'_i$) $(1 \leq i \leq n)$, and consider $C' = \{(s'_1, c'_1), \ldots, (s'_n, c'_n)\}$ such that $h_i = s_i[1 : k_i]$ and $d_i = c_i[1 : k_i]$ where $k_i = max\{1 \leq y \leq |c_i| \mid \forall x\ 1 \leq x \leq y, c_i[x] = 1\}$.

(2-1-1) $\exists i\ |s'_i| = 0$. Then, make_pattern(C') returns "\$" and make_pattern(C) returns "x\$" where x is a variable. Therefore, C2P(C) returns failure.

The function make_pattern($\{(s_i, c_i\}$) for such i returns a pattern $p_i = y_i$ where y_i is a variable. However, make_pattern($\{(s_j, c_j\}$) such that $|s'_j| \neq 0$ re-

turns a pattern $p_j = y_j p'_j$ where p'_j is a pattern. Therefore, C2P'(C) also returns failure.

(2-1-2) $\forall i\ |s'_i| \neq 0$. We can easily confirm that C' is a coloring of $S' = \{s'_1, \ldots, s'_n\}$ because $\forall i\ c'_i \in \{0,1\}^+$ and $|s'_i| = |c'_i|$. Since $dzero(C') = dzero(C)$ and $done(C') = done(C) - 1$, we can apply induction hypothesis for C' and $q = C2P(C') = C2P'(C')$.

In the situation, C2P(C) outputs a pattern $p = xq$ where x is a variable because headone(C) is true and $C' = shiftone(C)$.

The function make_pattern($\{(s_i, c_i\}$) for each i returns a pattern $p_i = y_i q$ where y_i is a variable. The pattern output by C2P'(C') is $p' = y_1 q$ because all generated patterns by the function are the same. Therefore, $p \equiv p'$.

(2-2) headzero(C) & samehead(C). We define the prefixes h_i of s_i (i.e. $s_i = h_i s'_i$) and d_i of c_i (i.e. $c_i = d_i c'_i$) ($1 \leq i \leq n$), and $C' = \{(s'_1, c'_1), \ldots, (s'_n, c'_n)\}$ such that $h_i = s_i[1]$ and $d_i = c_i[1]$.

(2-2-1) $\forall i\ |s'_i| = 0$. In this situation, $\forall i\ |s_i| = 1$ and $s_1 = \cdots = s_n$. The pattern output by C2P(C) is $p = s_1$.

On the other hand, the function make_pattern($\{(s_i, c_i\}$) for each i returns a pattern $p_i = s_i$. The pattern output by C2P'(C) is $p' = s_1$. Therefore, $p \equiv p'$.

(2-2-2) $\exists i\ |s'_i| \neq 0$. We have (2-2-2-1) $\exists j\ |s'_j| = 0$ or (2-2-2-2) $\forall j\ |s'_j| \neq 0$.

(2-2-2-1) $\exists j\ |s'_j| = 0$. Then, the function make_pattern(C') returns "\$" and make_pattern(C) returns "$s_1[1]$\$". Therefore, C2P(C) returns failure.

The function make_pattern($\{(s_i, c_i\}$) for such i returns a pattern $p_i = s_i[1]p'_i$ where p'_i is a pattern. However, make_pattern($\{(s_j, c_j\}$) for such j returns a pattern $p_j = s_j[1]$. Therefore, C2P'(C) also returns failure.

(2-2-2-2) $\forall j\ |s'_j| \neq 0$. We can easily confirm that C' is a coloring of $S' = \{s'_1, \ldots, s'_n\}$ because $\forall i\ c'_i \in \{0,1\}^+$ and $|s'_i| = |c'_i|$. Since $dzero(C') = dzero(C) - 1$ and $done(C') = done(C)$, we can apply induction hypothesis for C' and $q = C2P(C') = C2P'(C')$.

In the situation, the pattern output by C2P(C) is $p = s_1[1]q$ because headzero(C) & samehead(C) is true and $C' = shiftzero(C)$.

The function make_pattern($\{(s_i, c_i\}$) for each i returns a pattern $p_i = s_i[1]q$. The pattern output by C2P'(C) is $p' = s_1[1]q$. Therefore, $p \equiv p'$.

(2-3) Otherwise. The situations are that (1) there exist integers i, j such that $c_i[1] \neq c_j[1]$, and (2) $c_i[1] = 0$ for all i but there exist integers i, j such that $s_i[1] \neq s_j[1]$, and (3) there exists an integer i such that (s_i, c_i) is (null, null) but there exists an integer j such that (s_j, c_j) is not (null, null). In these situations, C2P(C) returns failure. On the other hand, there exist integers i, j such that make_pattern($\{s_i, c_i\}$) $\neq$ make_pattern($\{s_j, c_j\}$). Therefore, C2P'(C) also returns failure.

We have now fulfilled both conditions of the principle of mathematical induction. □

6 Component Set

This section considers how to solve the coloring problem. A proposing approach is to find a set of substrings of strings in a sample. The set is called a *component set*.

Definition 12. *Let $S = \{s_1, \ldots, s_n\}$ be a sample, $C = \{(s_1, c_1), \ldots, (s_n, c_n)\}$ be a coloring of S. A set CS of strings is a component set of S with respect to C iff for all $w \in CS$ there exists $s_i \in S$ such that $w = s_i[k : k + |w| - 1]$ and $c_i[j] = 0$ for all j such that $k \leq j \leq k + |w| - 1$.*

For instance, let $C = \{(cabaabaac, 000000010), (cabbabac, 01110000)\}$ be a coloring of a sample $S = \{cabaabaac, cabbabac\}$. A component set of S with respect to C is $\{aba, c\}$.

A component set divides each string in a sample into two regions which are colored and non-colored regions. If we find a component set with respect to a minimum common coloring of a sample, the algorithm C2P outputs a descriptive pattern of the sample from the coloring. Therefore, the coloring problem is replaced as the following problem.

Definition 13 (Component Set Discovery Problem). *The component set discovery problem is, given a sample, to find a component set of the sample with respect to a minimum common coloring of the sample.*

We proposed algorithms to discovery a component set and showed the effectiveness of the algorithms by experiments using Web documents [8, 9].

7 Conclusion

We described pattern detection of Web documents with the same type of contents using pattern languages. The pattern detection problem of the Web documents is to find a descriptive regular pattern, such that successive variables do not appear in the pattern, of a given sample.

Instead of detecting a pattern of a sample directory, our pattern detection approach is to divide each string in the sample into two regions which are colored and non-colored regions. A coloring of a sample is a set of pairs of each string in the sample and a string over $\{0, 1\}$. We proposed an algorithm to generate a pattern of a sample from a coloring of the sample. Under the approach, the pattern detection problem is replaced as the coloring problem which is to obtain a minimum common coloring of a given sample.

In order to solve the coloring problem, our approach is to find a component set which is a set of substrings of strings in a sample. A component set determines a coloring of a sample. Therefore, the coloring problem is replaced as the component set discovery problem which is to find a component set of a given sample with respect to a minimum common coloring of the sample.

We proposed component set discovery algorithms in [8, 9], where we showed the effectiveness of the algorithms by experiments using actual Web documents. The theoretical analysis of the algorithms is an important task.

References

1. D. Angluin. Finding Patterns Common to a Set of Strings. *Journal of Computer and System Sciences*, 21:46–62, 1980.
2. A. Arasu and H. Garcia-Molina. Extracting Structured Data from Web Pages. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 337–348. ACM Press, 2003.
3. Z. Bar-Yossef and S. Rajagopalan. Template Detection via Data Mining and its Applications. In *Proceedings of the 11th international Conference on World Wide Web*, pages 580–591. ACM Press, 2002.
4. D. Buttler, L. Liu, and C. Pu. A Fully Automated Object Extraction System for the World Wide Web. In *Proceedings of the 21th International Conference on Distributed Computing Systems*, pages 361–370, 2001.
5. C.-H. Chang and S.-C. Lui. IEPAD: Information Extraction Based on Pattern Discovery. In *Proceedings of the 10th International Conference of World Wide Web*, pages 681–688. ACM Press, 2001.
6. V. Crescenzi, G. Mecca, and P. Merialdo. Road Runner: Towards Automatic Data Extraction from Large Web Sites. In *Proceedings of the 27th International Conference on Very Large Data Bases*, pages 109–118, 2001.
7. T. W. Hong and K. L. Clark. Using Grammatical Inference to Automate Information Extraction from the Web. In *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, volume 2168 of *Lecture Notes in Computer Science*, pages 216–227. Springer-Verlag, 2001.
8. D. Ikeda, Y. Yamada, and S. Hirokawa. Eliminating Useless Parts in Semi-structured Documents using Alternation Counts. In *Proceedings of the 4th International Conference on Discovery Science*, volume 2226 of *Lecture Notes in Artificial Intelligence*, pages 113–127. Springer-Verlag, 2001.
9. D. Ikeda, Y. Yamada, and S. Hirokawa. A Pattern Discovery Algorithm by Substring Amplification. *IPSJ Transactions on Mathematical Modeling and Its Applications*, 46(SIG 2 (TOM 11)):56–66, 2005. (in Japanese).
10. K. Lerman, C. A. Knoblock, and S. N. Minton. Automatic Data Extraction from Lists and Tables in Web Sources. In *Proceedings of Workshop on Adaptive Text Extraction and Mining*, 2001.
11. S.-H. Lin and J.-M. Ho. Discovering Informative Content Blocks from Web Documents. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 588–593. ACM Press, 2002.
12. T. Shinohara. Polynomial Time Inference of Extended Regular Pattern Languages. In *Proceedings of the RIMS Symposium on Software Science and Engineering*, Lecture Notes in Computer Science, pages 115–127. Springer-Verlag, 1983.
13. Y. Yamada, D. Ikeda, and S. Hirokawa. Automatic Wrapper Generation for Multilingual Web Resources. In *Proceedings of the 5th International Conference on Discovery Science*, volume 2534 of *Lecture Notes in Computer Science*, pages 332–339. Springer-Verlag, 2002.
14. L. Yi, B. Liu, and X. Li. Eliminating Noisy Information in Web Pages for Data Mining. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 296–305. ACM Press, 2003.