



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0075295
(43) 공개일자 2020년06월26일

(51) 국제특허분류(Int. Cl.)
G06F 16/22 (2019.01) G06F 7/02 (2006.01)
(52) CPC특허분류
G06F 16/22 (2019.01)
G06F 7/02 (2013.01)
(21) 출원번호 10-2018-0163814
(22) 출원일자 2018년12월18일
심사청구일자 2018년12월18일

(71) 출원인
인하대학교 산학협력단
인천광역시 미추홀구 인하로 100(용현동, 인하대학교)
(72) 발명자
김정섭
인천광역시 남동구 남동대로 860, 120동 601호(간석동, 간석래미안자이아파트)
유광모
서울특별시 영등포구 시흥대로 589-8, 102동 1404호(대림동, 신대림자이)
김영호
인천광역시 미추홀구 낙섬동로 6, 102동 301호(용현동, 한국아파트)
(74) 대리인
양성보

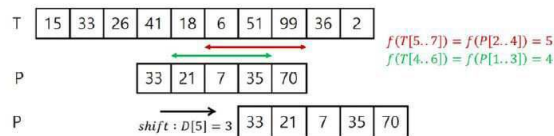
전체 청구항 수 : 총 7 항

(54) 발명의 명칭 2차 q-그램에 대한 핑거프린트를 이용한 순위패턴매칭 알고리즘

(57) 요약

일 실시예에 따른 핑거프린트를 이용한 순위패턴매칭 방법은, 패턴(P)에 대한 위치테이블, 이동테이블 및 복수 개의 q-그램에 대한 핑거프린트를 계산하는 전처리 단계; 및 상기 패턴(P)에 대한 위치테이블, 상기 이동테이블 및 상기 복수 개의 q-그램에 대한 핑거프린트를 계산한 값을 이용하여 상기 패턴(P)과 순위동형인 텍스트(T)의 부분문자열을 탐색하는 계산 단계를 포함할 수 있다.

대표도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호 1711069835

부처명 과학기술정보통신부

연구관리전문기관 한국연구재단

연구사업명 전략과제(산업수학)

연구과제명 [Ezbaro] 대용량 시계열 데이터의 반복성 검출과 패턴 예측을 위한 새로운 모델 및 효율적
인 알고리즘 개발

기 여 율 1/1

주관기관 인하대학교

연구기간 2018.03.01 ~ 2019.02.28

명세서

청구범위

청구항 1

전처리부 및 계산부를 포함하는 순위패턴매칭 장치에서 수행되는 핑거프린트를 이용한 순위패턴매칭 방법에 있어서,

상기 전처리부에서, 패턴(P)에 대한 위치테이블, 이동테이블 및 복수 개의 q-그램에 대한 핑거프린트를 계산하는 전처리 단계; 및

상기 계산부에서, 상기 패턴(P)에 대한 위치테이블, 상기 이동테이블 및 상기 복수 개의 q-그램에 대한 핑거프린트를 계산한 값을 이용하여 상기 패턴(P)과 순위동형인 텍스트(T)의 부분문자열을 탐색하는 계산 단계를 포함하는 순위패턴매칭 방법.

청구항 2

제1항에 있어서,

상기 복수 개의 q-그램은,

상기 패턴(P)과 텍스트(T)가 주어졌을 때, 상기 패턴(P)의 가장 오른쪽 q-그램을 1차 q-그램, 상기 1차 q-그램을 기준으로 상기 1차 q-그램의 이전에 위치한 q-그램을 2차 q-그램이라고 하고, 상기 1차 q-그램 및 상기 2차 q-그램을 포함하는 순위패턴매칭 방법.

청구항 3

제1항에 있어서,

상기 계산 단계는,

$P[m-1]$ 과 비교를 시작하는 텍스트(T)의 위치를 i 라고 할 때, 각 $i(m-1 \leq i < n)$ 에 대해 1차 q-그램의 핑거프린트인 $f(P[m-q..m-1])$ 와 $f(T[i-q+1..i])$ 가 일치하는지 판단하는 단계

를 포함하는 순위패턴매칭 방법.

청구항 4

제3항에 있어서,

상기 계산 단계는,

상기 각 $i(m-1 \leq i < n)$ 에 대해 1차 q-그램의 핑거프린트인 $f(P[m-q..m-1])$ 와 $f(T[i-q+1..i])$ 가 일치할 경우, 2차 q-그램의 핑거프린트인 $f(P[m-q-1..m-2])$ 와 $f(T[i-q..i-1])$ 가 일치하는지 확인하는 단계

를 포함하는 순위패턴매칭 방법.

청구항 5

제4항에 있어서,
상기 계산 단계는,

상기 2차 q-그램의 핑거프린트인 $f(P[m-q-1..m-2])$ 와 $f(T[i-q..i-1])$ 가 일
치할 경우, 패턴(P)와 $T[i-m+1..i]$ 의 순위동형 여부를 위치테이블을 이용하여 확인하고,
 $P \approx T[i-m+1..i]$ 을 단계
를 포함하는 순위패턴매칭 방법.

청구항 6

제1항에 있어서,
상기 계산 단계는,

순위동형 여부에 관계없이 텍스트의 위치(i)를 $D[f(T[i-q+1..i])]$ 만큼 증가시키는 단계
를 포함하는 순위패턴매칭 방법.

청구항 7

순위패턴매칭 장치에 있어서,

패턴(P)에 대한 위치테이블, 이동테이블 및 복수 개의 q-그램에 대한 핑거프린트를 계산하는 전처리부; 및

상기 패턴(P)에 대한 위치테이블, 상기 이동테이블 및 상기 복수 개의 q-그램에 대한 핑거프린트를 계산한 값을
이용하여 상기 패턴(P)과 순위동형인 텍스트(T)의 부분문자열을 탐색하는 계산부

를 포함하는 순위패턴매칭 장치.

발명의 설명

기술 분야

[0001] 아래의 설명은 복수 개의 q-그램에 대한 핑거프린트를 이용하여 수행시간을 개선한 순위패턴매칭 알고리즘에 관
한 것이다.

배경 기술

[0003]두 문자열의 길이가 같고, 각 문자열에서 문자들의 상대적인 순위가 모든 위치에서 일치하면 순위동형이라
한다. 예를 들면, $T=(15, 33, 26, 41, 7, 6, 51)$ 와 $P=(33, 21, 7)$ 가 주어졌을 때,

$T[3..5] = (41, 7, 6)$ 과 P 는 순위동형이다. 순위패턴매칭문제(order-preserving pattern matching

problem)는 대소비교가 가능한 알파벳 Σ 로 구성된 텍스트 T 와 패턴 P 가 주어졌을 때, P 와 순위동형인 T 의 모
든 부분문자열을 찾는 문제이다. 이때, 순위패턴매칭문제는 두 음악의 유사멜로디 검출이나 주가분석 등에 활
용될 수 있다.

[0004]최근 순위패턴매칭문제를 해결하기 위한 연구들이 활발하게 진행되고 있다. 길이가 각각 n, m 인 텍스트 T 와 패
턴 P 가 주어졌을 때, 순위패턴매칭문제를 KMP알고리즘을 기반으로 $O(n+m \log m)$ 시간에 해결하는 알고리
즘이 제시되었다. 다른 연구에서는 P 와 순위동형인 T 의 부분문자열의 개수를 occ 라 할 때, 순위접미사트리

(order-preserving suffix trees)를 $O(n\sqrt{\log n})$ 시간에 생성하는 알고리즘을 제시하고, 이를 이용하여 순위패턴매칭문제를 $O(m+occ)$ 시간에 해결하는 알고리즘을 제시하였다. 또 다른 연구에서는 q 개의 연속된 문자인 q -그램(q -gram)에 대한 핑거프린트(fingerprint)와 Horspool 알고리즘의 오문자규칙(bad-character heuristic)을 이용하여, 정수 알파벳일 때 $O(nm+nq+q!)$ 시간, 일반 알파벳일 때 $O(nm+nq\log q+q!)$ 시간에 순위패턴매칭문제를 해결하는 알고리즘을 제시하였고, KMP 알고리즘을 함께 이용하여 정수 알파벳일 때 $O((m+n)q+q!)$ 시간, 일반 알파벳일 때 $O((m+n)q\log q+q!+m\log m)$ 시간에 순위패턴매칭문제를 해결하는 알고리즘이 제시되었다.

발명의 내용

해결하려는 과제

[0006] 복수 개의 q -그램에 대한 핑거프린트들을 이용한 순위패턴매칭 알고리즘을 제공할 수 있다.

과제의 해결 수단

[0008] 전처리부 및 계산부를 포함하는 순위패턴매칭 장치에서 수행되는 핑거프린트를 이용한 순위패턴매칭 방법은, 상기 전처리부에서, 패턴(P)에 대한 위치테이블, 이동테이블 및 복수 개의 q -그램에 대한 핑거프린트를 계산하는 전처리 단계; 및 상기 계산부에서, 상기 패턴(P)에 대한 위치테이블, 상기 이동테이블 및 상기 복수 개의 q -그램에 대한 핑거프린트를 계산한 값을 이용하여 상기 패턴(P)과 순위동형인 텍스트(T)의 부분문자열을 탐색하는 계산 단계를 포함할 수 있다.

[0009] 상기 복수 개의 q -그램은, 상기 패턴(P)과 텍스트(T)가 주어졌을 때, 상기 패턴(P)의 가장 오른쪽 q -그램을 1차 q -그램, 상기 1차 q -그램을 기준으로 상기 1차 q -그램의 이전에 위치한 q -그램을 2차 q -그램이라고 하고, 상기 1차 q -그램 및 상기 2차 q -그램을 포함할 수 있다.

[0010] 상기 계산 단계는, $P[m-1]$ 과 비교를 시작하는 텍스트(T)의 위치를 i 라고 할 때, 각 $i(m-1 \leq i < n)$ 에 대해 1차 q -그램의 핑거프린트인 $f(P[m-q..m-1])$ 와 $f(T[i-q+1..i])$ 가 일치하는지 판단하는 단계를 포함할 수 있다.

[0011] 상기 계산 단계는, 상기 각 $i(m-1 \leq i < n)$ 에 대해 1차 q -그램의 핑거프린트인 $f(P[m-q..m-1])$ 와 $f(T[i-q+1..i])$ 가 일치할 경우, 2차 q -그램의 핑거프린트인 $f(P[m-q-1..m-2])$ 와 $f(T[i-q..i-1])$ 가 일치하는지 확인하는 단계를 포함할 수 있다.

[0012] 상기 계산 단계는, 상기 2차 q -그램의 핑거프린트인 $f(P[m-q-1..m-2])$ 와

$f(T[i-q..i-1])$ 가 일치할 경우, 패턴(P)와 $T[i-m+1..i]$ 의 순위동형 여부를 위치테이블을 이용하여 확인하고, $P \approx T[i-m+1..i]$ 을 단계를 포함할 수 있다.

[0013] 상기 계산 단계는, 순위동형 여부에 관계없이 텍스트의 위치(i)를 $D[f(T[i-q+1..i])]$ 만큼 증가시키는 단계를 포함할 수 있다.

[0014] 순위패턴매칭 장치는, 패턴(P)에 대한 위치테이블, 이동테이블 및 복수 개의 q-그램에 대한 핑거프린트를 계산하는 전처리부; 및 상기 패턴(P)에 대한 위치테이블, 상기 이동테이블 및 상기 복수 개의 q-그램에 대한 핑거프린트를 계산한 값을 이용하여 상기 패턴(P)과 순위동형인 텍스트(T)의 부분문자열을 탐색하는 계산부를 포함할 수 있다.

발명의 효과

[0016] 2개의 q-그램을 이용하여 순위동형을 검증하는 횟수를 감소시켜 수행속도를 개선할 수 있다.

도면의 간단한 설명

[0018] 도 1은 일 실시예에 따른 2개의 q-그램에 대한 핑거프린트들을 이용한 순위패턴매칭 알고리즘을 나타낸 예이다.
 도 2는 일 실시예에 따른 순위패턴매칭 장치의 구성을 설명하기 위한 블록도이다.
 도 3은 일 실시예에 따른 순위패턴매칭 장치에서 수행되는 순위패턴매칭 방법을 설명하기 위한 흐름도이다.
 도 4는 일 실시예에 있어서, 텍스트 T와 q-그램 핑거프린트의 분포율을 나타낸 그래프이다.

발명을 실시하기 위한 구체적인 내용

[0019] 이하, 실시예를 첨부한 도면을 참조하여 상세히 설명한다.

[0021] 알파벳 Σ 는 상수시간에 비교 가능한 정수문자 집합이다. Σ 로 구성된 길이가 m인 문자열들의 집합을 Σ^m 로 표기한다. 문자열 x 의 길이를 $|x|$, x 의 i번째 $(0 \leq i < |x|)$ 문자를 $x[i]$ 로 표시한다. 문자열 x 의 i번째부터 j번째까지의 부분문자열 $x[i], x[i+1], \dots, x[j]$ 를 $x[i..j] (0 \leq i \leq j < |x|)$ 로 표기한다. $x[i..j]$ 에서 $i=0$ 이면, 이를 x 의 접두사(prefix)라고 하고, $j=|x|-1$ 이면 x 의 접미사(suffix)라 한다. 두 문자열 x, y 에 대해 $|x|=|y|$ 이고 $x[i] \leq x[j] \Leftrightarrow y[i] \leq y[j] \quad (0 \leq i, j < |x|)$ 이면 x, y 는 순위동형이라 하고 $x \approx y$ 로 표기한다.

[0022] 문자열의 순위관계를 표현하는 방법으로 접두사표현(prefix representation)과 최근접이웃표현(nearest neighbor representation)이 있다. 문자열 x 의 접두사표현은 접두사테이블(prefix table) μ_x 를 이용하여 정의된다.

$$\mu_x[i] = |\{j: x[j] < x[i], 0 \leq j < i\}|$$

[0023]

다시 말해서, $\mu_x[i]$ 는 문자열 $x[0..i-1]$ 에서 $x[i]$ 보다 작은 문자들의 수이다. 문자열 x, y ($|x|=|y|$) 에 대한 μ_x, μ_y 가 주어졌을 때, $\mu_x = \mu_y$ 이면, $x \approx y$ 이다.

[0024]

문자열 x 에 대한 최근접이웃표현은 위치테이블(location table) $LMax_x, LMin_x$ 을 이용하여 정의된다.

[0025]

$$LMax_x[i] = j \text{ if } x[j] = \max\{x[k] : k \in [0, i-1], x[k] < x[i]\}$$

[0026]

다시 말해서, $LMax_x[i]$ 는 $x[0..i-1]$ 에서 $x[i]$ 보다 작은 값 중 가장 큰 값의 위치 j ($0 \leq j < i$) 이고, $LMin_x[i]$ 은 $x[0..i-1]$ 에서 $x[i]$ 보다 큰 값 중 가장 작은 값의 위치 j 이다. 만약 그러한 j 가 복수 개 존재한다면, 가장 큰 위치를 저장한다. 만약, 그러한 위치 j 가 존재하지 않으면, -1을 저장한다. 문자열 x, y ($|x|=|y|$) 와 $LMax_x, LMin_x$ 가 주어졌을 때, $y[LMax_x[i]] < y[i] < y[LMin_x[i]]$ ($1 \leq i < |x|$) 이면, $x \approx y$ 이다.

[0027]

표 1은 문자열 $P=(11, 83, 32, 40, 9, 56)$ 일 때, P 에 대한 점두사태이블 μ_P 와 위치테이블 $LMax_P, LMin_P$ 를 보여준다. 점두사태이블은 순위통계트리, 위치테이블은 정렬알고리즘을 이용하여 각각 $O(|P| \log |P|)$ 시간에 계산할 수 있다.

[0028]

표 1: $P=(11, 83, 32, 40, 9, 56)$ 에 대한 점두사태이블과 위치테이블

[0029]

i	0	1	2	3	4	5
P	11	83	32	40	9	56
μ_P	0	1	1	2	0	4
$LMax_P$	-1	0	0	2	-1	3
$LMin_P$	-1	-1	1	1	0	1

[0030]

순위패턴매칭문제에서 입력은 텍스트 $T(\in \Sigma^n)$, 패턴 $P(\in \Sigma^m)$ 이고, 출력은 $P \approx T[i..i+m-1]$ 인 모든 위치 i 이다. 편의상 P 와 T 는 서로 다른 정수로 구성된다고 가정한다.

[0031]

도 1은 일 실시예에 따른 2개의 q -그램에 대한 핑거프린트들을 이용한 순위패턴매칭 알고리즘을 나타낸 예이다.

[0032]

실시예에서는 복수 개의 q -그램, 예를 들면, 2개의 q -그램을 이용하여 순위동형을 검증하는 횟수를 감소시켜 수

[0033]

행속도를 개선하는 방법을 설명하기로 한다. 예를 들면, q 개의 연속된 문자인 q -그램 x 에 대한 핑거프린트

$f(x)$ 가 다음과 같이 정의될 수 있다.

$$f(x) = \sum_{k=0}^{q-1} \mu_x[k] \times k!$$

이때, $f(x)$ 는 계승수체계(factorial number system)를 사용하므로 $0 \sim q!-1$ 중 하나의 정수로 변환될 수 있다.

예를 들면, $q=3$ 일 때, q -그램 $x=(11, 83, 32)$ 라면, $\mu_x=(0, 1, 1)$ 이며, x 에 대한 핑거프린트

$f(x)=(0 \times 0!)+(1 \times 1!)+(1 \times 2!)=3$ 이다.

$f(P[m-q..m-1])=f(T[i-q+1..i])$ 일 때,

$f(P[m-q-1..m-2])=f(T[i-q..i-1])$ 의 일치여부를 확인한 뒤 일치하였을

경우에 위치테이블을 이용하여 순위동형 여부를 확인한다. 이때, $f(P[m-q..m-1])$ 는 패턴 P 의 가장 오른쪽 q -그램 핑거프린트를 의미할 수 있다.

도 2는 일 실시예에 따른 순위패턴매칭 장치의 구성을 설명하기 위한 블록도이고, 도 3은 일 실시예에 따른 순위패턴매칭 장치에서 수행되는 순위패턴매칭 방법을 설명하기 위한 흐름도이다.

실시예에서 제시하는 알고리즘은 전처리 단계(310)와 계산 단계(320)로 구성될 수 있다. 이러한 알고리즘은 전처리부(210)와 계산부(220)를 포함하는 순위패턴매칭 장치(200)에 의해 수행될 수 있다.

$P[m-q..m-1]$ 을 P 의 1차 q -그램, $P[m-q-1..m-2]$ 을 P 의 2차 q -그램이라고 하자. 다시 말해서, 2차 q -그램은 1차 q -그램의 바로 앞에 위치한 q -그램이 된다.

전처리 단계(310)에서 전처리부(210)는 패턴 P 에 대한 위치테이블과 이동테이블, 2개의 q -그램에 대한 핑거프린트를 계산할 수 있다. 이때, 예를 들면, 이동테이블 D 는 모든 가능한 q -그램 x 에 대해 최대이동거리 $m-q+1$ 로 초기화한 뒤, 다음의 수식을 이용하여 계산될 수 있다. 이동테이블 D 는 모든 가능한 q -그램에 대한 정보를 저장할 수 있다.

$$t = \max\{j \mid \mu_{P[j-q+1..j]} = \mu_x, q-1 \leq j < m-1\},$$

$$D[f(x)] = \min(m-q+1, m-t-1)$$

계산 단계(320)에서 계산부(220)는 패턴 P 에 대한 위치테이블과 이동테이블, 2개의 q -그램에 대한 핑거프린트를 계산한 각각의 값을 이용하여 패턴 P 와 순위동형인 텍스트 T 의 모든 부분문자열을 탐색할 수 있다.

구체적으로, $P[m-1]$ 과 비교를 시작하는 텍스트 T 의 위치를 i 라고 할 때, 각 $i(m-1 \leq i < n)$ 에 대해 1차 q -그램의 핑거프린트인 $f(P[m-q..m-1])$ 와

$f(T[i-q+1..i])$ 가 일치하면 2차 q -그램의 핑거프린트인 $f(P[m-q-1..m-2])$ 와

$f(T[i-q..i-1])$ 가 일치하는지 확인한다. 만약, 2차 q -그램의 핑거프린트도 일치하면, 패턴 P 와 $T[i-m+1..i]$ 의 순위동형 여부를 위치테이블을 이용하여 확인하여 $P \approx T[i-m+1..i]$

이면, 위치 $i-m+1$ 을 출력할 수 있다. 이후, 순위동형 여부에 관계없이 i 를 만큼 증가시키고 위 과정을 반복할 수 있다. $D[f(T[i-q+1..i])]$

[0042] 도 1을 참고하면, 예를 들면, $T=(15, 33, 26, 41, 18, 6, 51, 99, 36, 2)$ 와 $P=(33, 21, 7, 35, 70)$ 이 주어졌을 때, 단계 7은 다음과 같이 수행될 수 있다. 텍스트 T 와 패턴 P 의 1차 q -그램의 핑거프린트가 5로 일치하므로 2차 q -그램의 핑거프린트를 계산하고 비교한다. 이 두 값도 4로 일치하므로 두 문자열이 순위동형인지

확인한다. 순위동형을 확인한 결과, $P \approx T[3..7]$ 이므로 3을 출력하고 이동테이블 $D[5]=3$ 을 이용하여 비교위치 i 를 10으로 증가시킨다. 이때, $i > n-1$ 이므로 탐색을 종료한다.

[0043] 실시예에서 제안된 알고리즘의 전처리 단계에서 패턴 P 에 대한 위치테이블, 이동테이블은 각각

$O(m \log m)$, $O(q! + mq \log q)$ 시간에 계산할 수 있다. 계산 단계에서 q -그램들의 핑거프린트 계산과, P 와 $T[i-m+1..i]$ 의 순위동형 확인은 각각 총 $O(nq \log q)$ 시간, $O(nm)$ 시간에 수행할 수 있다. 이에 따라, 실시예에서 제안된 알고리즘의 총 수행시간은 $O(nm + nq \log q + q!)$ 시간이 된다.

[0044] 도 4는 일 실시예에 있어서, 텍스트 T 와 q -그램 핑거프린트의 분포율을 나타낸 그래프이다.

[0045] 실시예에서 제안된 알고리즘에 대한 실험이 수행될 수 있다. 실험 환경으로 OS는 Windows 10(64 비트), CPU 및 RAM은 Intel Core i7-6700(3.4GHz), 32GB RAM, 개발 툴 및 언어는 Visual studio 2015, c++이 구성될 수 있다.

텍스트 T 와 패턴 P 는 C++ boost library를 이용하여 $|\Sigma| = 2^{32}$ 에 대해 무작위로 생성한 정수형 문자열 ($n=5,000,000$)과 1985년 2월 16일부터 2010년 12월 31일까지의 다우존스산업평균지수의 일별 증가($n=34,685$)로 생성한 다우존스지수 데이터를 이용하였다. m 은 5, 10, 15에 대해, q 는 3부터 7까지 변경하며 실험이 진행될 수 있다.

[0046] 수행시간은 무작위로 생성된 정수형 문자열에 대해 1000번 실험한 수행시간의 합으로, 다우존스지수 데이터에 대해 10000번 실험한 수행시간의 합이 측정될 수 있다. 무작위로 생성한 문자열의 경우 패턴 또한 무작위로 생성될 수 있으며, 다우존스지수 데이터에서 패턴은 T 의 위치에서 무작위로 추출될 수 있다. 편의상 종래의 연속된 문자인 q -그램(q -gram)에 대한 핑거프린트(fingerprint)와 Horspool 알고리즘의 오문자규칙(bad-character heuristic)을 이용하여 순위패턴매칭문제를 해결하는 알고리즘을 OPQ, 실시예에서 제안된 알고리즘을 OPSQ로 기재하여 설명하기로 한다.

[0047] 제1 실험으로서, 입력 데이터에 대한 수행시간을 비교하는 실험이 수행될 수 있다. 표 2는 입력 데이터에 따른 OPQ와 OPSQ의 수행시간을 나타낸 것이다. 각 실험에서 수행시간이 가장 빠른 알고리즘을 진하게 표시하였다.

[0048] 표 2: $m=4, 10, 15, 3 \leq q \leq 7$ 에 대한 OPQ와 OPSQ의 수행시간

Input data	Algorithm	m	Execution time(second)				
			q = 3	q = 4	q = 5	q = 6	q = 7
ran	OPQ	5	17.406	19.51	-	-	-
	OPSQ		15.125	18.018	-	-	-
	OPQ	10	10.228	6.195	8.598	12.762	22.543
	OPSQ		8.976	5.789	7.612	11.922	20.9
	OPQ	15	9.1	4.275	4.827	6.303	10.122
	OPSQ		8.195	4.017	4.285	6.001	9.463
dow	OPQ	5	1.704	1.913	-	-	-
	OPSQ		1.46	1.857	-	-	-
	OPQ	10	1.008	0.794	0.925	1.197	1.798
	OPSQ		0.908	0.783	0.909	1.191	1.755
	OPQ	15	0.867	0.585	0.59	0.647	0.903
	OPSQ		0.758	0.577	0.583	0.654	0.891

[0049]

[0050]

ran은 무작위로 생성된 입력 데이터, dow는 다우존스지수 입력 데이터를 의미한다. ran의 경우, $m=10, q=3$ 일 때, OPSQ가 OPQ보다 약 12% 빠르게 수행되었다. m, q 의 크기와 상관없이 모든 경우에서 OPSQ가 OPQ보다 평균 약 9% 빠른 수행시간을 보여준다. $|\Sigma| = 4, 10$ 인 경우에도 유사한 결과를 보인다.

[0051]

dow의 경우, $m=10, q=3$ 일 때, OPSQ가 OPQ보다 최대 약 10% 빠르게 수행되었다. 그런데, $q \geq 4$ 일 때에는 ran의 경우와 달리, OPSQ의 수행시간은 OPQ에 비해 크게 개선되지 않고 평균적으로 1% 정도 빠르게 수행되었다. 특히, $m=15, q=6$ 일 때, OPSQ는 OPQ보다 오히려 느린 수행시간을 보인다. 이러한 결과에 대해서 다음 실험을 통하여 분석할 수 있다.

[0052]

제2 실험은 알고리즘의 성능에 영향을 미치는 요소는 분석한 것이다. 제1 실험의 결과, 실시예에서 제안된 알고리즘은 dow의 경우, $q \geq 4$ 일 때 기존의 알고리즘보다 수행시간이 크게 개선되지 않았다. 이는 다우존스지수의 특성상 q -그램 핑거프린트가 균일하게 나타나지 않기 때문이라 유추될 수 있다. 이러한 경우, 알고리즘의 성능을 향상시키기 위하여 제1 실험에서 사용된 텍스트 T 의 모든 q -그램 핑거프린트의 분포를 확인하고 그 결과를 이용하여 패턴을 추출한 실험이 진행될 수 있다.

[0053]

도 4는 제1 실험에서 사용한 ran과 dow에서 길이가 각각 3, 4이고, 시작 위치가 $i(0 \leq i \leq n-q)$ 인 q -그램 핑거프린트 $f(T[i..i+q-1])$ 의 분포를 나타낸 것이다. 핑거프린트 함수에 의하면, q 가 커질수록 가능한 핑거프린트의 범위는 증가한다. ran에 대해서는 모든 핑거프린트가 거의 동일하게 분포되지만, dow는 그렇지 않으며, 특히 핑거프린트가 $q!-1$ 인 경우 자주 발생되었다. 과거부터 지속적으로 주가가 상승해온 dow의 특성에 의한 것으로 판단된다. 이러한 이유로 제1 실험의 dow에서 패턴을 랜덤으로 추출하였지만 2차 q -그램 핑거프린트가 $q!-1$ 인 경우가 상대적으로 많았고, 이로 인하여 순위동형 검증 횟수를 줄인 효과가 감소하여 ran에 비해 수행시간이 크게 줄지 않았다.

[0054]

표 3은 제1 실험에서 사용된 dow를 패턴 P 에서 생성할 때, P 의 1, 2차 q -그램 핑거프린트가 $q!-1$ 이 되지 않도록 추출하여 실험한 OPQ와 OPSQ의 수행시간이다. 실험결과, 모든 경우에서 제1 실험의 OPSQ보다 제2 실험의 OPSQ가 더 빠르게 수행되었다. $q=3$ 일 때, OPSQ가 OPQ보다 평균 18% 정도 빠르게 수행되었고, $q \geq 4$ 일 때도 평균적으로 11% 정도 빠르게 수행되었다. 특히, $m=15, q=3$ 일 때는 OPSQ가 OPQ보다 약 23% 빠르게 수행되었다.

[0055]

표 3: T 에서 드물게 나타나는 부분문자열을 P 로 설정했을 때의 OPQ와 OPSQ의 수행시간

Input data	Algorithm	m	Execution time(second)				
			q = 3	q = 4	q = 5	q = 6	q = 7
dow	OPQ	5	1.058	1.494	-	-	-
	OPSQ		0.901	1.403	-	-	-
	OPQ	10	0.692	0.555	0.64	0.923	1.652
	OPSQ		0.579	0.505	0.575	0.865	1.508
	OPQ	15	0.671	0.331	0.365	0.467	0.734
	OPSQ		0.519	0.302	0.322	0.434	0.678

[0056]

[0057]

이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로

세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0058] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

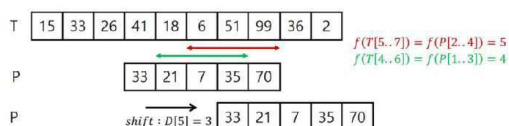
[0059] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

[0060] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

[0061] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

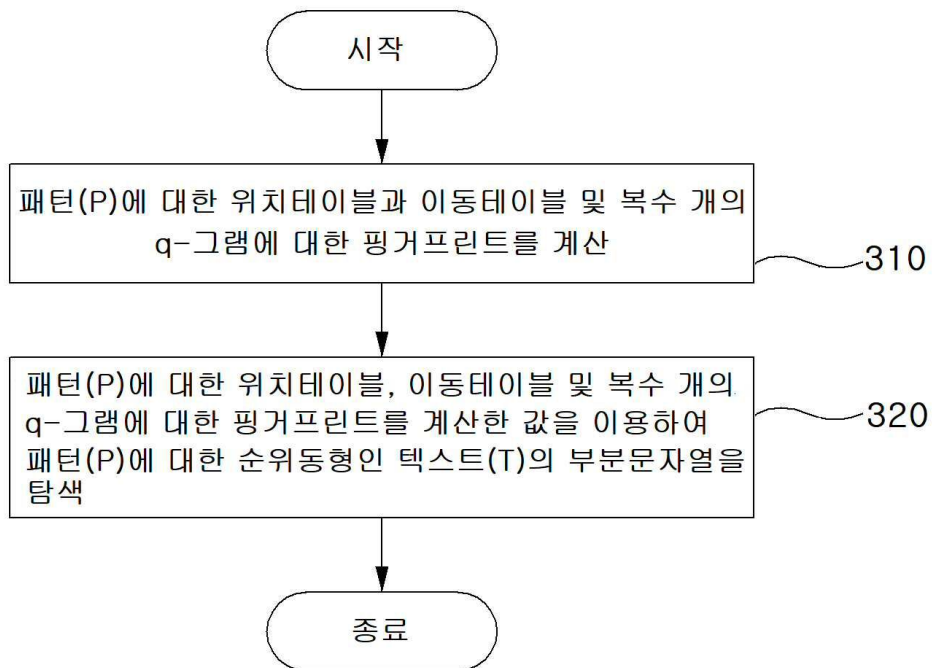
도면1



도면2



도면3



도면4

