(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0092511
(43) 공개일자 2024년06월24일

(51) 국제특허분류(Int. Cl.)
G06F 16/903 (2019.01) *G06F 18/213* (2023.01)
(52) CPC특허분류
G06F 16/90344 (2019.01)
G06F 18/213 (2023.01)
(21) 출원번호 10-2023-0003300
(22) 출원일자 2023년01월10일
심사청구일자 2023년01월10일
(30) 우선권주장
1020220174591 2022년12월14일 대한민국(KR)

(71) 출원인
인하대학교 산학협력단
인천광역시 미추홀구 인하로 100(용현동, 인하대학교)
(72) 발명자
심정섭
인천광역시 남동구 남동대로 860, 120동 601호(간석동, 간석래미안자이아파트)
김영호
인천광역시 미추홀구 인하로 100, 하이테크센터 1010호(용현동, 인하대학교)
(74) 대리인
양성보

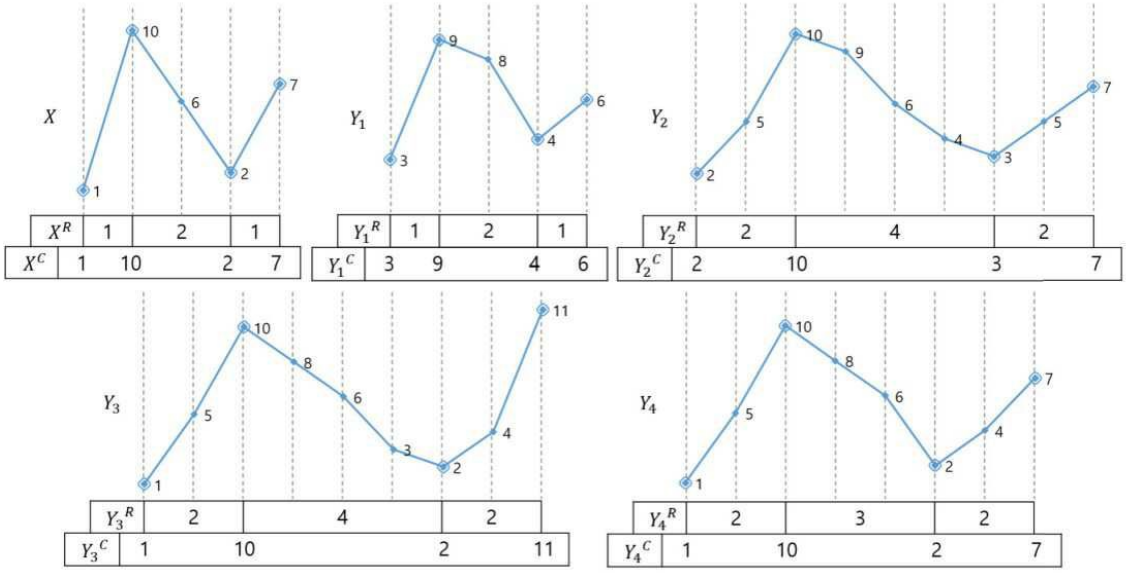
전체 청구항 수 : 총 5 항

(54) 발명의 명칭 **스케일링을 통한 순위 패턴 매칭 방법 및 시스템**

(57) 요 약

스케일링을 통한 순위 패턴 매칭 방법 및 시스템이 개시된다. 일 실시예에 따른 매칭 시스템에 의해 수행되는 스케일링을 통한 순위 패턴 매칭 방법은, 패턴의 문자열을 이용하여 패턴을 순위 동형으로 스케일링할 수 있는 텍스트의 위치 정보를 탐색하는 단계; 및 상기 탐색된 위치 정보에 기초하여 텍스트에 대해 상기 패턴이 스케일링된 순위 동형인지 여부를 확인하는 단계를 포함할 수 있다.

대 표 도 - 도3



이 발명을 지원한 국가연구개발사업

과제고유번호	1711174118
과제번호	00155915
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	인공지능융합연구센터지원사업(국고)
연구과제명	[Ezbaro][정부] 인공지능융합연구센터지원(3차년도)
기 여 율	1/1
과제수행기관명	인하대학교 산학협력단
연구기간	2022.01.01 ~ 2022.12.31

명 세 서

청구범위

청구항 1

매칭 시스템에 의해 수행되는 스케일링을 통한 순위 패턴 매칭 방법에 있어서,

패턴의 문자열을 이용하여 패턴을 순위 동형으로 스케일링할 수 있는 텍스트의 위치 정보를 탐색하는 단계; 및

상기 탐색된 텍스트의 위치 정보에 기초하여 텍스트에 대해 상기 패턴이 스케일링된 순위 동형인지 여부를 확인하는 단계

를 포함하는 스케일링을 통한 순위 패턴 매칭 방법.

청구항 2

제1항에 있어서,

상기 탐색하는 단계는,

커스프(cusp)의 상대적인 순서 정보와, 상기 커스프 사이의 문자열 길이의 스케일을 고려하여 상기 텍스트에서 상기 스케일링된 패턴이 발생하는 위치 정보를 탐색하는 단계

를 포함하는 스케일링을 통한 순위 패턴 매칭 방법.

청구항 3

제1항에 있어서,

상기 탐색하는 단계는,

상기 패턴의 중심 문자열만 고려하여 상기 패턴이 순위 동형으로 스케일링될 수 있는 텍스트의 후보 하위 문자열을 검색하는 단계

를 포함하는 스케일링을 통한 순위 패턴 매칭 방법.

청구항 4

제1항에 있어서,

상기 확인하는 단계는,

상기 패턴의 중심 문자열을 제외한 문자를 고려하여 상기 패턴이 상기 텍스트의 후보 하위 문자열의 각각에 대해 스케일링된 순위 동형인지 여부를 확인하는 단계

를 포함하는 스케일링을 통한 순위 패턴 매칭 방법.

청구항 5

스케일링을 통한 순위 패턴 매칭을 위한 매칭 시스템에 있어서,

패턴의 문자열을 이용하여 패턴을 순위 동형으로 스케일링할 수 있는 텍스트의 위치 정보를 탐색하는 위치 탐색부; 및

상기 탐색된 텍스트의 위치 정보에 기초하여 텍스트에 대해 상기 패턴이 스케일링된 순위 동형인지 여부를 확인하는 순위 동형 확인부

를 포함하는 매칭 시스템.

발명의 설명

기 술 분 야

[0001] 아래의 설명은 배율순위패턴매칭 기술에 관한 것이다.

배 경 기 술

[0003] 순위 패턴 매칭(ORDER-PRESERVING PATTERN MATCHING; OPPM)은 정확한 값 대신 문자의 상대 순서를 비교한다. 길이가 같은 두 문자열 X와 Y가 동일한 상대 순서를 가질 때, 순위 동형이라고 한다. 길이 n의 텍스트 T와 길이 m의 패턴 P가 주어지면 순위 패턴 매칭 문제는 P와 순위 동형인 T의 모든 하위 문자열을 찾는 것이다. 예를 들어, T=(4, 3, 8, 1, 2)와 P=(4, 6, 1)가 주어졌을 때, T(2, 3, 1)의 하위 문자열 T[1..3] =(3, 8, 1)의 개별 문자의 순서는 P의 문자의 순서와 같다. 순위 패턴 매칭(OPPM) 문제는 일반 알파벳의 경우 O(n+mlogm) 시간에, 선형 시간 정렬 가능 알파벳의 경우 O(n+m) 시간에 해결할 수 있다.

[0004] 최근에는 매칭 오류를 허용하는 대략적인 순위 패턴 매칭 문제가 연구되고 있다. 비특허문헌 1에 k-불일치로 순위 패턴 매칭 문제를 정의하고 O(n(loglogm+kloglogk) 시간 알고리즘이 제시된 바 있고, 비특허문헌 2에 k-불일치를 가진 순위 패턴 매칭 문제에 대한 O(nm[m/w]+logm)) 시간 알고리즘과 O(n[m/w]loglogw +mlogm) 시간 알고리즘이 제시된 바 있다(여기서, w는 컴퓨터 워드 크기). 또한, 고전적인 패턴 매칭에서 문자열의 스케일링 근사치가 연구된 바 있다. 텍스트 T와 패턴 P가 주어졌을 때, 스케일링의 패턴 매칭 문제는 T에서 스케일링된 P가 발생하는 모든 위치를 찾는 것이다. 예를 들어, T=aaaabbbbcccd와 P=abbc가 주어졌을 때, T[1..8]=aabbbbcc는 P가 2로 스케일링된 것과 일치한다.

[0005] 그러나 스케일 패턴을 고려한 순위 패턴 매칭 문제에 대한 연구는 아직 수행되지 않았다.

선행기술문헌

비특허문헌

[0006] (비특허문헌 0001) 비특허문헌 1: P. Gawrychowski, P. Uznanski, Order-preserving pattern matching with k mismatches, Theor. Comput. Sci. 638 (2016) 136-144, https://doi.org/10.1016/j.tcs.2015.08.022.

(비특허문헌 0002) 비특허문헌 2: T. Chhabra, E. Giaquinta, J. Tarhio, Filtration algorithms for approximate order-preserving matching, in: C.S. Iliopoulos, S.J. Puglisi, E. Yilmaz (Eds.), String Processing and Information Retrieval - 22nd International Symposium, SPIRE 2015, London, UK, September 1-4, 2015, Proceedings, in: Lecture Notes in Computer Science, vol. 9309, Springer, 2015, pp. 177-187.

(비특허문헌 0003) 비특허문헌 3: J. Kim, P. Eades, R. Fleischer, S. Hong, C.S. Iliopoulos, K. Park, S.J. Puglisi, T. Tokuyama, Order-preserving matching, Theor. Comput. Sci. 525 (2014) 68-79, https://doi.org/10.1016/j.tcs.2013.10.006.

발명의 내용

해결하려는 과제

[0007] 문자열 사이의 문자를 무시하고 문자열의 커스프(cusp)를 고려하여 스케일 순위 동형을 정의할 수 있다.

[0008] 스케일링이 있는 순위 패턴 매칭(OPPM) 문제에 대한 효율적인 O(n+mlogm) 시간 알고리즘을 제공할 수 있다.

과제의 해결 수단

[0010] 매칭 시스템에 의해 수행되는 스케일링을 통한 순위 패턴 매칭 방법은, 패턴의 문자열을 이용하여 패턴을 순위 동형으로 스케일링할 수 있는 텍스트의 위치 정보를 탐색하는 단계; 및 상기 탐색된 텍스트의 위치 정보에 기초하여 텍스트에 대해 상기 패턴이 스케일링된 순위 동형인지 여부를 확인하는 단계를 포함할 수 있다.

[0011] 상기 탐색하는 단계는, 커스프(cusp)의 상대적인 순서 정보와, 상기 커스프 사이의 문자열 길이의 스케일을 고

려하여 상기 텍스트에서 상기 스케일링된 패턴이 발생하는 위치 정보를 탐색하는 단계를 포함할 수 있다.

[0012] 상기 탐색하는 단계는, 상기 패턴의 중심 문자열만 고려하여 상기 패턴이 순위 동형으로 스케일링될 수 있는 텍스트의 후보 하위 문자열을 검색하는 단계를 포함할 수 있다.

[0013] 상기 확인하는 단계는, 상기 패턴의 중심 문자열을 제외한 문자를 고려하여 상기 패턴이 상기 텍스트의 후보 하위 문자열의 각각에 대해 스케일링된 순위 동형인지 여부를 확인하는 단계를 포함할 수 있다.

[0014] 스케일링을 통한 순위 패턴 매칭을 위한 매칭 시스템은, 패턴의 문자열을 이용하여 패턴을 순위 동형으로 스케일링할 수 있는 텍스트의 위치 정보를 탐색하는 위치 탐색부; 및 상기 탐색된 텍스트의 위치 정보에 기초하여 텍스트에 대해 상기 패턴이 스케일링된 순위 동형인지 여부를 확인하는 순위 동형 확인부를 포함할 수 있다.

발명의 효과

[0016] 문자열의 커스프를 고려하여 스케일 순위 동형을 정의할 수 있다.

[0017] 스케일링이 있는 순위 패턴 매칭 문제에 대해 효율적인 시간 알고리즘을 제공할 수 있다.

도면의 간단한 설명

[0019] 도 1은 일 실시예에 있어서, 매칭 시스템의 구성을 설명하기 위한 블록도이다.

도 2는 일 실시예에 있어서, 스케일링을 통한 순위 패턴 매칭 방법을 설명하기 위한 흐름도이다.

도 3은 일 실시예에 있어서, 스케일링된 순위 동형을 설명하기 위한 도면이다.

도 4는 일 실시예에 있어서, 스케일링을 통해 매칭된 순위 패턴을 설명하기 위한 예이다.

발명을 실시하기 위한 구체적인 내용

[0020] 이하, 실시예를 첨부한 도면을 참조하여 상세히 설명한다.

[0022] 텍스트 T와 패턴 P가 주어지면, 순위 패턴 매칭 문제는 패턴 P와 동일한 상대 순서를 갖는 텍스트 T의 모든 하위 문자열을 찾는 것이다. 최근에는 k-불일치를 가진 순위 패턴 매칭과 같은 오류를 허용하는 대략적인 순위 패턴 매칭이 연구되고 있다. 실시예에서는 커스프(cusp)의 상대적 순서와 상대적 문자열 길이의 스케일을 고려하여 대략적인 순위 패턴 매칭에 대한 새로운 기준인 스케일링을 이용한 순위 패턴 매칭을 정의하기로 한다. 또한 가장 잘 알려진 정확한 순위 패턴 매칭 알고리즘과 동일한 시간 범위인 O(n+mlogm) 시간의 스케일링으로 순위 패턴 매칭 문제를 해결하는 알고리즘을 제시하기로 한다.

[0023] 실시예에서 제안되는 스케일링을 통한 순위 패턴 매칭 동작의 이해를 돕기 위해 우선적으로 아래의 내용을 설명하기로 한다.

[0024] Σ가 두 문자를 일정한 시간 내에 비교할 수 있는 문자 집합을 나타낸다 하자. 문자열 S에 대해 |S|는 길이를 나타내고 S[i]는 S의 i번째(0≤i <|S|) 문자를 나타내도록 하자. 하위 문자열[substring] S[i] ··· S[j](0≤i, j <|S|)는 S[i..j] 및 S[i..j]i >j일 때는 빈 문자열이다. 두 문자열 X와 Y의 연결은 X ∘ Y로 표시된다. 편의상, 문자열의 모든 문자가 비특허문헌 3에서와 같이 구별된다고 가정하기로 한다.

[0025] 순위 동형과 가장 가까운 인접 표현을 공식적으로 정의한다. 길이 m의 두 문자열 X와 Y가 X[i]<X[j] ⇔ Y[i]<Y[j] (0≤i, j<m)를 만족하면, X와 Y는 X≈Y로 표시되는 순위 동형이다. 순위 동형은 가장 가까운 인접 표현을 사용하여 효율적으로 결정될 수 있다. X의 가장 가까운 인접 표현은 다음과 같이 위치 테이블 $LMax_X$와 $LMin_X$를 사용하여 정의된다.

$$LMax_X[i] = \begin{cases} j & \text{if } X[j] = \max\{X[k] : X[k] < X[i], \\ & \quad 0 \le k \le i-1\} \\ -1 & \text{if there is no such } j \end{cases}$$

[0026]

$$LMin_X[i] = \begin{cases} j & \text{if } X[j] = \min\{X[k] : X[k] > X[i], \\ & \quad 0 \le k \le i-1\} \\ -1 & \text{if there is no such } j \end{cases}$$

[0027]

[0028] 즉, $LMax_X[i]$는 X[0..i-1]의 X[i]보다 작은 문자 중에서 가장 큰 문자의 인덱스 j를 저장하고, $LMin_X[i]$는 X[0..i-1]의 X[i]보다 큰 문자 중에서 가장 작은 문자의 인덱스 j를 저장한다.

[0029] 표 1:

Location tables for $X = (33, 24, 43, 30, 93, 28, 13, 81, 58)$.

i	0	1	2	3	4	5	6	7	8
$X[i]$	33	24	43	30	93	28	13	81	58
$LMax_X[i]$	−1	−1	0	1	2	1	−1	2	2
$LMin_X[i]$	−1	0	−1	0	−1	3	1	4	7

[0030]

[0031] 표 1은 X=(33, 24, 43, 30, 93, 28, 13, 81, 58)에 대한 $LMax_X$와 $LMin_X$를 보여준다. $LMax_X$와 $LMin_X$는 정렬 알고리즘 또는 순서 통계 트리를 사용하여 O(mlogm) 시간으로 계산할 수 있다. X에 대한 위치 테이블이 주어졌을 때, X와 Y의 순위 동형은 모든 Y[i](0≤i<m)[2,3]에 대해 부등식 (1)을 확인함으로써 O(m)시간에 결정될 수 있다.

$$Y[LMax_X[i]] < Y[i] < Y[LMin_X[i]]. \qquad (1)$$

[0032]

[0033] 다음의 보조정리와 결과는 두 문자열의 순위 동형을 결정하는 몇 가지 속성을 보여준다.

[0034] 보조정리 1. 길이 m의 두 문자열 X와 Y에 대해, X와 Y는 각각 X와 Y로부터 같은 순열로 얻은 문자열이라고 하자. 그러면 $X \approx Y \Leftrightarrow X' \approx Y'$이다.

[0035] 증명. 그것은 순위 동형의 정의에 의해 명백하다.

[0036] 그렇다면, 보조정리1에서 다음과 같은 결과를 얻을 수 있다.

[0037] 결과 2. 길이 m의 두 문자열 X와 Y에 대해 $X[1..m-1]{\circ}X[0] \approx Y[1..m-1]Y[0] \Leftrightarrow X \approx Y$이다.

[0038] 결과 2에 의해 X 대신 $X[1..m-1]{\circ}X[0]$의 위치 테이블을 사용하여 X와 Y의 순위 동형을 결정할 수 있다.

[0039] 도 1은 일 실시예에 있어서, 매칭 시스템의 구성을 설명하기 위한 블록도이고, 도 2는 일 실시예에 있어서, 스케일링을 통한 순위 패턴 매칭 방법을 설명하기 위한 흐름도이다.

[0040] 매칭 시스템(100)의 프로세서는 위치 탐색부(110) 및 순위 동형 확인부(120)를 포함할 수 있다. 이러한 프로세서의 구성요소들은 매칭 시스템에 저장된 프로그램 코드가 제공하는 제어 명령에 따라 프로세서에 의해 수행되는 서로 다른 기능들(different functions)의 표현들일 수 있다. 프로세서 및 프로세서의 구성요소들은 도 2의 스케일링을 통한 순위 패턴 매칭 방법이 포함하는 단계들(210 내지 220)을 수행하도록 매칭 시스템을 제어할 수 있다. 이때, 프로세서 및 프로세서의 구성요소들은 메모리가 포함하는 운영체제의 코드와 적어도 하나의 프로그램의 코드에 따른 명령(instruction)을 실행하도록 구현될 수 있다.

[0041] 프로세서는 스케일링을 통한 순위 패턴 매칭 방법을 위한 프로그램의 파일에 저장된 프로그램 코드를 메모리에 로딩할 수 있다. 예를 들면, 매칭 시스템에서 프로그램이 실행되면, 프로세서는 운영체제의 제어에 따라 프로

그램의 파일로부터 프로그램 코드를 메모리에 로딩하도록 매칭 시스템을 제어할 수 있다. 이때, 프로세서는 위치 탐색부(110) 및 순위 동형 확인부(120) 각각은 메모리에 로딩된 프로그램 코드 중 대응하는 부분의 명령을 실행하여 이후 단계들(210 내지 220)을 실행하기 위한 프로세서의 서로 다른 기능적 표현들일 수 있다.

[0042] 단계(210)에서 위치 탐색부(110)는 패턴의 문자열을 이용하여 패턴을 순위 동형으로 스케일링할 수 있는 텍스트의 위치 정보를 탐색할 수 있다. 위치 탐색부(110)는 커스프(cusp)의 상대적인 순서 정보와, 커스프 사이의 문자열 길이의 스케일을 고려하여 텍스트에서 스케일링된 패턴이 발생하는 위치 정보를 탐색할 수 있다. 위치 탐색부(110)는 패턴의 중심 문자열만 고려하여 패턴이 순위 동형으로 스케일링될 수 있는 텍스트의 후보 하위 문자열을 검색할 수 있다.

[0043] 단계(220)에서 순위 동형 확인부(120)는 탐색된 텍스트의 위치 정보에 기초하여 텍스트에 대해 패턴이 스케일링된 순위 동형인지 여부를 확인할 수 있다. 순위 동형 확인부(120)는 패턴의 중심 문자열을 제외한 문자를 고려하여 패턴이 텍스트의 후보 하위 문자열의 각각에 대해 스케일링된 순위 동형인지 여부를 확인할 수 있다.

[0044] 보다 상세하게, 순위 패턴 매칭 동작에 대하여 설명하기로 한다. 이를 위해 우선적으로, 스케일 순위 동형에 대하여 설명하기로 한다.

[0045] 길이 l의 문자열 X에 대하여, (X[i-1]<X[i]^ X[i]>X[i+1]) 또는 (X[i-1]>X[i])^(X[i]<X[i+1])일 경우 X[i](1≤i≤l-2)를 커스프(cusp)라 한다. 설명의 단순성을 위해 X[0]와 X[l-1] 문자를 커스프로 간주한다. 도 3을 참고하면, 문자열 X = (1, 10, 6, 2, 7), Y_1 = (3, 9, 8, 4, 6), Y_2 = (2, 5, 10, 9, 6, 4, 3, 5, 7), Y_3 = (1, 5, 10, 8, 6, 3, 2, 4, 11), and Y_4 = (1, 5, 10, 8, 6, 2, 4, 7)일 때, X^R, X^C, Y_1^R, Y_1^C, Y_2^R, Y_2^C, Y_3^R, Y_3^C, Y_4^R, Y_4^C 이다. 도 3에서 X=(1, 10, 6, 2, 7)인 경우, 커스프는 1, 10, 2, 7이다. 또한, 인접한 커스프 사이의 간격을 런(run)이라 하며 런(run)의 길이를 런-길이(run-length)라고 한다. 예를 들어, X는 길이가 각각 1, 2, 1인 세 개의 런(run)으로 구성된다. X^C로 표시되는 X의 커스프 문자열을 순서대로 X의 모든 커스프의 연결로 정의한다. 또한 X^R로 표시되는 X의 런-길이(run-length) 문자열을 순서대로 X의 모든 런-길이(run-length)의 시퀀스로 정의한다. 즉, X=(1, 10, 6, 2, 7), X^C=(1, 10, 2, 7), X^R=(1, 2, 1)이다.

[0046] 정의 1. $X^C \approx Y^C$ 및 $X^R \approx Y^R$이면 길이가 같은 두 개의 문자열 X와 Y는 커스프 순위 동형이다.

[0047] 관측치 1. $\sum_{i=0}^{|X^R|-1} X^R[i] = |X| - 1$

[0048] 예를 들어, 도 3의 X와 Y_1은 커스프 순위 동형이다. X[2]와 Y_1[2]은 커스프가 아니기 때문에 무시한다.

[0049] 주로 다른 문자, 더 구체적으로, 커스프의 상대적 순서와 커스프 사이의 런-길이(run-length)를 무시하는 커스프에 초점을 맞추기로 한다. 예를 들어, 도 3의 X는 Y_1으로의 스케일링 순위 동형이다. X의 커스프와 Y_1의 커스프는 순위 동형이며 두 문자열의 런-길이(run-length)는 같다. 도 3의 문자열 X 또한 Y_2로의 스케일링 순위 동형이다. X의 커스프와 Y_2의 커스프는 순위 동형이며 Y_2의 각 런-길이(run-length)는 X의 해당 런-길이(run-length)의 두 배이다. 한편, X는 Y_3도 Y_4도 아닌 스케일링 순위 동형이다. Y_3의 커스프는 X의 커스프와 순위 동형이 아니며, Y_4의 런-길이(run-length)와 X의 런-길이(run-length)의 비율은 일정하지 않다.

[0050] 스케일링 순위 동형의 공식적인 정의는 다음과 같다.

[0051] 정의 2. 길이의 두 문자열 X와 길이 n(n≥m)의 Y가 주어졌을 때, 만약 양의 정수 $k \le \lfloor \frac{n-1}{m-1} \rfloor$에 대해 다음 조건을 만족한다면, X는 $X \cong^k Y$로 표시되는 Y와 k-스케일링 순위 동형이다(관측값 1에서 n = k(m-1) +1.).

[0052] 조건 1: 모든 0≤i<$|X^R|$에 대해 $|X^R|=|Y^R|$ 및 $k \times X^R[i]=Y^R[i]$이다.

[0053] 조건 2: $X^C \approx Y^C$이다.

[0054] 예를 들면, 도 3에서 $X \cong^1 Y_1$ 및 $X \cong^2 Y_2$이다. 한편, 조건 2가 충족되지 않기 때문에 X는 Y_3에 대한 순위 동형이 아니다($X^C \not\approx Y_3^C$). 또한, 조건 1이 충족되지 않기 때문에 X는 Y_4에 대한 순위 동형이 아니다(X^R=(1, 2, 1) 및 Y_4^R=(2, 3, 2)).

[0055] 문제 1. 스케일링과 관련된 순위 패턴 매칭 문제.

[0056] 입력: 길이가 n인 문자열 T와 길이가 m인 문자열 P(n≥m).

[0057] 출력: 어떤 정수 $k\ (1 \le k \le \lfloor \frac{n-1}{m-1} \rfloor)$에 대해 $P \cong^k T[i..i+k(m-1)]\ \ (0 \le i \le n-m)$이 되도록 하는 T의 모든 위치 i

[0058] 다음으로 스케일링의 순위 패턴 매칭 문제에 대한 알고리즘을 설명하기로 한다. O(n+mlogm) 시간 내 스케일링으로 순위 패턴 매칭 문제를 해결하는 알고리즘이 제공될 수 있다. $|T^R|$=n', $|P^R|$=m'이라 하자. 그러면 $|T^C|$=n'+1, $|P^C|$=m'+1이다. T(0≤j≤n'-1)의 j번째 런(run)은 T^C[j] 문자로 시작한다. 또한 T^C[j] = T[j']이며, 여기서 $j' = \sum_{q=0}^{j-1} T^R[q]$이다. m' ≤ 3인 경우 중요하지 않기 때문에, m'≥4라고 가정하기로 한다.

[0059] 패턴 P와 텍스트 T의 몫(quotient) 문자열을 정의할 수 있다. 패턴 P의 몫 문자열 P^Q는 길이가 m'-1의 문자열로, P의 인접 런(run) 길이 사이의 비율, 즉 P^R에서 인접 문자 사이의 비율을 나타낸다.

[0060] $$P^Q = \left(\frac{P^R[0]}{P^R[1]}, \frac{P^R[1]}{P^R[2]}, \ldots, \frac{P^R[m'-2]}{P^R[m'-1]} \right)$$

[0061] 텍스트 T의 몫(quotient) 문자열도 유사하게 정의될 수 있다. 만약 문자열이 어떤 정수 k(k ≥ 1)에 의해 스케일링 된다면, 문자열의 인접 런(run) 길이 간의 비율은 k에 관계없이 유지된다. 예를 들어, X^R=(1, 2, 1, 2), Y^R=(2, 4, 2, 4)인 경우, 두 문자열 X와 Y에 대해 $X \cong^2 Y$라고 가정하기로 한다. 그러면 X와 Y의 몫(quotient) 문자열은 각각 (1/2, 2/1, 1/2)과 (2/4, 4/2, 2/4)로 동일하다.

[0062] 실시예에서 제안된 알고리즘은 T^R, P^R, T^C, P^C, T^Q, P^Q를 사용한다. 텍스트 T와 패턴 P를 스캔하여 T^R, P^R, T^C, P^C를 구성할 수 있고, 각각 T^R과 P^R을 스캔하여 T^Q와 P^Q를 구성할 수 있다. 이 모든 문자열은 O(n+m) 공간을 사용하여 O(n+m) 시간에 구성될 수 있다.

[0063] 스케일링을 이용한 순위 패턴 매칭 문제에 대한 실시예에서 제안된 알고리즘은 두 단계로 구성된다. 패턴 P의 중심 문자열을 P에서 첫 번째 P^R[0] 문자와 마지막 P^R[m'-1]문자를 제외한 패턴 P의 하위 문자열이라 하자. 1단계에서는 패턴 P의 중심 문자열만 고려하여 패턴 P가 순위 동형으로 스케일링될 수 있는 텍스트 T의 후보 하위 문자열 w를 검색한다. 2단계에서는 패턴 P의 중심 문자열을 제외한 문자를 고려하여 패턴 P가 각 하위 문자열 w에 대해 스케일링된 순위 동형인지 확인한다. 중심 문자열과 다른 문자를 따로 취급하는 이유는 패턴 P의 중심 문자열의 커스프가 텍스트 T의 커스프에 해당하는 반면, 패턴 P의 처음과 마지막 문자는 텍스트 T의 커스프에 해당하지 않을 수 있기 때문이다. 예를 들어, 도 4의 T[8..20]에서 패턴 P가 스케일링된 순위 동형이라고 가정하자. 첫 번째 문자 P[0]와 마지막 문자 P[6]는 각각 T[8]와 T[20]에 해당하며, 둘 다 텍스트 T의 커스프가 아니다.

[0064] 보다 상세하게는, 1단계에서, 패턴 P의 중심 문자열만을 고려하여 패턴 P를 순위 동형으로 스케일링될 수 있는 텍스트 T의 모든 후보 하위 문자열 w를 탐색한다. 정의 2의 각 조건에 대해 조건을 만족하는 모든 위치를 검색한다.

[0065] 조건 1을 충족하는 T^R 내 모든 위치의 J_1 집합을 찾는다. P^Q[1..m'-3](P의 중심 문자열의 quotient 문자열)이 정확히 T^Q의 j_1에서 발생하고 스케일 비율 $k=T^R[j_1]/P^R[1]$이 정수인 모든 위치 $j_1 \in J_1$을 찾는다.

[0066] 각 발생마다 다를 수 있는 스케일 비율 k를 알 수 없다는 점에 유의한다. 따라서, j_1을 검색하기 위해 T^R과 P^R을 사용하는 대신, 몫 문자열을 사용할 수 있다. 왜냐하면, 앞에서 언급한 것처럼 문자열이 정수 k로 스케일링되면, k와 상관없이 인접 런-길이(run-length) 간의 비율이 보존되기 때문이다. 즉, 정수 k에 대하여 $T^R[j_1+\ell-1]=kP^R[\ell]\ (1\le\ell\le m'-2)$ 이면, $T^Q[j_1..j_1+m'-4]=P^Q[1..m'-3]$ 이다.

[0067] m'=5일 때 도 4의 예에서, P^Q[1..2]=(1/2, 2/1)의 발생은 T^Q[1..2](k=1)와 T^Q[7..8](k=2)이며, 즉 J_1={1, 7}이다.

[0068] 조건 2를 충족하는 T^C의 모든 위치의 J2 집합을 찾는다. P^c[1..m'-1]과 순위 동형인 T^C[j2..j2+m'-2]와 같은 모든 위치 $j_2\in J_2$를 탐색한다(P의 중심 문자열의 커스프 문자열).

[0069] 도 4의 예에서 T^C[3.6] 및 T^C[7.10]는 P^C[1.4]와 순위 동형이며, 즉 J2={3,7}이다.

[0070] 각 $j\in J_1\cap J_2$에 대해, 후보 하위 문자열 w = T[s..e]의 시작 위치와 끝 위치 e를 결정한다. 분명히, k가 스케일 비율 T^R[j]/P^R[1]일 때 e = s+k(m-1)이다. j'이 T^C[j]에 해당하는 T의 위치라 하자. 그렇다면 앞서 설명한 바와 같이 $j'=\sum_{q=0}^{j-1}T^R[q]$ 이다. 후보 하위 문자열 w의 첫 번째 런(run) 대 패턴 P의 스케일 비율도 k여야 하므로 s = j'-kP^R[0]이다. 도 4의 예에서, j = 7일 때, k = T^R[7]/P^R[1] = 2, $s=\sum_{q=0}^{6}T^R[q]-2P^R[0]$ = 10 -2 = 8이고, e = 8+2 X 6 = 20이다. 즉, T[8..20]이 후보이다.

[0071] 2단계에서, 패턴 P의 첫 번째 런(run)과 마지막 런(run)을 고려하여 1단계에서 발견된 각 후보 하위 문자열 w = T[s..e]에 대해 패턴 P가 스케일링된 순위 동형인지 여부를 결정한다.

[0072] w^R 및 P^R에 대해 조건 1이 충족되는지 확인한다.

[0073] $T^R[j-1]\ge kP^R[0]$ 및 $T^R[j+m'-2]\ge kP^R[m'-1]$ 여부를 확인한다.

[0074] 조건 1을 확인하기 위해서 $|w^R|=|P^R|$과 $w^R[i]=kP^R[i]\ (0\le i<m')$을 확인해야 한다. 이미 조건 1이 1단계에 의해 P^R[1..m'-2]을 만족한다는 것을 알고 있기 때문이다. 나머지 점검할 부분은 P^R[0]와 P^R[m'-1]이다. P^R[0]에 해당하는 후보 하위 문자열 w은 w[0..r] = T[s..s+r]이다. 여기서 r=kP^R[0]이다. T[s+r]에서 끝나는 런(run)의 길이는 T^R[j-1]이므로, T^R[j-1]≥r이면 T[s..s+r]와 w^R[0] = kP^R[0]에 하나의 런(run)만 존재한다. 마지막 실행에 대해서도 유사하게 확인함으로써, 조건 1이 충족되는지 확인할 수 있다.

[0075] 도 4의 예에서, T^R[6](=3) ≥kP^R[0](=2) 및 T^R[10](=3) ≥ kP^R[4](=2)이므로, 후보 T[8..20](j=7, k=2, m'=5)는 조건 1을 만족한다.

[0076] w^C 및 P^C에 대해 조건 2가 충족하는지 확인한다.

[0077] $P^C[1..m'-1]\circ P^C[m']\circ P^C[0]$의 위치 테이블을 사용하여 TC [j..j+m'-2]T [e]T[s]에서 마지막 두 문자에 대한 부등식 (1)을 확인한다.

[0078] 조건 1을 만족하므로 $w^C=T[s]\circ T^C[j..j+m'-2]\circ T[e]$이다. 결과 2에 의해 $T^C[j..j+m'-2]\circ T[e]\circ T[s]\approx P^C[1..m'-1]\circ P^C[m']\circ P^C[0]$ 을 체크함으로써 조건 2 $(w^C\approx P^C)$를 확인할 수 있다. 1단계에 의해 T^C[j..j+m'-2]≈P^C이므로 $P^C[1..m'-1]\circ P^C[m']\circ P^C[0]$의 위치 테이블을 이용하여 나머지 문자 T[e]와 T[s]에 대한 부등식 (1)을 확인해야 한다.

[0079] 도 4의 예에서, 조건 2는 TC[j..j+m'-2]T[e]T[s] = (41, 17, 35, 24, 29, 34) 및 $P^C[1..m'-1]P^C[m]P^C[0]$ = (40, 23, 37, 28, 30, 35)]의 마지막 두 문자에 대해 부등식 (1)이 만족되므로, T[8..20], P 후보에 대해 만족된다.

[0080] 도 4의 예에서, T[8..20]와 P는 k=2, $P \overset{\sim}{=}^2$ T[8..20]에 대해 양 조건을 만족하기 때문이다.

[0081] 다음으로, 알고리즘의 실행 시간이 분석될 수 있다. 먼저 1단계를 고려해 보자. 집합 J_1은 알고리즘과 일치하는 패턴을 사용하여 O(n+m) 시간으로 계산될 수 있다. 또한, 집합 J_2는 OPPM 알고리즘을 사용하여 O(n +mlogm) 시간으로 계산할 수 있다. 모든 후보 w의 시작 위치를 계산할 때, $j' = \sum_{q=0}^{j-1} T^R[q]$ 계산은 증가 순서로 모든 위치 j를 처리함으로써 총 O(n) 시간이 소요된다. 따라서 1단계에서는 총 O(n+mlogm) 시간이 필요하다. 2단계에서 $P^C[1..m'] \circ P^C[0]$ 의 위치 테이블을 O(mlogm) 시간으로 계산한 후, 각 후보 하위 문자열 w는 O(1) 시간에 조건 1과 2를 확인할 수 있다. 최대 n개의 후보가 있으므로 2단계에서는 총 O(n+mlogm) 시간이 필요하다. 따라서 실시예에서 제안된 알고리즘은 O(n+mlogm) 시간이 소요된다. 또한, P^C의 문자를 선형 시간으로 정렬할 수 있다면, O(m) 시간에서 1단계와 2단계의 조건 2를 확인하는 데 사용되는 위치 테이블을 계산할 수 있다. 따라서, 다음과 같은 정리 3을 도출할 수 있다.

[0082] 정리 3. T(|T|=n)와 P(|P|=m)가 주어지면 스케일링과 관련된 순위 패턴 매칭 문제는 O(n+mlogm) 시간 내에 해결될 수 있다. P^C의 문자를 선형 시간으로 정렬할 수 있다면 O(n+m) 시간으로 문제를 해결할 수 있다.

[0083] 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, 프로세서, 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.

[0084] 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.

[0085] 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.

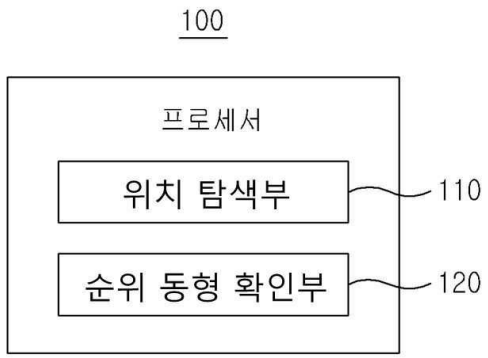
[0086] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다

른 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.

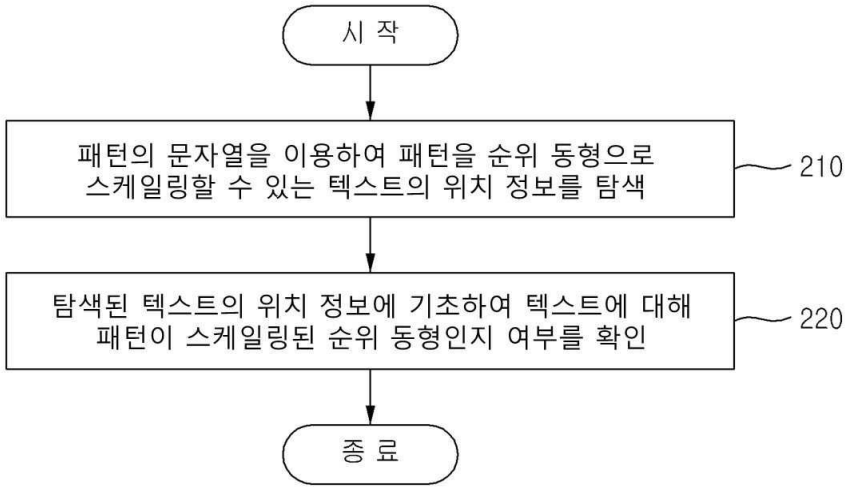
[0087] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

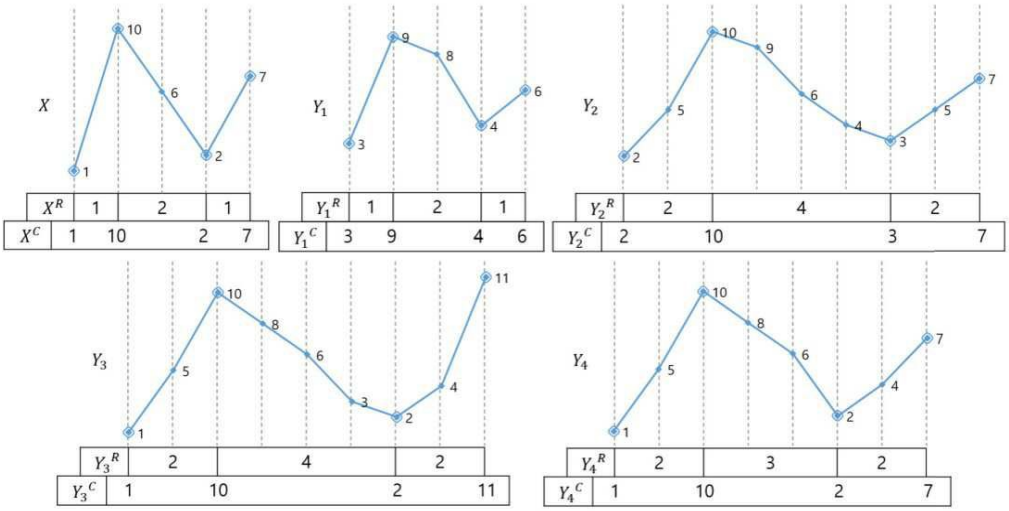
도면1



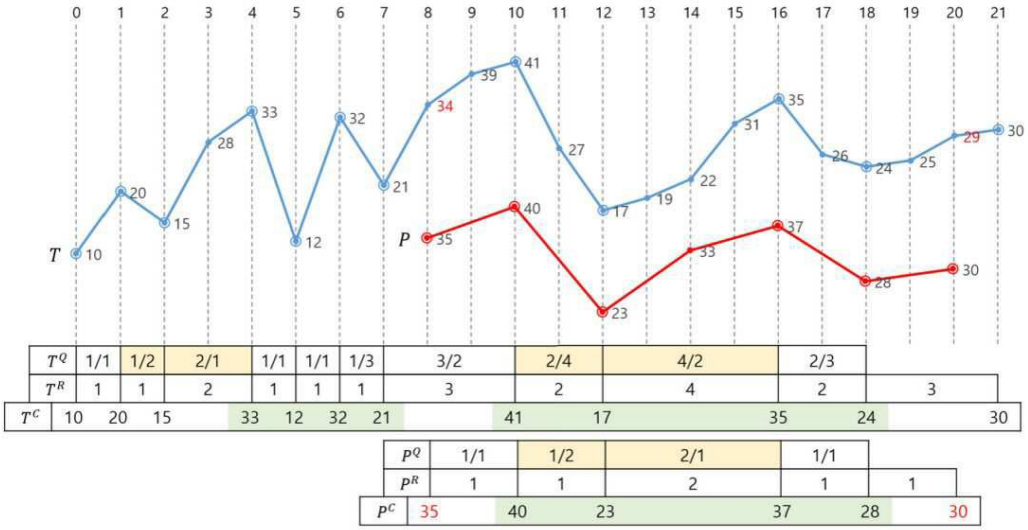
도면2



도면3


X
Y1
Y2
Y3
Y4
X^R 1 2 1
X^C 1 10 2 7
Y1^R 1 2 1
Y1^C 3 9 4 6
Y2^R 2 4 2
Y2^C 2 10 3 7
Y3^R 2 4 2
Y3^C 1 10 2 11
Y4^R 2 3 2
Y4^C 1 10 2 7

도면4


T
P
T^Q 1/1 1/2 2/1 1/1 1/1 1/3 3/2 2/4 4/2 2/3
T^R 1 1 2 1 1 1 3 2 4 2 3
T^C 10 20 15 33 12 32 21 41 17 35 24 30
P^Q 1/1 1/2 2/1 1/1
P^R 1 1 2 1 1
P^C 35 40 23 37 28 30