



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2011-0137220
(43) 공개일자 2011년12월22일

(51) Int. Cl.

G06F 17/30 (2006.01) G06F 17/26 (2006.01)
G06F 17/00 (2006.01)

(21) 출원번호 10-2010-0057317

(22) 출원일자 2010년06월16일

심사청구일자 2010년06월16일

(71) 출원인

인하대학교 산학협력단

인천 남구 용현동 253 인하대학교

(72) 발명자

심정섭

인천광역시 남동구 간석2동 금호어울림아파트 11
0동 1101호

(74) 대리인

김건우

전체 청구항 수 : 총 4 항

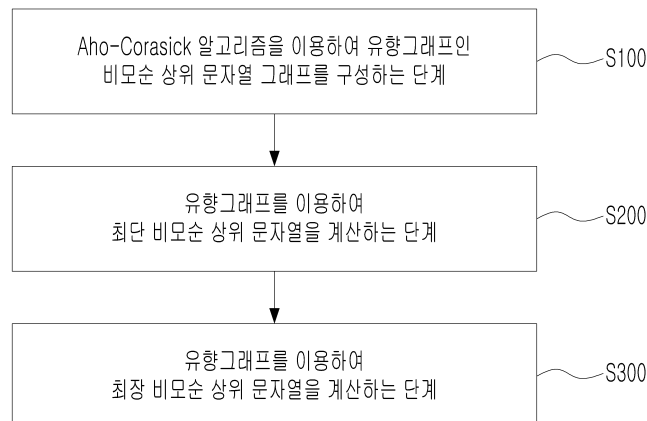
(54) 유향그래프를 이용한 비모순 상위 문자열 검색 방법

(57) 요약

본 발명은 유향그래프를 이용한 비모순상위 문자열 검색 방법에 관한 것으로서, 보다 구체적으로는, (1) Aho-Corasick 알고리즘에 기반하여 문자열 집합 N과 P를 표현하는 결정 유한 오토마톤 G_{AC} 를 생성하고 모든 음의 문자열에 대응하는 정점과 간선을 제거하여 유향그래프인 비모순 상위 문자열 그래프를 구성하는 단계, (2) 상기 단계 (1)에서 구성한 비모순 상위 문자열 그래프를 이용하여 모든 양의 문자열을 포함하는 최단 경로를 구하여 최단 비모순 상위 문자열을 계산하는 단계, 및 (3) 상기 단계 (1)에서 구성한 비모순 상위 문자열 그래프를 이용하여 모든 양의 문자열을 포함하는 최장 경로를 구하여 최장 비모순 상위 문자열을 계산하는 단계를 포함하는 것을 그 구성상의 특징으로 한다.

본 발명에서 제안하고 있는 유향그래프를 이용한 비모순 상위 문자열 검색 방법에 따르면, Aho-Corasick 알고리즘에 기반하여 문자열 집합 N과 P를 표현하는 새로운 그래프 G_{css} 를 구성하는 단계, G_{css} 그래프를 이용하여 최단 비모순 상위 문자열을 계산하는 단계, G_{css} 그래프를 이용하여 최장 비모순 상위 문자열을 계산하는 단계를 거침으로서, 주어진 문자열 집합에 대해 최단 비모순 상위 문자열과 최장 비모순 상위 문자열 검색을 능률적으로 수행할 수 있다.

대표도 - 도1



특허청구의 범위

청구항 1

비모순상위 문자열 검색 방법으로서,

- (1) Aho-Corasick 알고리즘에 기반하여 문자열 집합 N과 P를 표현하는 결정 유한 오토마톤 G_{AC} 를 생성하고 모든 음의 문자열에 대응하는 정점과 간선을 제거하여 유향그래프인 비모순 상위 문자열 그래프를 구성하는 단계;
- (2) 상기 단계 (1)에서 구성한 비모순 상위 문자열 그래프를 이용하여 모든 양의 문자열을 포함하는 최단 경로를 구하여 최단 비모순 상위 문자열을 계산하는 단계; 및
- (3) 상기 단계 (1)에서 구성한 비모순 상위 문자열 그래프를 이용하여 모든 양의 문자열을 포함하는 최장 경로를 구하여 최장 비모순 상위 문자열을 계산하는 단계를 포함하는 것을 특징으로 하는, 유향그래프를 이용한 비모순 상위 문자열 검색 방법.

청구항 2

제1항에 있어서, 상기 단계 (1)은,

Aho-Corasick 알고리즘을 이용하여 상기 문자열 집합 N과 P를 표현하는 결정 유한 오토마톤 G_{AC} 를 생성하는 단계; 및

상기 G_{AC} 그래프 내의 경로가 음의 문자열을 접미사로 포함하는 정점을 제거하고 제거된 정점과 인접한 모든 간선을 제거하여 유향그래프인 비모순 상위 문자열 그래프를 구성하는 단계를 포함하는 것을 특징으로 하는 유향그래프를 이용한 비모순 상위 문자열 검색 방법.

청구항 3

제1항에 있어서, 상기 단계 (2)는,

상기 G_{CSS} 그래프에서 각각의 양의 문자열을 포함하는 정점 집합들 중 하나의 정점씩을 모든 양의 문자열에 대해 선택하고, 선택한 정점들을 모두 순회하는 최단경로를 계산하는 단계; 및

상기 계산된 최단경로를 이용하여 최단 비모순 상위 문자열을 계산하는 단계를 포함하는 것을 특징으로 하는 유향그래프를 이용한 비모순 상위 문자열 검색 방법.

청구항 4

제1항에 있어서, 상기 단계 (3)은,

상기 G_{CSS} 그래프에서 각각의 양의 문자열을 포함하는 정점 집합들 중 하나의 정점씩을 모든 양의 문자열에 대해 선택하고, 선택한 정점들을 모두 순회하는 최장경로를 계산하는 단계; 및

상기 계산된 최장경로를 이용하여 최장 비모순 상위 문자열을 계산하는 단계를 포함하는 것을 특징으로 하는 유향그래프를 이용한 비모순 상위 문자열 검색 방법.

명세서

기술 분야

본 발명은 비모순 상위 문자열 검색 방법에 관한 것으로서, 보다 구체적으로는 비모순 상위 문자열 검색을 위해 유향 그래프를 생성하고 이를 이용하여 비모순 상위 문자열을 검색하는 방법에 관한 것이다.

[0001]

배경 기술

[0002] 문자열 포함 문제는 주어진 문자열들의 공통 부분서열 중 가장 긴 서열(Longest Common Subsequence)을 찾는 문제와 공통 상위서열 중 가장 짧은 서열(Shortest Common Subsequence)을 찾는 문제 등이 연구되어 왔다. 한편, 주어진 문자열에 대한 포함 불포함 문제가 압축 알고리즘, 분자 생물학, 컴퓨터 보안 등 다양한 분야에서 필요성이 대두되어 진행되고 있다.

[0003] 문자열 집합 N 과 P 가 주어졌을 때, 만약 문자열 a 가 N 내의 모든 음의 문자열 $x_i (1 \leq i \leq t)$ 을 포함하지 않고, P 내의 모든 양의 문자열 $y_j (1 \leq j \leq r)$ 을 포함하면, a 를 N 과 P 의 비모순 상위문자열(Consistent SuperString)이라 한다. 또한, 문자열 a 가 유한 길이이고, N 과 P 의 비모순 상위문자열들 중 가장 짧은 문자열이라면, a 를 N 과 P 의 최단 비모순 상위문자열(Shortest Consistent SuperString)이라 한다. 또한, 문자열 a 가 유한 길이이고, N 과 P 의 비모순 상위문자열들 중 가장 긴 문자열이라면, a 를 N 과 P 의 최장 비모순 상위문자열(Longest Consistent SuperString)이라 한다. 편의를 위하여, 상기 비모순 상위문자열과 최단 비모순 상위문자열과 최장 비모순 상위문자열을 각각 CSS와 SCSS와 LCSS라 나타낸다. 또한, N 의 모든 문자열의 개수를 t 로, P 의 모든 문자열의 개수를 r 로, N 의 모든 문자열의 길이의 합을 n 으로, P 의 모든 문자열의 길이의 합을 p 로 표기하기로 한다.

[0004]

[0005] 최단 비모순 상위 문자열 문제와 최장 비모순 상위 문자열 문제를 유향그래프를 구성하여 풀 수 있는 것이 알려져 있으나, 문자열 집합 N 과 P 의 제한 조건이 많고, 시간이 오래 걸리는 등의 능률적이지 못한 문제점이 있다.

발명의 내용

해결하려는 과제

[0006] 본 발명은 기존에 제안된 방법들의 상기와 같은 문제점들을 해결하기 위해 제안된 것으로서, Aho-Corasick 알고리즘에 기반하여 문자열 집합 N 과 P 를 표현하는 새로운 그래프 G_{css} 를 구성하는 단계, G_{css} 그래프를 이용하여 최단 비모순 상위 문자열을 계산하는 단계, G_{css} 그래프를 이용하여 최장 비모순 상위 문자열을 계산하는 단계를 거침으로서, 주어진 문자열 집합에 대해 최단 비모순 상위 문자열과 최장 비모순 상위 문자열 검색을 능률적으로 수행할 수 있는, 유향그래프를 이용한 비모순 상위 문자열 검색 방법을 제공하는 것을 그 목적으로 한다.

과제의 해결 수단

[0007] 상기한 목적을 달성하기 위한 본 발명의 특징에 따른 유향그래프를 이용한 비모순상위 문자열 검색 방법은,

[0008] (1) Aho-Corasick 알고리즘에 기반하여 문자열 집합 N 과 P 를 표현하는 결정 유한 오토마톤 G_{AC} 를 생성하고 모든 음의 문자열에 대응하는 정점과 간선을 제거하여 유향그래프인 비모순 상위 문자열 그래프를 구성하는 단계;

[0009] (2) 상기 단계 (1)에서 구성한 비모순 상위 문자열 그래프를 이용하여 모든 양의 문자열을 포함하는 최단 경로를 구하여 최단 비모순 상위 문자열을 계산하는 단계; 및

[0010] (3) 상기 단계 (1)에서 구성한 비모순 상위 문자열 그래프를 이용하여 모든 양의 문자열을 포함하는 최장 경로를 구하여 최장 비모순 상위 문자열을 계산하는 단계를 포함하는 것을 그 구성상의 특징으로 한다.

[0011] 바람직하게는, 상기 단계 (1)은,

[0012] Aho-Corasick 알고리즘을 이용하여 상기 문자열 집합 N 과 P 를 표현하는 결정 유한 오토마톤 G_{AC} 를 생성하는 단계; 및

[0013] 상기 G_{AC} 그래프 내의 경로가 음의 문자열을 접미사로 포함하는 정점을 제거하고 제거된 정점과 인접한 모든 간선을 제거하여 유향그래프인 비모순 상위 문자열 그래프를 구성하는 단계를 포함할 수 있다.

- [0014] 바람직하게는, 상기 단계 (2)는,
- [0015] 상기 Gcss 그래프에서 각각의 양의 문자열을 포함하는 정점 집합들 중 하나의 정점씩을 모든 양의 문자열에 대해 선택하고, 선택한 정점들을 모두 순회하는 최단경로를 계산하는 단계; 및
- [0016] 상기 계산된 최단경로를 이용하여 최단 비모순 상위 문자열을 계산하는 단계를 포함할 수 있다.

- [0017] 바람직하게는, 상기 단계 (3)은,
- [0018] 상기 Gcss 그래프에서 각각의 양의 문자열을 포함하는 정점 집합들 중 하나의 정점씩을 모든 양의 문자열에 대해 선택하고, 선택한 정점들을 모두 순회하는 최장경로를 계산하는 단계; 및
- [0019] 상기 계산된 최장경로를 이용하여 최장 비모순 상위 문자열을 계산하는 단계를 포함할 수 있다.

발명의 효과

- [0020] 본 발명에서 제안하고 있는 유향그래프를 이용한 비모순 상위 문자열 검색 방법에 따르면, Aho-Corasick 알고리즘에 기반하여 문자열 집합 N 과 P 를 표현하는 새로운 그래프 Gcss를 구성하는 단계, Gcss 그래프를 이용하여 최단 비모순 상위 문자열을 계산하는 단계, Gcss 그래프를 이용하여 최장 비모순 상위 문자열을 계산하는 단계를 거침으로서, 주어진 문자열 집합에 대해 최단 비모순 상위 문자열과 최장 비모순 상위 문자열 검색을 능률적으로 수행할 수 있다.

도면의 간단한 설명

- [0021] 도 1은 본 발명의 일실시예에 따른 유향그래프를 이용한 비모순 상위 문자열 검색 방법의 흐름을 도시한 도면.
- 도 2는 본 발명의 일실시예에 따른 유향그래프를 이용한 비모순 상위 문자열 검색 방법의 Aho-Corasick 알고리즘을 이용하여 유향그래프인 비모순 상위 문자열 그래프를 구성하는 단계의 세부적인 흐름을 도시한 도면.
- 도 3는 본 발명의 일실시예에 따른 Aho-Corasick 알고리즘을 이용하여 음의 문자열 집합 $N=\{aa, abba\}$ 과 양의 문자열 집합 $P=\{aba, bb\}$ 와 알파벳 집합 $\Sigma=\{a, b\}$ 인 AC machine을 나타내는 도면.
- 도 4는 본 발명의 일실시예에 따른 Aho-Corasick 알고리즘을 이용하여 음의 문자열 집합 $N=\{aa, abba\}$ 과 양의 문자열 집합 $P=\{aba, bb\}$ 와 알파벳 집합 $\Sigma=\{a, b\}$ 인 AC machine의 결정 유한 오토마톤 버전 G_{AC} 을 나타내는 도면.
- 도 5는 본 발명의 일실시예에 따른 새로운 그래프 Gcss를 나타내는 도면.
- 도 6은 본 발명의 일실시예에 따른 유향그래프를 이용한 비모순 상위 문자열 검색 방법의 유향그래프를 이용하여 최단 비모순 상위 문자열을 계산하는 단계의 세부적인 흐름을 도시한 도면.
- 도 7은 본 발명의 일실시예에 따른 유향그래프를 이용한 비모순 상위 문자열 검색 방법의 유향그래프를 이용하여 최장 비모순 상위 문자열을 계산하는 단계의 세부적인 흐름을 도시한 도면.

발명을 실시하기 위한 구체적인 내용

- [0022] 이하, 첨부된 도면을 참조하여 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자가 본 발명을 용이하게 실시할 수 있도록 바람직한 실시예를 상세히 설명한다. 다만, 본 발명의 바람직한 실시예를 상세하게 설명함에 있어, 관련된 공지 기능 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략한다. 또한, 유사한 기능 및 작용을 하는 부분에 대해서는 도면 전체에 걸쳐 동일한 부호를 사용한다.
- [0023] 덧붙여, 명세서 전체에서, 어떤 부분이 다른 부분과 ‘연결’되어 있다고 할 때, 이는 ‘직접적으로 연결’되어 있는 경우뿐만 아니라, 그 중간에 다른 소자를 사이에 두고 ‘간접적으로 연결’되어 있는 경우도 포함한다. 또한, 어떤 구성요소를 ‘포함’한다는 것은, 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있다는 것을 의미한다.

- [0024] 도 1은 본 발명의 일실시예에 따른 유향그래프를 이용한 비모순 상위 문자열 검색 방법의 흐름을 도시한 도면이다. 도 1에 도시된 바와 같이, 본 발명의 일실시예에 따른 유향그래프를 이용한 비모순 상위 문자열 검색 방법은, Aho-Corasick 알고리즘을 이용하여 유향그래프인 비모순 상위 문자열 그래프를 구성하는 단계(S100), 유향그래프를 이용하여 최단 비모순 상위 문자열을 계산하는 단계(S200), 유향그래프를 이용하여 최장 비모순 상위 문자열을 계산하는 단계(S300)를 포함하여 구성될 수 있다.
- [0025] 단계 S100에서는, Aho-Corasick 알고리즘에 기반하여 문자열 집합 N과 P를 표현하는 결정 유한 오토마톤 G_{AC} 를 생성하고 모든 음의 문자열에 대응하는 정점과 간선을 제거하여 유향그래프인 비모순 상위 문자열 그래프를 구성한다.
- [0026] 단계 S200에서는, 상기 단계 S100에서 구성한 비모순 상위 문자열 그래프를 이용하여 모든 양의 문자열을 포함하는 최단 경로를 구하여 최단 비모순 상위 문자열을 계산한다.
- [0027] 단계 S300에서는, 상기 단계 S100에서 구성한 비모순 상위 문자열 그래프를 이용하여 모든 양의 문자열을 포함하는 최장 경로를 구하여 최장 비모순 상위 문자열을 계산한다.
- [0028] 이하에서는, 도 2 내지 도 7을 참조하여 각 단계에 대한 세부적인 구성 및 흐름에 대해 상세히 설명하도록 한다.
- [0029]
- [0030] 도 2는 본 발명의 일실시예에 따른 유향그래프를 이용한 비모순 상위 문자열 검색 방법의 Aho-Corasick 알고리즘을 이용하여 유향그래프인 비모순 상위 문자열 그래프를 구성하는 단계의 세부적인 흐름을 도시한 도면이고, 도 3은 본 발명의 일실시예에 따른 Aho-Corasick 알고리즘을 이용하여 음의 문자열 집합 $N=\{aa, abba\}$ 과 양의 문자열 집합 $P=\{aba, bb\}$ 와 알파벳 집합 $\Sigma=\{a, b\}$ 인 AC machine을 나타내는 도면이며, 도 4는 본 발명의 일실시예에 따른 Aho-Corasick 알고리즘을 이용하여 음의 문자열 집합 $N=\{aa, abba\}$ 과 양의 문자열 집합 $P=\{aba, bb\}$ 와 알파벳 집합 $\Sigma=\{a, b\}$ 인 AC machine의 결정 유한 오토마톤 버전 G_{AC} 을 나타내는 도면이다.
- [0031] 본 발명에서는, T를 N과 P에 있는 모든 문자열의 접미사 집합이라고 표기한다. 예를 들어 $N=\{aa, abba\}$ 이고 $P=\{aba, bb\}$ 이라고 할 경우, $T=\{\lambda, a, aa, ab, abb, abba, aba, b, bb\}$ 이다. 상수 크기 알파벳 Σ 에 대해 N과 P가 주어지고, 정점의 집합 V와 간선의 집합 E를 가지고 있는 유향그래프 $G_{AC}=(V, E)$ 를 다음과 같이 정의하자. $a \in T$ 인 각 문자열 a에 관하여 정점 v를 정의하고 그 외의 정점은 정의하지 않는다. 이때, T에 있는 각 문자열과 V 안에 있는 각 정점들은 일대일 대응이다. A와 대응하는 정점들을 $ver(a)$ 라고 하고 정점 v에 대응하는 문자열을 $str(v)$ 라 한다. $str(ver(a))=a$ 임은 명백하다. 또한 $ver(\lambda)$ 를 특별한 정점 v_0 라고 한다. $a \in T$ 와 $a \sigma$ 에 대하여 함수 $\delta: T \times \Sigma \rightarrow T$ 를 정의한다. $\delta(a, \sigma)$ 는 $\beta \in T$ 인 $a \sigma$ 의 가장 긴 접미사 β 이다. 그리고 $\beta = \delta(a, \sigma)$ 이어야만 $ver(a)$ 에서 $ver(\beta)$ 까지 연결된 간선을 σ 로 표시한다. $a \in T$ 와 $\sigma \in \Sigma$ 인 각 문자열에서 오직 하나의 간선만 정의된다. 그러므로 각 정점으로부터 나가는 간선은 정확히 $|\Sigma|$ 개이다. 이때, G_{AC} 는 AC machine의 NUP에 대한 결정 유한 오토마톤 버전이다.
- [0032]
- [0033] a가 $\gamma \in T$ 인 γ 의 접미사라 하면 $Vs(a)$ 가 $ver(\gamma)$ 의 집합이라고 정의한다. $Vs(P)$ 는 모든 $y \in P$ 에 대하여 $Vs(y)$ 의 합집합으로 표기한다. 그리고 $Vs(N)$ 은 모든 $x \in N$ 에 대하여 $Vs(x)$ 의 합집합으로 표기한다. 이때, 모든 $i \neq j (1 \leq i, j \leq r)$ 에 대하여 $Vs(x_i) \cap Vs(y_j) = \emptyset$ 이다. 도 4는 $N=\{aa, abba\}$ 이고 $P=\{aba, bb\}$ 인 G_{AC} 를 나타낸다. $Vs(aba)=\{v_4\}$, $Vs(bb)=\{v_5, v_8\}$, $Vs(aa)=\{v_2\}$, $Vs(abba)=\{v_6\}$ 이고, $Vs(P)=\{v_4, v_5, v_8\}$, $Vs(N)=\{v_2, v_6\}$ 이다. 문자열

의 접두사의 수가 문자열의 길이와 같으므로 $|V|$ 의 개수는 $O(n+p)$ 이다. 또한 모든 정점은 최대한 상수 $|Σ|$ 개의 간선을 가지기 때문에 간선의 개수 $|E|$ 또한 $O(n+p)$ 이다.

[0034] G_{AC} 그래프 내의 경로는 정점의 순서 $A=\{u_1, u_2, \dots, u_k\}$ 로 나타낼 수 있다. 이때, 모든 $i = 1, 2, \dots, k-1$ 에 대하여 (u_i, u_{i+1}) 은 그래프 내의 간선을 의미한다. A 의 길이는 $|A|$ 로 표기하고, $|A|$ 는 A 에 있는 간선의 개수이다. $A_1=\{u_1, \dots, u_k\}$ 와 $A_2=\{u_k, \dots, u_1\}$ 인 두 개의 경로에 대하여 A_1 의 마지막 정점이 A_2 의 첫 번째 정점과 동일하다고 할 때, $A_1A_2=\{u_1, \dots, u_k, \dots, u_1\}$ 이 A_1 과 A_2 의 합쳐진 경로를 나타낸다고 표기한다. λ -path는 v_0 부터 시작하는 경로를 나타낸다고 표기한다. P -path가 $y \in P$ 에 대하여 $V_s(y)$ 의 최소한 한 개의 정점을 지나는 λ -path를 나타낸다고 표기한다. N -path는 $V_s(N)$ 에 있는 어떠한 정점도 지나지 않는 λ -path를 나타낸다고 표기한다.

[0035]

[0036] G_{AC} 에서 경로 $A=\{u_1, \dots, u_k\}$ 가 주어졌을 때, $1 \leq i \leq k-1$ 인 각 l_i 가 간선 (u_i, u_{i+1}) 의 라벨이라고 하면, $pstr(A)$ 가 문자열 $l_1 \dots l_{k-1}$ 을 나타낸다고 표기한다. 역으로 $1 \leq i \leq k-1$ 인 각 (u_i, u_{i+1}) 은 l_i 로 라벨된 간선이라 하면, 문자열 $\alpha = l_1 \dots l_{k-1}$ 이 주어졌을 때, $path(\alpha)$ 가 경로 $(u_1, \dots, u_k)$ 를 나타낸다고 표기한다. λ -path가 v_0 에서 시작하는 $path(\alpha)$ 라고 가정했을 때, 모든 $v \in V$ 에 대해 $\sigma \in \Sigma$ 인 σ 로 표기되는 간선이 존재하기 때문에 λ -path(α)와 마찬가지로 $path(\alpha)$ 또한 G_{AC} 에 항상 존재한다.

[0037] 도 5는 본 발명의 일실시예에 따른 새로운 그래프 G_{css} 를 나타내는 도면이다. 도 5에는 본 발명의 일실시예에 따른 G_{AC} 에서 음의 문자열에 대응하는 정점과 해당 정점과 인접한 간선을 제거하는 단계(S120)의 음의 문자열 집합 $N=\{aa, abba\}$ 과 양의 문자열 집합 $P=\{aba, bb\}$ 와 알파벳 집합 $\Sigma=\{a, b\}$ 인 G_{css} 그래프가 도시되었다.

[0038]

[0039] G_{AC} 그래프에서 음의 문자열에 대응하는 정점과 해당 정점과 인접한 간선을 제거하는 단계(S120)에서는, G_{AC} 그래프에서 $V' = V \setminus V_s(N)$ 과 $E' = E \setminus \{(v_a, v_b) \mid v_a \in V_s(N) \text{ or } v_b \in V_s(N)\}$ 인 유향그래프 G_{css} 그래프를 생성한다. G_{css} 그래프는 음의 문자열에 대응하는 정점과 인접한 간선을 모두 제거함으로써 G_{AC} 그래프로부터 만들어진다.

[0040]

[0041] 도 6은 본 발명의 일실시예에 따른 유향그래프를 이용한 비모순 상위 문자열 검색 방법의 유향그래프를 이용하여 최단 비모순 상위 문자열을 계산하는 단계의 세부적인 흐름을 도시한 도면이다. 도 6에 도시된 바와 같이, 본 발명의 일실시예에 따른 유향그래프를 이용한 비모순 상위 문자열 검색 방법의 유향그래프를 이용하여 최단 비모순 상위 문자열을 계산하는 단계는, 비모순 상위 문자열 그래프에서 가장 짧은 최단경로를 계산하는 단계(S210), 및 최단경로를 이용하여 최단 비모순 상위 문자열을 계산하는 단계(S220)을 포함하여 구현될 수 있다.

[0042] G_{css} 그래프 상의 P -path를 찾음으로써 N 을 위한 비모순 상위 문자열을 찾을 수 있음을 판단할 수 있다. $V_s(P)$ 에 있는 정점을 이용하여 P -path를 찾을 수 있다. $1 \leq i \leq r$ 에 대하여 각 $V_s(y_i)$ 로부터 하나의 정점을 선택하고 선택한 정점을 w_i 라고 표기한다. 이때, P -sequence는 $w_1, w_2, \dots, w_r$ 의 순열로 정의한다. $\pi = \langle u_1, u_2, \dots, u_r \rangle$ 가 G_{css} 그래프에서 P -sequence라고 표기한다. P -path A 가 π 의 정점들을 순서대로 모두 지났다면, 즉, A 가 v_0 부터 u_1 까지, u_1 부터 u_2 까지, ..., 그리고 u_{r-1} 부터 u_r 까지의 경로의 연속일 때, 우리는 P -path A 가 P -sequence π 를 포함한다고 한다. 만약 P -sequence π 를 포함하는 P -path가 존재한다면, π 를 유효한 P -sequence라고 하고, 그렇지 않다면 무효한 P -sequence라고 한다. 이때, 각 P -path에 대하여, P -path를 포함한 P -sequence가 적어도 한 개가 존재한다. 따라서 P -sequence를 이용하여 P -path를 구할 수 있다.

[0043] 단계 S210에서는, 비모순 상위 문자열 그래프에서 각각의 양의 문자열을 포함하는 정점 집합들 중 하나의 정점을 모든 양의 문자열에 대해 선택하고, 선택한 정점들을 모두 순회하는 최단경로를 계산한다. 각 P -sequence

$\pi = \langle u_1, u_2, \dots, u_r \rangle$ 에 대하여 v_0 부터 u_1 까지, u_1 부터 u_2 까지, ..., 그리고 u_{r-1} 부터 u_r 까지의 가장 짧은 경로들을 연속 시킴으로써 π 를 포함하는 가장 짧은 P-path를 구한다.

[0044] 단계 S220에서는, 계산된 최단경로를 이용하여 최단 비모순 상위 문자열을 계산한다. 전 단계에서 얻은 각 P-sequence π 에 따른 가장 짧은 P-path들 중 가장 짧은 P-path를 찾고, 해당 P-path의 경로 상에 있는 간선의 라벨을 연결시킴으로써 최단 비모순 상위 문자열을 계산한다.

[0045]

[0046] 가능한 P-sequence의 개수를 고려해보면, 각 $y_i (1 \leq i \leq r)$ 에 대하여 $|V_s(y_i)|$ 개의 정점들이 있기 때문에 가능한

P-sequence의 개수는 이고, 이것은 Gcss 그래프 성질에 의하여 $r! \left(\frac{n+r}{r} \right)^r$ 를 초과하지 않는다. 따라서 가능한 P-sequence의 개수는 최대 $r! \left(\frac{n+r}{r} \right)^r$ 이다. Gcss 그래프가 사이클이 없는 유향그래프(Directed Acyclic Graph; DAG)라면 DAG상에서의 경로 $L = (z_1, \dots, z_q)$ 를 고려해보면, DAG의 위상정렬 순서에서 $1 \leq i \leq q-1$ 인 z_i 는 z_{i+1} 전에 나타나야 한다. A가 유효한 P-sequence를 포함하는 P-path라고 하고, TP가 Gcss의 위상정렬 순서라고 하면, DAG의 성질에 의하여 A와 TP와 π 가 같은 순서로 나타나게 된다. 따라서 P-path를 찾기 위해서 TP와 같은 순서로 나타난 P-sequence에 대해서만 계산해주면 된다. 따라서 DAG일 때 가능한 P-sequence의 개수는 최대 $r! \left(\frac{n+r}{r} \right)^r$ 이다.

[0047]

[0048] Aho-Corasick 알고리즘을 이용하여 G_{AC} 그래프를 생성하는 시간은 패턴 문자열의 길이의 합에 대해 선형 시간이 걸리므로 $O(n+p)$ 이다. 또한 G_{AC} 의 간선의 개수가 $O(n+p)$ 이므로 $x \in N$ 인 $ver(x)$ 의 들어오고 나가는 간선을 제거하는 것 역시 $O(n+p)$ 이다. 따라서 Gcss 그래프의 생성 시간은 $O(n+p)$ 이다. 만약 P-sequence를 이용하여 Gcss 그래프에서 가장 짧은 P-path를 계산하는 단계(S210)에서 Dijkstra 알고리즘을 이용한다면 $O\left(\left(\frac{n+r}{r}\right)^r \times r(n+p) \log(n+p)\right)$ 시간이 걸린다. 또한 만약 Gcss 그래프가 DAG라면 가장 짧은 경로는

정점의 수에 대해 선형 시간에 계산할 수 있으므로 $O\left(\left(\frac{n+r}{r}\right)^r \times r(n+p)\right)$ 시간이 걸린다. 또한 Gcss 그래프가 DAG이고 $|N|$ 과 $|P|$ 가 상수에 상계되거나 N 의 어떤 문자열도 P 의 문자열의 부분문자열이 아니고, P 의 어떤 문자열도 N 의 문자열의 부분문자열이 아니라면 최단 비모순 상위 문자열은 $O(n+p)$ 시간에 찾을 수 있다.

[0049]

[0050] 도 7은 본 발명의 일실시예에 따른 유향그래프를 이용한 비모순 상위 문자열 검색 방법의 유향그래프를 이용하여 최장 비모순 상위 문자열을 계산하는 단계의 세부적인 흐름을 도시한 도면이다. 도 7에 도시된 바와 같이, 본 발명의 일실시예에 따른 유향그래프를 이용한 비모순 상위 문자열 검색 방법의 유향그래프를 이용하여 최장 비모순 상위 문자열을 계산하는 단계는, 비모순 상위 문자열 그래프에서 가장 긴 최장경로를 계산하는 단계(S310), 및 최장경로를 이용하여 최장 비모순 상위 문자열을 계산하는 단계(S320)을 포함하여 구현될 수 있다.

[0051] 단계 S310에서는, 상기 비모순상위 문자열 그래프에서 각각의 양의 문자열을 포함하는 정점 집합들 중 하나의 정점씩을 모든 양의 문자열에 대해 선택하고, 선택한 정점들을 모두 순회하는 최장경로를 계산한다. 본 단계는 최단 비모순 상위 문자열을 찾는 단계와 거의 유사하다. P-sequence에서 정점 간의 가장 짧은 경로를 찾는 대신에 가장 긴 경로를 찾는다. 일반적으로 가장 긴 경로를 찾는 것은 NP-hard 문제이다. 그러므로 Gcss 그래프는 사이클이 없는 그래프라고 가정한다.

[0052] 단계 S320에서는, 계산된 최장경로를 이용하여 최장 비모순 상위 문자열을 계산한다. 각 P-sequence $\pi = \langle u_1, u_2, \dots, u_r \rangle$ 에 대하여 v_0 부터 u_1 까지, u_1 부터 u_2 까지, ..., 그리고 u_{r-1} 부터 u_r 까지의 가장 긴 경로들을 연속시킴

으로써 π 를 포함하는 가장 긴 P-path를 구한다. P-path를 이용하여 최장 비모순 상위 문자열을 계산하는 단계 (S320)에서는, 전 단계에서 얻은 각 P-sequence π 에 따른 가장 긴 P-path들 중 가장 긴 P-path를 찾고, 해당 P-path의 경로 상에 있는 간선의 라벨을 연결시킴으로써 최장 비모순 상위 문자열을 계산한다.

[0053]

[0054] 최장 비모순 문자열은 Gcss 그래프가 DAG라면 $O((\frac{n+r}{r})^r \times r(n+p))$ 시간이 걸리고, 또한 Gcss 그래프가 DAG이고 |N|과 |P|가 상수에 상계되거나 N의 어떤 문자열도 P의 문자열의 부분문자열이 아니고, P의 어떤 문자열도 N의 문자열의 부분문자열이 아니라면 최장 비모순 상위 문자열은 $O(n+p)$ 시간에 찾을 수 있다.

[0055] 이상 설명한 본 발명은 본 발명이 속한 기술분야에서 통상의 지식을 가진 자에 의하여 다양한 변형이나 응용이 가능하며, 본 발명에 따른 기술적 사상의 범위는 아래의 특허청구범위에 의하여 정해져야 할 것이다.

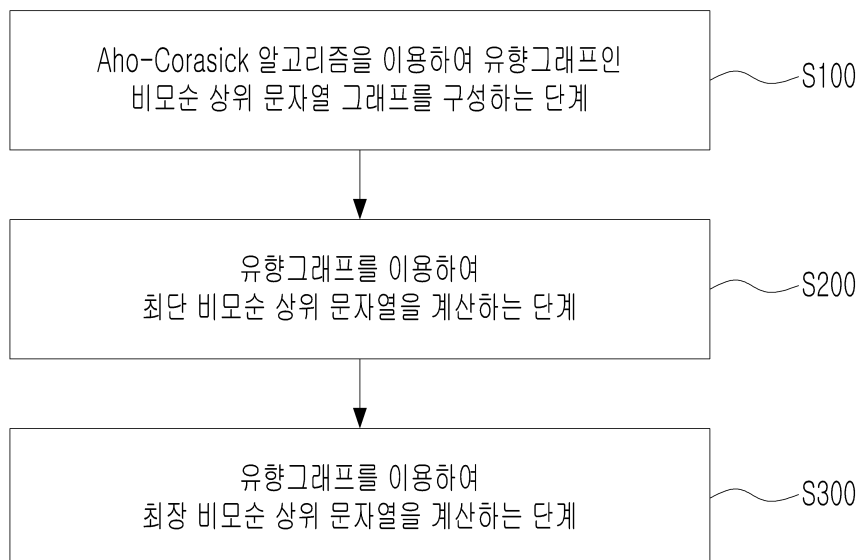
부호의 설명

[0056]

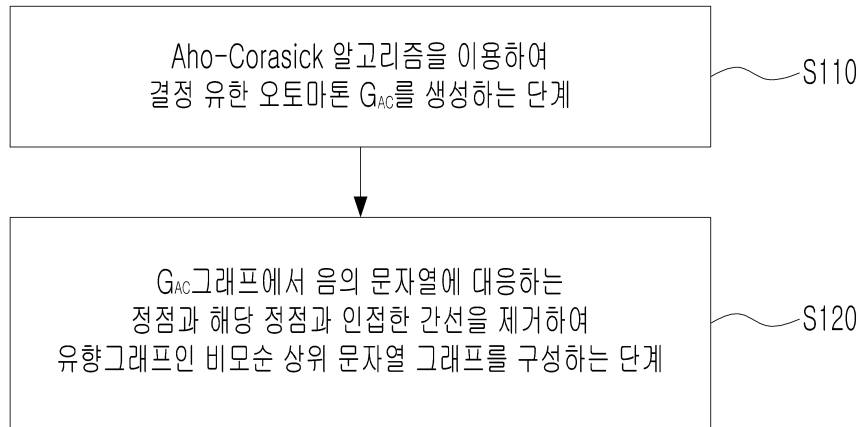
- S100: Aho-Corasick 알고리즘을 이용하여 유향그래프인 비모순 상위 문자열 그래프를 구성하는 단계
- S110: Aho-Corasick 알고리즘을 이용하여 결정 유한 오토마톤 c를 생성하는 단계
- S120: G_{AC} 그래프에서 음의 문자열에 대응하는 정점과 해당 정점과 인접한 간선을 제거하여 유향그래프인 비모순 상위 문자열 그래프를 구성하는 단계
- S200: 유향 그래프를 이용하여 최단 비모순 상위 문자열을 계산하는 단계
- S210: 비모순 상위 문자열 그래프에서 가장 짧은 최단경로를 계산하는 단계
- S220: 최단경로를 이용하여 최단 비모순 상위 문자열을 계산하는 단계
- S300: 유향 그래프를 이용하여 최장 비모순 상위 문자열을 계산하는 단계
- S310: 비모순 상위 문자열 그래프에서 가장 긴 최장경로를 계산하는 단계
- S320: 최장경로를 이용하여 최장 비모순 상위 문자열을 계산하는 단계

도면

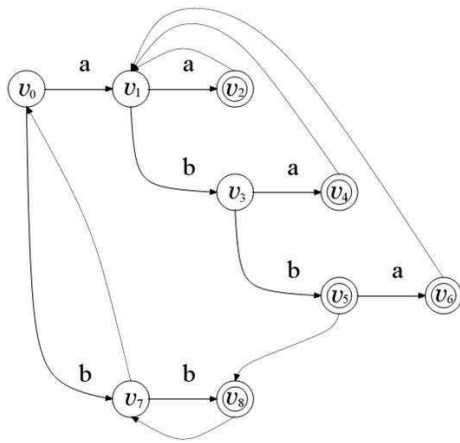
도면1



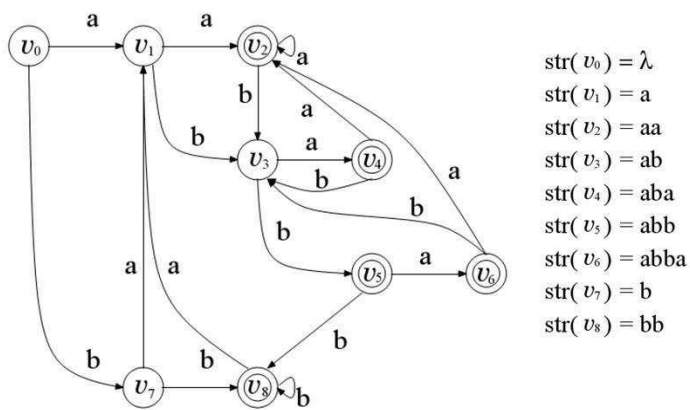
도면2



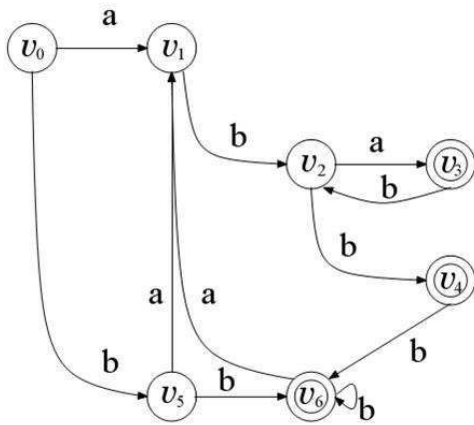
도면3



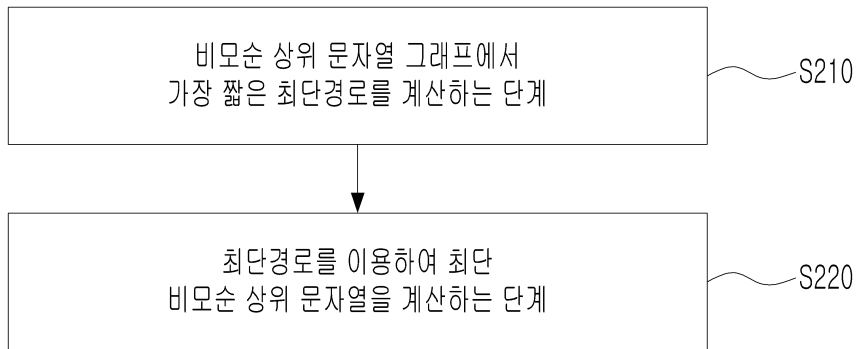
도면4



도면5



도면6



도면7

