



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2013-0095516
(43) 공개일자 2013년08월28일

(51) 국제특허분류(Int. Cl.)
G06F 17/00 (2006.01) G06F 17/30 (2006.01)
(21) 출원번호 10-2012-0017017
(22) 출원일자 2012년02월20일
심사청구일자 2012년02월20일

(71) 출원인
경희대학교 산학협력단
경기도 용인시 기흥구 덕영대로 1732, 국제캠퍼스
내 (서천동, 경희대학교)
(72) 발명자
이영구
경기도 용인시 기흥구 서천동 1번지 경희대학교
전자정보관 314호
박기성
경기도 용인시 기흥구 서천동 경희대학교 전자정
보관 349호
한용구
경기도 용인시 기흥구 서천동 경희대학교 전자정
보관 349호
(74) 대리인
서재승

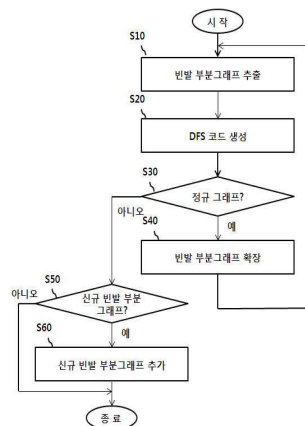
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 **빈발 부분그래프의 마이닝 방법**

(57) 요 약

본 발명은 빈발 부분그래프 마이닝 방법에 관한 것으로, 보다 구체적으로 빈발 부분그래프 마이닝을 반복적으로 수행하는 다양한 분석 기법에서 정규 그래프 탐색 트리의 DFS 코드를 비교하여 빈발 부분그래프가 최소 DFS 코드를 가지는 정규 그래프인지 판단함으로써 고비용의 정규화 연산을 줄일 수 있으며 빈발 부분그래프 마이닝의 수행 시간을 줄일 수 있는 빈발 부분그래프 마이닝 방법에 관한 것이다.

대표도 - 도2



특허청구의 범위

청구항 1

컴퓨터에서 그래프 데이터베이스에 저장되어 있는 그래프로부터 빈발 부분그래프를 마이닝하는 방법에 있어서,

- (a) 상기 그래프 데이터베이스로부터 패턴 확장 방식으로 확장된 빈발 부분그래프를 추출하는 단계;
- (b) 상기 추출한 빈발 부분그래프의 각 에지의 확장 순서로 상기 각 에지의 깊이 우선 탐색 코드(depth first search code, 이하 DFS 코드)를 생성하는 단계; 및
- (c) 상기 생성한 각 에지의 DFS 코드와 정규 그래프 탐색 트리의 정규 코드를 비교하여 상기 추출한 빈발 부분그래프가 최소 DFS 코드를 가지는 정규 그래프인지 판단하는 단계를 포함하는 빈발 부분그래프의 마이닝 방법.

청구항 2

제 1 항에 있어서, 상기 정규 그래프인지 판단하는 단계는

상기 생성한 각 에지의 DFS 코드가 상기 정규 그래프 탐색 트리에 존재하는지 판단하는 단계;

상기 판단 결과에 따라 상기 빈발 부분그래프의 각 에지의 확장 순서와 상기 정규 그래프 탐색 트리에서 상기 각 에지의 DFS 코드와 동일한 정규 코드의 저장 레벨 순서가 일치하는지 판단하는 단계; 및

상기 빈발 부분그래프의 각 에지의 확장 순서와 상기 각 에지의 DFS 코드와 동일한 정규 코드의 저장 레벨 순서가 일치하는 경우, 상기 추출한 빈발 부분그래프를 정규 그래프로 판단하는 단계를 포함하는 것을 특징으로 하는 빈발 부분그래프 마이닝 방법.

청구항 3

제 2 항에 있어서, 상기 추출한 빈발 부분 그래프가 최소 DFS 코드를 가지는 정규 그래프가 아닌 경우, 상기 추출한 빈발 부분 그래프를 삭제하는 것을 특징으로 하는 빈발 부분그래프의 마이닝 방법.

청구항 4

제 1 항 또는 제 2 항에 있어서, 상기 빈발 부분그래프의 마이닝 방법은

- (d) 상기 추출한 빈발 부분그래프가 정규 그래프인 경우, 상기 추출한 빈발 부분그래프를 패턴 확장 방식으로 에지를 확장하여 추가 빈발 부분그래프를 추출하는 단계를 더 포함하며,

상기 추가 빈발 부분그래프에 대해 상기 (b) 단계 및 (c) 단계를 반복하여 수행하는 것을 특징으로 하는 것을 특징으로 하는 빈발 부분그래프의 마이닝 방법.

청구항 5

제 4 항에 있어서, 상기 추출한 빈발 부분그래프의 DFS 코드는

상기 추출한 빈발 부분 그래프에서 확장 순서로 확장된, 상기 추출한 빈발 부분그래프를 구성하는 부모 에지와 자식 에지의 DFS 코드를 구비하며,

상기 부모 에지 또는 상기 자식 에지의 DFS 코드는 상기 부모 에지 및 자식 에지를 구성하는 2개의 버텍스, 상기 2개의 버텍스에 대한 레이블 및 상기 2개의 버텍스를 연결하는 에지 레이블을 구비하는 것을 특징으로 하는 빈발 부분그래프의 마이닝 방법.

청구항 6

제 4 항에 있어서, 상기 정규 그래프 탐색 트리는

상기 그래프 데이터베이스에서 검색된, 최소 DFS 코드를 가지는 빈발 부분그래프를 구성하는 부모 에지의 DFS 코드, 상기 부모 에지로부터 패턴 확장된 자식 에지의 DFS 코드 또는 상기 부모 에지로부터 상기 자식 에지로의 확장 포인터를 구비하는 것을 특징으로 하는 빈발 부분그래프 마이닝 방법.

청구항 7

제 6 항에 있어서, 상기 정규 그래프 탐색 트리에서 상기 부모 에지의 DFS 코드 또는 자식 에지의 DFS 코드는 에지의 버텍스 레이블과 에지 레이블로 구성되는 에지 코드의 발생 빈도에 서로 다른 데이터로 표현되어 있는 것을 특징으로 하는 빈발 부분그래프 마이닝 방법.

청구항 8

제 7 항에 있어서, 상기 에지 코드는 호프만 코드로 부호화되어 상기 정규 그래프 탐색 트리에 저장되어 있는 것을 특징으로 하는 빈발 부분그래프 마이닝 방법.

청구항 9

컴퓨터에서 그래프 데이터베이스에 저장되어 있는 그래프로부터 도약 탐색 기반으로 빈발 부분그래프를 마이닝 방법에 있어서,

- (a) 그래프 데이터베이스로부터 최소 지지도를 만족하는 빈발 부분그래프를 추출하는 단계;
- (b) 상기 추출한 빈발 부분그래프의 각 에지의 확장 순서로 깊이 우선 탐색 코드(depth first search code, 이하 DFS 코드)를 생성하는 단계; 및
- (c) 상기 생성한 DFS 코드와 정규 그래프 탐색 트리의 DFS 코드를 비교하여 상기 추출한 빈발 부분그래프가 최소 DFS 코드를 가지는 정규 그래프인지 판단하는 단계;
- (d) 상기 추출한 빈발 부분그래프가 정규화 그래프인 경우 상기 추출한 빈발 부분그래프의 에지를 확장하여 부분 그래프를 생성하며, 상기 생성한 부분 그래프로부터 상기 최소 지지도를 만족하는 추가 빈발 부분그래프를 추출하는 단계를 포함하는 빈발 부분그래프의 마이닝 방법.

청구항 10

제 9 항에 있어서, 상기 최소 지지도가 변경되는 경우 상기 그래프 데이터베이스에 상기 (a) 단계 내지 (d) 단계를 반복하여 상기 변경된 최소 지지도를 만족하는 빈발 부분그래프를 추출하는 것을 특징으로 하는 빈발 부분그래프의 마이닝 방법.

청구항 11

컴퓨터에서 그래프 데이터베이스에 저장되어 있는 그래프로부터 모델 기반 탐색 트리 방식으로 빈발 부분그래프를 마이닝하는 방법에 있어서,

- (a) 그래프 데이터베이스로부터 빈발 부분그래프를 추출하고 상기 추출한 빈발 부분그래프의 특성에 따라 상기 그래프 데이터베이스를 분류하는 단계;
- (b) 상기 분류 그래프 데이터베이스에서 분류 빈발 부분그래프를 추출하는 단계;
- (c) 상기 추출한 분류 빈발 부분그래프의 각 에지의 확장 순서로 깊이 우선 탐색 코드(depth first search code, 이하 DFS 코드)를 생성하는 단계;
- (d) 상기 생성한 DFS 코드와 정규 그래프 탐색 트리의 DFS 코드를 비교하여 상기 추출한 분류 빈발 부분그래프가 최소 DFS 코드를 가지는 정규 그래프인지 판단하는 단계;
- (e) 상기 추출한 분류 빈발 부분그래프가 정규화 그래프인 경우 상기 추출한 분류 빈발 부분그래프의 에지를 확장하여 부분 그래프를 생성하며, 상기 생성한 부분 그래프로부터 추가 분류 빈발 부분그래프를 추출하는 단계를 포함하는 빈발 부분그래프의 마이닝 방법.

청구항 12

제 11 항에 있어서, 상기 추출한 분류 빈발 부분그래프 또는 상기 추가 분류 빈발 그래프의 특성에 따라 상기 분류 그래프 데이터베이스를 재분류하며,

상기 재분류한 그래프 데이터베이스에 대해 상기 (b) 단계 내지 (e) 단계를 반복하는 것을 특징으로 하는 빈발 부분그래프의 마이닝 방법.

명세서

기술분야

- [0001] 본 발명은 빈발 부분그래프 마이닝 방법에 관한 것으로, 보다 구체적으로 빈발 부분그래프 마이닝을 반복적으로 수행하는 다양한 분석 기법에서 정규 그래프 탐색 트리의 DFS 코드를 비교하여 빈발 부분그래프가 최소 DFS 코드를 가지는 정규 그래프인지 판단함으로써 고비용의 정규화 연산을 줄일 수 있으며 빈발 부분그래프 마이닝의 수행 시간을 줄일 수 있는 빈발 부분그래프 마이닝 방법에 관한 것이다.

배경 기술

- [0002] 빈발 부분그래프(frequent subgraph)는 그래프 데이터베이스에서 최소 지지도 이상으로 출현하는 부분그래프로서, 빈발 부분그래프는 그래프의 고유한 특성을 나타내어 그래프의 분류, 군집화, 인덱싱 등에서 사용된다. 빈발 부분그래프 마이닝은 그래프 데이터베이스에서 모든 빈발 부분그래프를 찾는 기법으로, 복잡한 구조를 가지는 그래프들이 대용량으로 저장되어 있는 그래프 데이터베이스로부터 최소 지지도 이상으로 출현하는 모든 부분그래프를 검색하는 과정을 의미한다.
- [0003] 빈발 부분그래프를 효율적으로 마이닝하기 위하여 AGM, FSG, gSpan 등의 다양한 기법들이 연구되고 있다. 이 중에서 성능이 탁월한 것으로 알려진 gSpan이 널리 사용되고 있다. gSpan은 패턴 확장 방법으로 그래프 데이터베이스로부터 빈발 부분그래프를 마이닝하는 알고리즘으로, 깊이우선탐색(depth first search, 이하 DFS) 순서로 방문한 그래프의 에지들을 각 에지의 DFS 코드로 표현한다. 그래프는 탐색 시작 버텍스(vertex)에 따라 다양한 DFS 코드가 생성되는데, gSpan에서는 최소값을 가지는 DFS 코드를 정규 그래프로 정의하여 빈발 부분그래프가 최소 DFS코드와 다르게 확장된 경우 제거하여 많은 수의 중복 부분그래프들을 제거한다.
- [0004] gSpan의 전체 알고리즘은 다음과 같다. 우선, 그래프 데이터베이스 내의 모든 1-에지 빈발 부분그래프들을 찾는다. 찾은 1-에지 빈발 부분그래프를 최소 DFS코드로 표현하고 에지를 하나씩 확장하며 그래프 데이터베이스에 존재하는 그래프들의 빈발 부분그래프들을 추출한다. 에지를 확장한 빈발 부분그래프를 DFS코드로 표현하고, DFS 코드로 표현된 빈발 부분그래프 중 최소 DFS코드를 가지는 정규 그래프인 경우에만 확장하여 빠르게 빈발 부분그래프를 마이닝하며 최소 DFS 코드를 가지지 않는 나머지 빈발 부분그래프는 중복 그래프로 간주하여 제거한다.
- [0005] 빈발 부분그래프의 DFS 코드 중 최소 DFS 코드를 빈발 부분그래프의 정규 코드 형태로 사용하여 빈발 부분그래프를 유일하게 표현한다. 빈발 부분그래프를 정규 코드 형태로 표현하는 것을 정규화 연산(canonicalized operation)이라 정의하고 정규 코드 형태로 표현되는 빈발 부분그래프를 빈발 부분그래프의 정규 그래프라 언급한다. 정규화 연산을 위하여 빈발 부분그래프의 최소 DFS코드를 찾기 위한 과정은 다음과 같다. 빈발 부분그래프의 에지들을 DFS코드로 표현하고, 각 에지의 DFS코드 중 가장 작게 표현된 값을 탐색 시작 지점으로 설정한다. 탐색 시작 지점부터 깊이우선탐색(DFS) 순서에 따라 버텍스의 식별자를 순차적으로 부여한다. 만약 동일 버텍스가 여러 개의 에지를 가지면, 가장 작은 DFS코드로 표현되는 에지를 선택하여 탐색한다. 만약 가장 작게 표현되는 에지가 여러 개 존재할 경우, 하나의 에지를 선택하여 모두 탐색한 후, 나머지 에지들에 대해서도 재귀적으로 모든 에지를 탐색한다. 탐색하여 얻은 DFS코드 집합 중에서 가장 작게 표현되는 최소 DFS코드를 찾는다.
- [0006] 빈발 부분그래프의 정규화 연산은 이와 같이 복잡한 과정을 요구하는 고비용 연산이다. 특히, 빈발 부분그래프의 에지 수가 많거나 빈발 부분그래프에 동일한 레이블을 갖는 에지들이 같은 버텍스에 많이 연결되어 있는 경우 탐색 공간이 증가하여 수행시간이 크게 증가한다.
- [0007] 최근 다양한 그래프 연구 분야에서 빈발 부분그래프 마이닝을 반복적으로 수행하는 기법들이 제안되고 있다. 대표적인 예로서 그래프 분류에 유용한 특징들을 찾기 위하여 제안된 도약탐색(leap search) 기법과 모델 기반 탐색 트리(model based search tree) 기법이 있다. 도약 탐색 기법은 최소지지도를 다양하게 변화시키며 그래프 데이터베이스에서 찾은 빈발 부분그래프들 가운데 가장 분류의 성능이 높은 것들을 특징으로 선택한다. 모델 기반 탐색 트리 기법은 그래프 데이터베이스에서 찾은 빈발 부분그래프들 가운데 가장 분류 성능이 높은 빈발 부분그래프의 포함 여부를 기준으로 그래프 데이터베이스를 분할하는 과정을 반복 수행한다. 이와 같은 기법

들의 수행 시간은 통상적으로 긴 시간을 요구하는데, 이와 같은 기법의 수행 시간에서 반복적으로 수행되는 빈발 부분그래프의 정규화 연산 시간이 대부분을 차지한다.

[0008] 도 1은 정규화 연산의 고비용을 확인하기 위하여 모델기반 탐색 트리의 수행시간에서 정규화 연산이 차지하는 비율을 측정한 실험이다. 그래프 데이터베이스는 그래프로 표현된 항암 화합물 molt-4, MCF-7, YEAST을 사용하였다. 실험 결과 모델기반 탐색 트리 기법에서 정규화 연산의 수행 시간이 약 20~40%를 차지한다.

[0009] 따라서 빈발 부분그래프 마이닝을 반복적으로 수행하는 다양한 기법에서 정규화 연산 수행 시간을 줄임으로써 다양한 기법의 전체 수행 시간을 개선할 수 있다.

발명의 내용

해결하려는 과제

[0010] 본 발명은 위에서 언급한 종래 빈발 부분그래프 마이닝시 수행되는 정규화 연산이 가지는 문제점을 해결하기 위한 것으로, 본 발명이 이루고자 하는 목적은 빈발 부분그래프 마이닝을 반복하여 수행하는 다양한 분석 기법에서 동일한 빈발 부분그래프에 대해 고비용의 정규화 연산을 한 번만 수행하여 고비용의 정규화 연산 시간을 줄일 수 있는 빈발 부래프 마이닝 방법을 제공하는 것이다.

[0011] 본 발명이 이루고자 하는 다른 목적은 빈발 부분그래프의 확장 패턴 순서에 따라 정규 코드를 저장하는 정규 그래프 탐색 트리를 통해 확장된 빈발 부분그래프가 최소 DFS 코드를 가지는 정규 그래프인지 빠르게 판단할 수 있는 빈발 부분그래프 마이닝 기법을 제공하는 것이다.

과제의 해결 수단

[0012] 본 발명의 목적을 달성하기 위하여 본 발명에 따른 빈발 부분그래프 마이닝 방법은 그래프 데이터베이스로부터 빈발 부분그래프를 추출하는 단계와(이하 a단계), 추출한 빈발 부분그래프의 각 에지의 확장 순서로 깊이 우선 탐색 코드(depth first search code, 이하 DFS 코드)를 생성하는 단계와(이하 b단계), 생성한 DFS 코드와 정규 그래프 탐색 트리의 DFS 코드를 비교하여 추출한 빈발 부분그래프가 최소 DFS 코드를 가지는 정규 그래프인지 판단하는 단계와(이하 c단계), 추출한 빈발 부분그래프가 정규 그래프인 경우 추출한 빈발 부분그래프를 패턴 확장 방식으로 에지를 확장하여 추가 빈발 부분그래프를 추출하는 단계를 포함하며, 추가 빈발 부분그래프에 대해 (b) 단계 및 (c)단계를 반복하여 수행하는 것을 특징으로 한다.

[0013] 여기서 정규 그래프인지 판단하는 단계는 생성한 각 에지의 DFS 코드가 정규 그래프 탐색 트리에 존재하는지 판단하는 단계와, 판단 결과에 따라 빈발 부분그래프의 각 에지의 확장 순서와 정규 그래프 탐색 트리에서 각 에지의 DFS 코드와 동일한 정규 코드의 저장 레벨 순서가 일치하는지 판단하는 단계와, 빈발 부분그래프의 각 에지의 확장 순서와 각 에지의 DFS 코드와 동일한 정규 코드의 저장 레벨 순서가 일치하는 경우 추출한 빈발 부분그래프를 정규 그래프로 판단하는 단계를 포함한다.

[0014] 여기서 추출한 빈발 부분 그래프가 최소 DFS 코드를 가지는 정규 그래프가 아닌 경우 추출한 빈발 부분 그래프를 삭제한다.

[0015] 바람직하게 추출한 빈발 부분그래프의 DFS 코드는 추출한 빈발 부분 그래프에서 확장 순서로 확장된, 추출한 빈발 부분그래프를 구성하는 부모 에지와 자식 에지의 DFS 코드를 구비하며, 부모 에지 또는 자식 에지의 DFS 코드는 부모 에지 및 자식 에지를 구성하는 2개의 버텍스, 2개의 버텍스에 대한 레이블 및 2개의 버텍스를 연결하는 에지 레이블을 구비한다.

[0016] 한편, 정규 그래프 탐색 트리는 그래프 데이터베이스에서 검색된, 최소 DFS 코드를 가지는 빈발 부분그래프를 구성하는 부모 에지의 DFS 코드, 부모 에지로부터 패턴 확장된 자식 에지의 DFS 코드 또는 부모 에지로부터 자식 에지로의 확장 포인터를 구비한다.

[0017] 정규 그래프 탐색 트리에서 부모 에지의 DFS 코드 또는 자식 에지의 DFS 코드는 에지의 버텍스 레이블과 에지 레이블로 구성되는 에지 코드의 발생 빈도에 서로 다른 데이터로 표현되어 있다. 바람직하게, 에지 코드는 호프만 코드로 부호화되어 정규 그래프 탐색 트리에 저장되어 있다.

[0018] 본 발명에 따른 도약 탐색 기반의 빈발 부분그래프 마이닝 방법은 그래프 데이터베이스로부터 최소 지지도를 만

족하는 빈발 부분그래프를 추출하는 단계와(이하 a1단계), 추출한 빈발 부분그래프의 각 에지의 확장 순서로 깊이 우선탐색 코드(depth first search code, DFS 코드)를 생성하는 단계와(이하 b1단계), 생성한 DFS 코드와 정규 그래프 탐색 트리의 DFS 코드를 비교하여 추출한 빈발 부분그래프가 최소 DFS 코드를 가지는 정규 그래프인지 판단하는 단계와(이하 c1단계), 추출한 빈발 부분그래프가 정규화 그래프인 경우 추출한 빈발 부분그래프의 에지를 확장하여 부분 그래프를 생성하며 생성한 부분 그래프로부터 최소 지지도를 만족하는 추가 빈발 부분그래프를 추출하는 단계(이하 d1단계)를 포함하며, 최소 지지도가 변경되는 경우 그래프 데이터베이스에 (a1) 단계 내지 (d1) 단계를 반복하여 상기 변경된 최소 지지도를 만족하는 빈발 부분그래프를 추출하는 것을 특징으로 하는 빈발 부분그래프의 마이닝 방법.

[0019] 본 발명에 따른 모델 기반 탐색 트리에 기초하여 빈발 부분그래프를 마이닝하는 방법은 그래프 데이터베이스로부터 빈발 부분그래프를 추출하고 추출한 빈발 부분그래프의 특성에 따라 상기 그래프 데이터베이스를 분류하는 단계와(이하 a2단계), 분류 그래프 데이터베이스에서 분류 빈발 부분그래프를 추출하는 단계와(이하 b2단계), 추출한 분류 빈발 부분그래프의 각 에지의 확장 순서로 깊이 우선 탐색 코드를 생성하는 단계와(이하 c2단계), 생성한 DFS 코드와 정규 그래프 탐색 트리의 DFS 코드를 비교하여 추출한 분류 빈발 부분그래프가 최소 DFS 코드를 가지는 정규 그래프인지 판단하는 단계와(이하 d2단계), 추출한 분류 빈발 부분그래프가 정규화 그래프인 경우 추출한 분류 빈발 부분그래프의 에지를 확장하여 부분 그래프를 생성하며 생성한 부분 그래프로부터 추가 분류 빈발 부분그래프를 추출하는 단계와(이하 e2단계)를 포함하며, 추출한 분류 빈발 부분그래프 또는 추가 분류 빈발 그래프의 특성에 따라 분류 그래프 데이터베이스를 재분류하여 (b2) 단계 내지 (e2) 단계를 반복하는 것을 특징으로 한다..

발명의 효과

- [0020] 본 발명에 따른 빈발 부분그래프 마이닝 방법은 종래 기술과 비교하여 다음과 같은 다양한 효과를 가진다.
- [0021] 첫째, 본 발명에 따른 빈발 부분그래프의 마이닝 방법은 정규 그래프 탐색 트리를 통해 확장된 빈발 부분그래프의 DFS 코드가 정규 코드인지 판단함으로써, 확장된 빈발 부분그래프에 대한 별도의 정규화 연산없이 빠르고 용이하게 확장된 빈발 부분그래프가 정규 코드인지 판단할 수 있다.
- [0022] 둘째, 본 발명에 따른 빈발 부분그래프의 마이닝 방법은 고비용의 정규화 연산을 한 번만 수행하여 정규 그래프 탐색 트리를 생성하고 생성한 정규 그래프 탐색 트리를 반복되는 빈발 부분그래프의 마이닝에 사용함으로써, 고비용의 정규화 연산을 줄일 수 있으며 빈발 부분그래프 마이닝의 수행 시간을 줄일 수 있다.
- [0023] 셋째, 본 발명에 따른 빈발 부분그래프의 마이닝 방법은 빈발 부분그래프에서 빈발하는 정도에 따라 서로 다른 데이터량으로 빈발 에지를 저장 관리함으로써, 정규 그래프 탐색 트리를 유지 관리하는데 사용되는 데이터 저장 공간을 줄일 수 있으며 빠르게 빈발 부분그래프의 정규 코드를 확인할 수 있다.

도면의 간단한 설명

- [0024] 도 1은 정규화 연산의 고비용을 확인하기 위하여 모델기반 탐색 트리의 수행시간에서 정규화 연산이 차지하는 비율을 측정한 실험이다.
- 도 2는 본 발명에 따른 빈발 부분그래프의 마이닝 방법을 설명하기 위한 흐름도이다.
- 도 3은 그래프 데이터베이스에서 확장되는 빈발 부분그래프를 설명하기 위한 도면이다.
- 도 4와 도 5는 각각 제1 빈발 부분그래프와 제2 빈발 부분그래프의 DFS 코드를 나타낸 것이다.
- 도 6은 정규 그래프 탐색 트리의 일 예와 정규 그래프 탐색 트리에 기초하여 추출한 빈발 부분그래프가 최소 DFS 코드를 가지는 정규 그래프인지 판단하는 방법을 설명하기 위한 도면이다.
- 도 7은 본 발명에 따른 정규 그래프 탐색 트리의 다른 예를 설명하기 위한 도면이다.
- 도 8은 본 발명에 따른 빈발 부분그래프 마이닝 장치를 설명하는 기능블록도이다.
- 도 9는 본 발명에 따른 빈발 부분그래프 마이닝 방법을 이용하여 도약 탐색 기법을 수행하는 방법을 설명하는 흐름도이다.
- 도 10은 본 발명에 따른 빈발 부분그래프 마이닝 방법을 이용하여 모델기반 탐색 기법을 수행하는 방법을 설명하는 흐름도이다.

도 11은 항암 화합물인 molt-4, MCF-7, TEAST로 구성된 그래프 데이터베이스를 모델기반 탐색 트리 방식으로 빈발 부분그래프 마이닝시 본원발명에 따른 빈발 부분그래프 마이닝에 따른 정규화 시간과 빈발 부분그래프에 대해 각각 정규화 연산을 수행하는 종래 방법에서 정규화 연산에 소요되는 시간을 비교한 그래프이다.

발명을 실시하기 위한 구체적인 내용

[0025] 이하 첨부한 도면을 참고로 본 발명에 따른 빈발 부분그래프의 마이닝 방법에 대해 보다 구체적으로 살펴본다.

[0026] 도 2는 본 발명에 따른 빈발 부분그래프의 마이닝 방법을 설명하기 위한 흐름도이다.

[0027] 도 2를 참고로 컴퓨터에서 그래프 데이터베이스에 저장되어 있는 그래프로부터 빈발 부분그래프를 마이닝하는 방법을 보다 구체적으로 살펴보면, 복잡한 구조를 가지는 그래프들이 대용량으로 저장되어 있는 그래프 데이터베이스로부터 최소 지지도(MinSup) 이상으로 출현하는 빈발 부분그래프를 추출한다(S10). 빈발 부분그래프(FG)은 아래의 수학적식(1)과 같이 정의된다.

[0028] [수학적식 1]

$$FG = \frac{|g \subseteq G|}{|GD|} > MinSup$$

[0030] 여기서 $|g \subseteq G|$ 는 부분그래프(g)를 가지는 그래프(G)의 개수이며 $|GD|$ 는 그래프 데이터베이스에 존재하는 그래프의 개수이다.

[0031] 추출한 빈발 부분그래프의 DFS 코드를 생성한다(S20). 추출한 빈발 부분그래프의 DFS 코드는 추출한 빈발 부분그래프를 구성하는 각 에지의 확장 순서로 깊이 우선 탐색(DFS) 방식으로 생성되는데, 빈발 부분그래프를 구성하는 각 에지의 DFS 코드는 에지를 구성하는 2개 버텍스(vertex) 식별자, 2개 버텍스 레이블 및 2개 버텍스를 연결하는 에지 레이블로 구성된다. 즉 빈발 부분그래프를 구성하는 각 에지의 DFS 코드는 아래의 수학적식(2)과 같이 표현된다.

[0032] [수학적식 2]

[0033] DFS 코드={i, j, l(v_i), l(e(v_i, v_j)), l(v_j)}

[0034] 여기서 i, j는 각각 에지(e)를 구성하는 2개 버텍스(vertex) 식별자, l(v_i)와 l(v_j)는 각각 2개 버텍스 레이블, l(e(v_i, v_j))는 2개 버텍스를 연결하는 에지 레이블이다.

[0035] 생성한 빈발 부분그래프의 DFS 코드와 정규 그래프 탐색 트리의 정규 코드를 비교하여 추출한 빈발 부분그래프가 최소 DFS 코드를 가지는 정규 그래프인지 판단한다(S30). 판단 결과 추출한 빈발 부분그래프가 정규 그래프인 경우 추출한 빈발 부분그래프에서 에지를 1개씩 확장하여 추출한 빈발 부분그래프로부터 생성되는 새로운 부분그래프를 생성한다(S40). 생성한 새로운 부분그래프에 대해 앞서 설명한 S10 단계 내지 S40 단계를 반복 수행하여 그래프 데이터베이스에 존재하는 최소 지지도를 만족하는 모든 빈발 부분그래프를 탐색한다.

[0036] 한편, 판단 결과 추출한 빈발 부분그래프가 최소 DFS 코드를 가지는 정규 그래프가 아닌 경우, 추출한 빈발 부분그래프의 각 에지의 DFS 코드가 정규 그래프 탐색 트리에 존재하는지 여부를 판단하여 추출한 빈발 부분그래프가 새로운 빈발 부분그래프인지 판단하고(S50), 판단 결과 추출한 빈발 부분그래프가 새로운 빈발 부분그래프인 경우 추출한 빈발 부분그래프의 DFS 코드가 정규 코드인지 판단하여 추출한 빈발 부분그래프의 DFS 코드를 정규 그래프 탐색 트리에 추가한다(S60). 한편, 추출한 빈발 부분그래프 중 정규 그래프 탐색 트리의 정규 코드를 가지지 않는 빈발 부분그래프는 제거하여 빈발 부분그래프가 중복하여 확장되는 것을 방지한다.

[0037] 도 3은 그래프 데이터베이스에서 확장되는 빈발 부분그래프를 설명하기 위한 도면이다.

[0038] 도 3을 참고로 보다 구체적으로 살펴보면, 그래프 데이터베이스에 존재하는 3개의 그래프(G₁, G₂, G₃)에서 최소 지지도(2/3)를 가지는 1에지의 빈발 부분그래프(0-H)를 추출한다. 1에지 빈발 부분그래프에서 에지를 1개씩 확

장하여 1에지 빈발 부분그래프에서 확장되는 2에지 빈발 부분그래프(g_1, g_2, g_3, g_4, g_5)를 판단한다. 여기서 1에지를 부모 에지로 표현하고 1에지에서 확장된 에지를 자식 에지로 표현할 수 있다. 2에지 제1 빈발 부분그래프(g_1, g_2, g_3)와 2에지 제2 빈발 부분그래프(g_4, g_5)는 부모 에지에서 자식 에지를 확장하여 생성되는 부분그래프 중 최소 지지도($2/3$)를 가지는 빈발 부분그래프이다.

[0039] 2에지 제1 빈발 부분그래프(g_1, g_2, g_3) 각각의 DFS 코드를 에지 확장 순서로 계산하고 추출한 제1 빈발 부분그래프(g_1, g_2, g_3)의 DFS 코드를 정규 그래프 탐색 트리의 정규 코드와 비교하여 제1 빈발 부분그래프(g_1, g_2, g_3) 중 최소 DFS 코드를 가지는 빈발 부분그래프(g_2)로부터 다시 자식 에지를 확장하여 3에지 부분그래프를 판단한다. 이와 동일하게 2에지 제2 빈발 부분그래프(g_4, g_5) 각각의 DFS 코드를 에지 확장 순서로 계산하고 제2 빈발 부분그래프(g_4, g_5)의 DFS 코드를 정규 그래프 탐색 트리의 정규 코드와 비교하여 제2 빈발 부분그래프(g_4, g_5) 중 최소 DFS 코드를 가지는 빈발 부분그래프(g_4)로부터 자식 에지를 확장하여 3에지 부분그래프를 판단한다.

[0040] 한편, 정규 그래프 탐색 트리의 정규 코드를 가지지 않는 추출 빈발 부분그래프(g_1, g_3, g_5)는 제거하여 중복되어 빈발 부분그래프가 확장되는 것을 방지한다.

[0041] 도 4와 도 5는 각각 제1 빈발 부분그래프와 제2 빈발 부분그래프의 DFS 코드를 나타낸 것이며, 도 6은 정규 그래프 탐색 트리의 일 예와 정규 그래프 탐색 트리에 기초하여 추출한 빈발 부분그래프가 최소 DFS 코드를 가지는 정규 그래프인지 판단하는 방법을 설명하기 위한 도면이다.

[0042] 먼저 도 4를 참고로 살펴보면, 제1 빈발 부분그래프(g_1, g_3)는 2개의 에지로 구성되는데, 버텍스 레이블(H)와 버텍스 레이블(O) 사이의 부모 에지(e_1)의 DFS 코드는 (1, 2, H, 1, 0)로 표현되며, 버텍스 레이블(H)와 버텍스 레이블(C) 사이의 자식 에지(e_2)의 DFS 코드는 (2, 3, H, 1, C)로 표현된다. 한편, 제1 빈발 부분그래프(g_2)는 2개의 에지로 구성되는데, 버텍스 레이블(C)와 버텍스 레이블(H) 사이의 부모 에지(e_1)의 DFS 코드는 (1, 2, C, 1, H)로 표현되며, 버텍스 레이블(H)와 버텍스 레이블(O) 사이의 자식 에지(e_2)의 DFS 코드는 (2, 3, H, 1, 0)로 표현된다.

[0043] 한편, 도 5를 참고로 살펴보면, 제2 빈발 부분그래프(g_4)는 2개의 에지로 구성되는데, 버텍스 레이블(H)와 버텍스 레이블(O) 사이의 부모 에지(e_1)의 DFS 코드는 (1, 2, H, 1, 0)로 표현되며, 버텍스 레이블(O)와 버텍스 레이블(P) 사이의 자식 에지(e_2)의 DFS 코드는 (2, 3, 0, 1, P)로 표현된다. 한편, 제2 빈발 부분그래프(g_5)는 2개의 에지로 구성되는데, 버텍스 레이블(O)와 버텍스 레이블(P) 사이의 부모 에지(e_1)의 DFS 코드는 (1, 2, 0, 1, P)로 표현되며, 버텍스 레이블(O)와 버텍스 레이블(H) 사이의 자식 에지(e_2)의 DFS 코드는 (1, 3, 0, 1, H)로 표현된다.

[0044]

[0045] 이하, 제1 빈발 부분그래프와 제2 빈발 부분그래프가 정규 코드인지 정규 그래프 탐색 트리에 기초하여 판단하는 과정을 도 6을 참고로 살펴보면, 제1 빈발 부분그래프(g_1, g_3)의 에지 확장 순서로 부모 에지(e_1)의 DFS 코드 (1, 2, H, 1, 0)가 정규 그래프 탐색 트리의 제1 저장 레벨에 존재하는지 판단한다. 판단 결과 부모 에지(e_1)의 DFS 코드가 정규 그래프 탐색 트리의 제1 저장 레벨에 존재하는 경우, 부모 에지(e_1)의 DFS 코드의 확장 포인터(A_1)을 따라 정규 그래프 탐색 트리의 제2 저장 레벨에 자식 에지(e_2)의 DFS 코드(2, 3, H, 1, C)가 존재하는지 판단한다. 판단 결과 정규 그래프 탐색 트리의 제2 저장 레벨에 자식 에지(e_2)의 DFS 코드(2, 3, H, 1, C)가 존재가 존재하지 않는 경우 제1 빈발 부분그래프 (g_1, g_3)는 최소 DFS 코드를 가지지 않는 정규 코드가 아니므로 삭제하여 빈발 부분그래프가 중복되어 확장되는 것을 방지한다. 한편, 제1 빈발 부분그래프(g_2)의 에지 확장 순서로 부모 에지(e_1)의 DFS 코드(1, 2, C, 1, H)가 정규 그래프 탐색 트리의 제1 저장 레벨에 존재하는지 판단한다. 판단 결과 부모 에지(e_1)의 DFS 코드가 정규 그래프 탐색 트리의 제1 저장 레벨에 존재하는 경우,

부모 에지(e_1)의 DFS 코드의 확장 포인터(A_2)을 따라 정규 그래프 탐색 트리의 제2 저장 레벨에 자식 에지(e_2)의 DFS 코드(2, 3, H, 1, 0)가 존재하는지 판단한다. 판단 결과 정규 그래프 탐색 트리의 제2 저장 레벨에 자식 에지(e_2)의 DFS 코드(2, 3, H, 1, 0)가 존재하는 경우 제1 빈발 부분그래프(g_2)는 최소 DFS 코드를 가지는 정규 그래프로 판단하며, 정규 그래프로 판단된 제1 빈발 부분그래프(g_2)를 에지 확장하여 제1 빈발 부분그래프(g_2)로부터 확장된 새로운 빈발 부분그래프를 추출한다.

[0046] 이와 동일하게 제2 빈발 부분그래프(g_4)의 부모 에지(e_1)와 자식 에지(e_2)의 DFS 코드가 에지 확장 순서에 대응하도록 정규 그래프 탐색 트리의 제1 저장 레벨과 제2 저장 레벨에 저장되어 있는 경우, 제2 빈발 부분그래프(g_4)를 정규 그래프로 판단하여 제2 빈발 부분그래프(g_4)를 에지 확장하여 제2 빈발 부분그래프(g_4)로부터 확장된 새로운 빈발 부분그래프를 추출하며, 제2 빈발 부분그래프(g_5)의 부모 에지(e_1)와 자식 에지(e_2)의 DFS 코드가 에지 확장 순서에 대응하도록 정규 그래프 탐색 트리의 제1 저장 레벨과 제2 저장 레벨에 저장되어 있지 않은 경우 제2 빈발 부분그래프(g_5)를 제거한다.

[0047] 도 7은 본 발명에 따른 정규 그래프 탐색 트리의 다른 예를 설명하기 위한 도면이다.

[0048] 도 7(a)에 도시되어 있는 정규 그래프(G_1)를 구성하는 각 에지($e_1, e_2, e_3, e_4, e_5, e_6$)에 대한 DFS 코드에서 벡스 레이블과 에지 레이블(에지의 결합 상태, 예를 들어 단일결합, 이중결합 등)로 표현되는 에지 코드를 생성하고, 각 에지의 에지 코드가 정규 그래프에서 빈발하게 반복되는 횟수에 따라 서로 다른 데이터로 표현하는 압축 코드를 생성한다. 본 발명에서 압축 코드는 호프만 코드를 사용하여 에지 코드를 압축한 압축 코드를 생성한다. 본 발명이 적용되는 분야에 따라 다양한 압축 방식으로 에지 코드를 압축하여 압축 코드를 생성할 수 있으며, 이는 본 발명의 범위에 속한다.

[0049] 도 7(b)에 도시되어 있는 것과 같이, 정규 그래프(G_1)를 구성하는 각 에지의 DFS 코드를 각 에지의 벡스 식별자와 압축 코드로 표현되는 압축 정규 코드로 변환하여 정규 그래프 탐색 트리를 저장함으로써, 정규 그래프 탐색 트리의 저장을 보다 적은 메모리로 할 수 있다.

[0050] 한편, 도 7(c)에 도시되어 있는 것과 같이 에지 코드와 각 에지 코드에 해당하는 압축 코드의 테이블을 저장하고 있는 순람표를 별도로 저장하고 순람표에 의해 각 에지의 압축 정규 코드를 생성함으로써 빠르게 정규 그래프의 압축 정규 코드를 생성할 수 있다.

[0051] 도 8은 본 발명에 따른 정규 그래프 탐색 트리의 생성 장치를 설명하기 위한 기능 블록도이다.

[0052] 도 8을 참고로 보다 구체적으로 살펴보면, 그래프 데이터베이스(110)에는 복잡한 구조를 가지는 그래프들이 대용량으로 저장되어 있으며, 빈발 부분그래프 추출부(120)는 그래프 데이터베이스(110)에 저장되어 있는 그래프에서 설정한 최소 지지도(MinSup) 이상으로 출현하는 빈발 부분그래프를 추출한다.

[0053] 정규 그래프 판단부(130)는 탐색 트리 저장부(160)에 저장되어 있는 정규 코드로 저장되어 있는 정규 그래프 탐색 트리에 기초하여 추출한 빈발 부분그래프가 최소 DFS 코드를 가지는 정규 그래프인지 판단한다. 정규 그래프 판단부(130)의 판단 결과에 기초하여 추출한 빈발 부분그래프가 정규 그래프인 경우, 빈발 부분그래프 확장부(140)는 추출한 빈발 부분그래프에서 에지 확장이 가능한지 판단하며, 에지 확장이 가능한 경우 빈발 부분그래프를 에지 확장하여 확장된 부분그래프를 생성한다. 빈발 부분그래프 추출부(120) 및 정규 그래프 판단부(130)는 에지 확장된 부분그래프에 대해 앞서 설명한 것과 동일하게 반복 동작한다.

[0054] 한편, 정규 그래프 판단부(130)의 판단 결과에 기초하여 추출한 빈발 부분그래프가 탐색 트리 저장부(160)에 저장된 정규 그래프 탐색 트리에 존재하지 않는 경우 탐색 트리 생성부(150)는 추출한 빈발 부분그래프를 정규화 연산하여 추출한 빈발 부분그래프의 최소 DFS(Depth First Search) 코드를 획득하고, 획득한 빈발 부분그래프의 최소 DFS 코드에 기초하여 빈발 부분그래프를 구성하는 각 에지의 DFS 코드와 각 에지 사이의 순차적인 확장 포인터를 생성하며, 생성한 각 에지의 DFS 코드와 각 에지 사이의 순차적인 확장 포인터를 탐색 트리 저장부(160)에 추가하여 정규 그래프 탐색 트리를 갱신한다.

[0055] 압축부(180)는 정규 그래프 탐색 트리 저장부(160)에 저장되어 있는, 정규 그래프를 구성하는 모든 에지의 에지

코드 발생 빈도를 계산하고 에지 코드의 발생 빈도 순서에 기초하여 정규 그래프 탐색 트리에 저장되어 있는 모든 에지의 DFS 코드를 정규 압축 코드로 압축하여 정규 그래프 탐색 트리에 재저장한다.

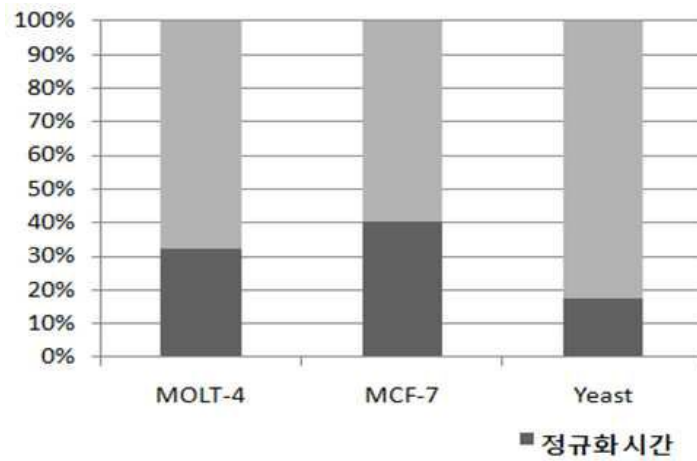
- [0056] 도 9는 본 발명에 따른 빈발 부분그래프 마이닝 방법을 이용하여 도약 탐색 기법을 수행하는 방법을 설명하는 흐름도이다.
- [0057] 도 9를 참고로 보다 구체적으로 살펴보면, 최소 지지도가 입력되면(S110) 그래프 데이터베이스에서 입력된 최소 지지도를 가지는 빈발 부분그래프를 추출하고(S120), 추출한 빈발 부분그래프의 DFS 코드를 생성한다(S130).
- [0058] 생성한 빈발 부분그래프의 DFS 코드와 정규 그래프 탐색 트리의 정규 코드를 비교하여 추출한 빈발 부분그래프가 최소 DFS 코드를 가지는 정규 그래프인지 판단한다(S140). 판단 결과 추출한 빈발 부분그래프가 정규 그래프인 경우 추출한 빈발 부분그래프에서 에지가 확장되는 부분그래프가 생성되는지 판단하여(S150), 에지가 확장될 때까지 추출한 빈발 부분그래프로부터 생성되는 새로운 부분그래프를 생성하고 생성한 새로운 부분그래프에 대해 앞서 설명한 S120 단계 내지 S140 단계를 반복 수행하여 그래프 데이터베이스에 존재하는 최소 지지도를 만족하는 모든 빈발 부분그래프를 탐색한다.
- [0059] 한편, 최소 지지도가 설정된 다른 최소 지지도로 변경되는 경우, 그래프 데이터베이스에서 변경된 최소 지지도로 위에서 설명한 단계 S120 내지 S150 단계를 수행하여 변경된 최소 지지도로 그래프 데이터베이스의 모든 빈발 부분그래프를 추출한다.
- [0060] 도 10은 본 발명에 따른 빈발 부분그래프 마이닝 방법을 이용하여 모델기반 탐색 기법을 수행하는 방법을 설명하는 흐름도이다.
- [0061] 도 10을 참고로 보다 구체적으로 살펴보면, 그래프 데이터베이스에서 추출한 빈발 부분그래프 중에서 1차 설정된 빈발 부분그래프를 구비하는 그래프와 설정된 빈발 부분그래프를 가지지 않는 그래프를 나누어 그래프 데이터베이스로부터 설정된 빈발 부분그래프를 구비하는 소그룹 그래프 데이터베이스를 생성한다(S210).
- [0062] 생성한 소그룹 그래프 데이터베이스에서 설정된 최소 지지도를 가지는 빈발 부분그래프를 추출하고(S220), 추출한 빈발 부분그래프의 DFS 코드를 생성한다(S230).
- [0063] 생성한 빈발 부분그래프의 DFS 코드와 정규 그래프 탐색 트리의 정규 코드를 비교하여 추출한 빈발 부분그래프가 최소 DFS 코드를 가지는 정규 그래프인지 판단한다(S240). 판단 결과 추출한 빈발 부분그래프가 정규 그래프인 경우 추출한 빈발 부분그래프에서 에지가 확장되는 부분그래프가 생성되는지 판단하여(S250), 에지가 확장될 때까지 추출한 빈발 부분그래프로부터 생성되는 새로운 부분그래프를 생성하고 생성한 새로운 부분그래프에 대해 앞서 설명한 S220 단계 내지 S250 단계를 반복 수행하여 소그룹 그래프 데이터베이스에 존재하는 최소 지지도를 만족하는 모든 빈발 부분그래프를 탐색한다.
- [0064] 한편, 소그룹 데이터베이스를 구성하는 그래프의 수가 임계값(TH)보다 큰지 판단하여, 소그룹 데이터베이스를 구성하는 그래프의 수가 임계값(TH)보다 큰 경우(260), 소그룹 데이터베이스에서 추출한 빈발 부분그래프 중에서 2차 설정된 빈발 부분그래프를 구비하는 그래프와 2차 설정된 빈발 부분그래프를 가지지 않는 그래프를 나누어 소그룹 그래프 데이터베이스로부터 2차 설정된 빈발 부분그래프를 구비하는 소그룹 그래프 데이터베이스를 생성하여 앞서 설명한 S220 단계 내지 S250 단계를 반복하여 수행한다. 여기서 1차 설정된 빈발 부분그래프와 2차 설정된 빈발 부분그래프는 분류 성능이 높은 빈발 부분그래프인 것을 특징으로 한다.
- [0065] 도 11은 항암 화합물인 molt-4, MCF-7, TEAST로 구성된 그래프 데이터베이스를 모델기반 탐색 트리 방식으로 빈발 부분그래프 마이닝시 본원발명에 따른 빈발 부분그래프 마이닝에 따른 정규화 시간과 빈발 부분그래프에 대해 각각 정규화 연산을 수행하는 종래 방법에서 정규화 연산에 소요되는 시간을 비교한 그래프이다. 도 11에 도시되어 있는 바와 같이, 본원발명에서는 한번의 정규화 연산만으로 유일한 최소 DFS 코드를 구비하는 정규 그래프 탐색 트리를 생성하고, 생성한 정규 그래프 탐색 트리에 기초하여 빈발 부분그래프가 정규 그래프인지 판단하는 반면 종래 모델기반 탐색 트리의 빈발 부분그래프 마이닝은 빈발 부분그래프에 대해 각각 정규화 연산을 수행함으로써, 본원발명에 따른 빈발 부분그래프 마이닝 방법을 이용한 모델 기반 탐색 트리보다 정규화 연산을 위해 오랜 시간이 소요됨을 알 수 있다.

부호의 설명

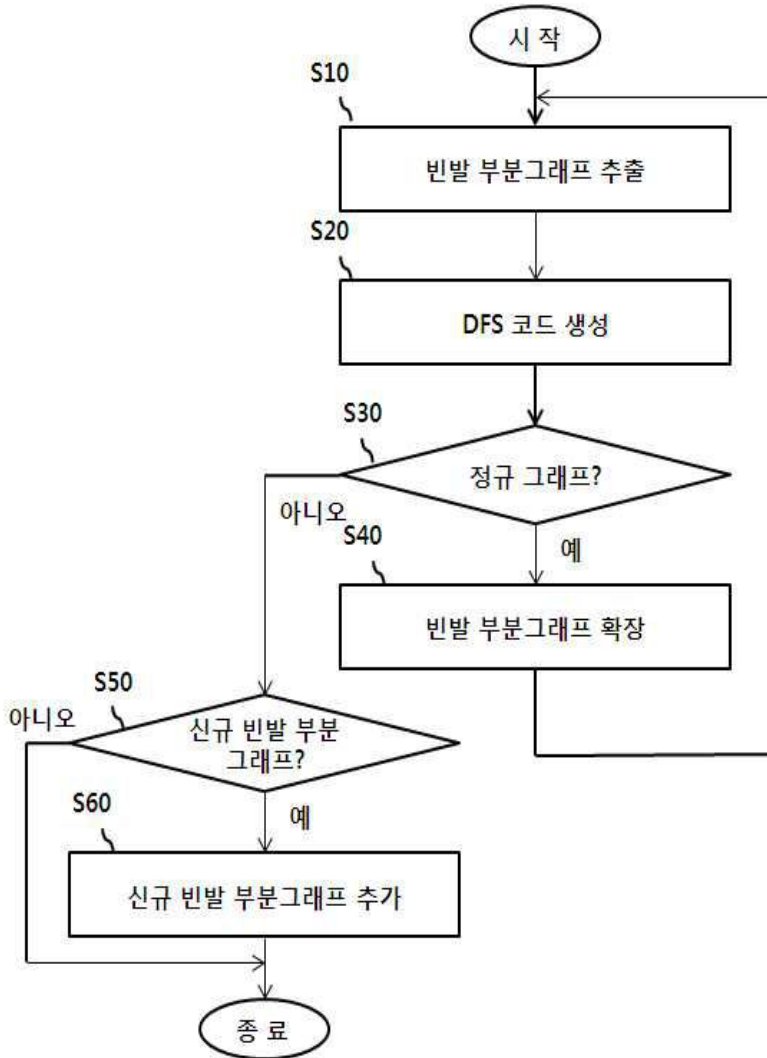
[0066]	110: 그래프 DB	120: 빈발 부분그래프 추출부
	130: 정규 그래프 판단부	140: 빈발 부분그래프 확장부
	150: 탐색 트리 생성부	160: 탐색 트리 저장부
	170: 압축부	

도면

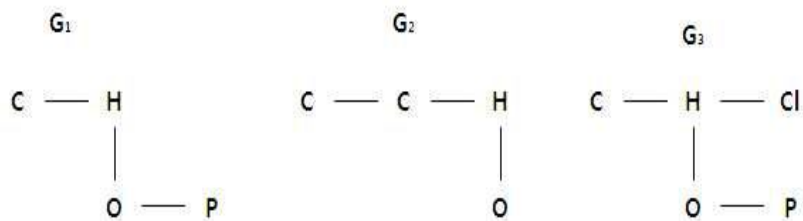
도면1



도면2



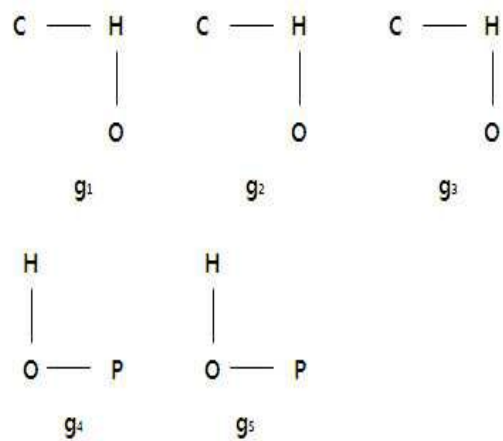
도면3



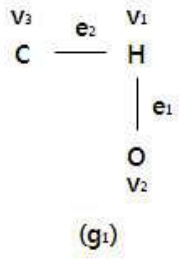
1에지



2에지

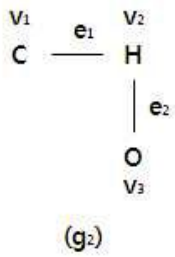


도면4



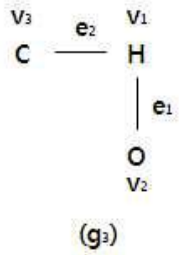
에지	DFS 코드
e1	(1, 2, H, 1, O)
e2	(2, 3, H, 1, C)

(a)



에지	DFS 코드
e1	(1, 2, C, 1, H)
e2	(2, 3, H, 1, O)

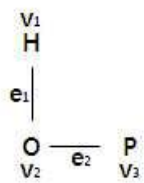
(b)



에지	DFS 코드
e1	(1, 2, H, 1, O)
e2	(2, 3, H, 1, C)

(c)

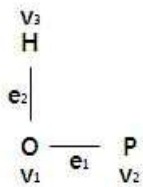
도면5



(g₄)

에지	DFS 코드
e1	(1, 2, H, 1, O)
e2	(2, 3, O, 1, P)

(a)

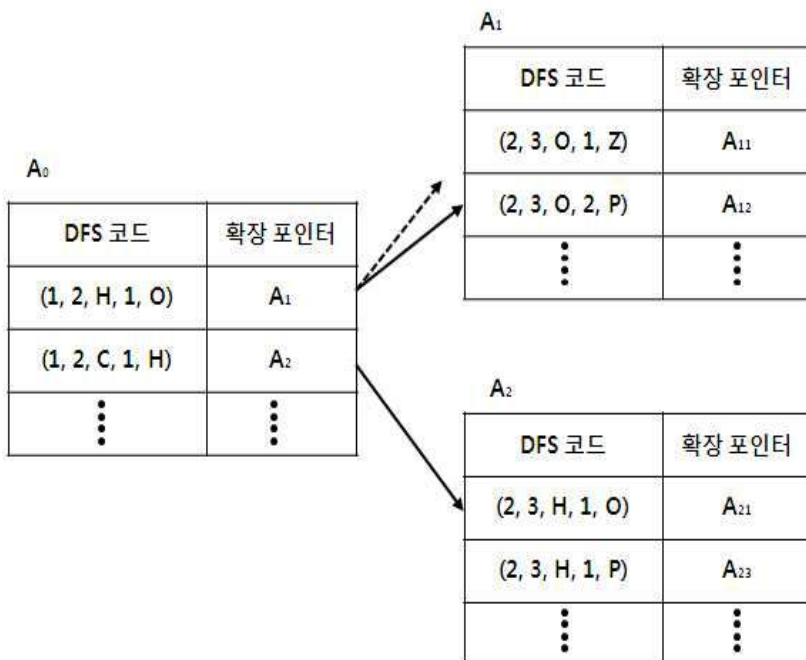


(g₅)

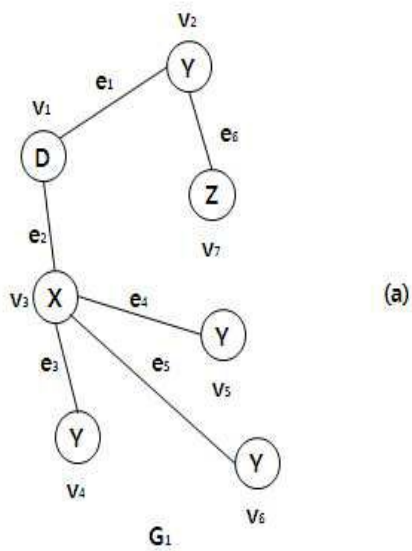
에지	DFS 코드
e1	(1, 2, O, 1, P)
e2	(1, 3, O, 1, H)

(b)

도면6



도면7



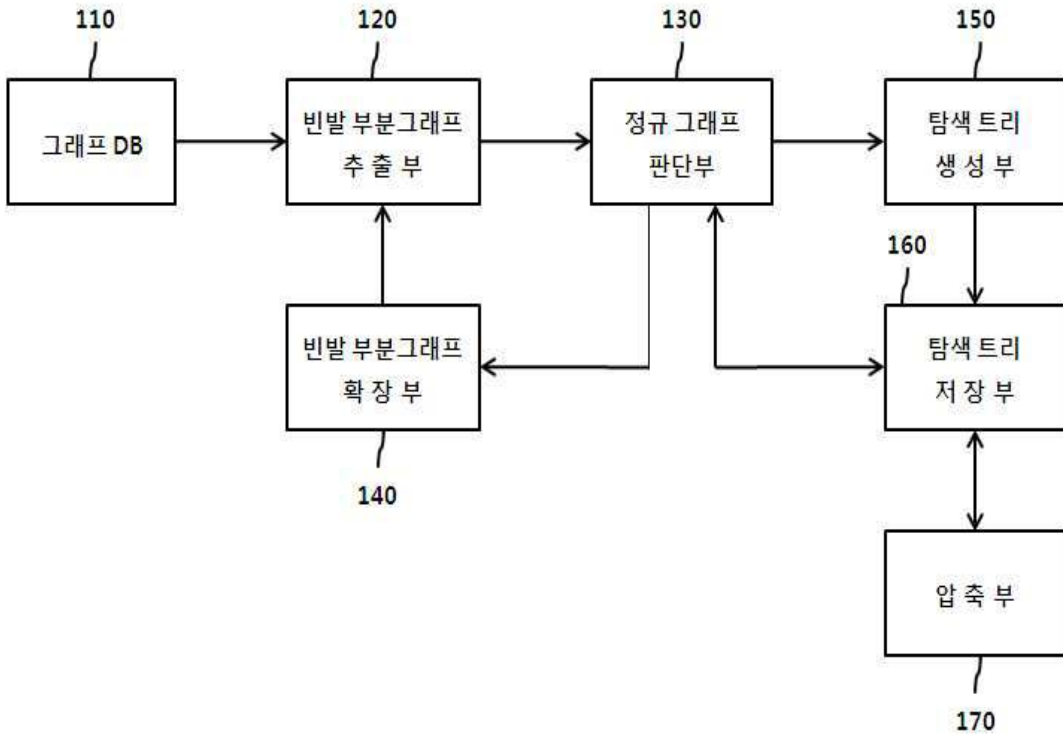
에지	압축 정규 코드		
	버텍스1	버텍스2	압축코드
e1	1	2	1110
e2	1	3	110
e3	3	4	0
e4	3	5	0
e5	3	6	0
e6	2	7	10

(b)

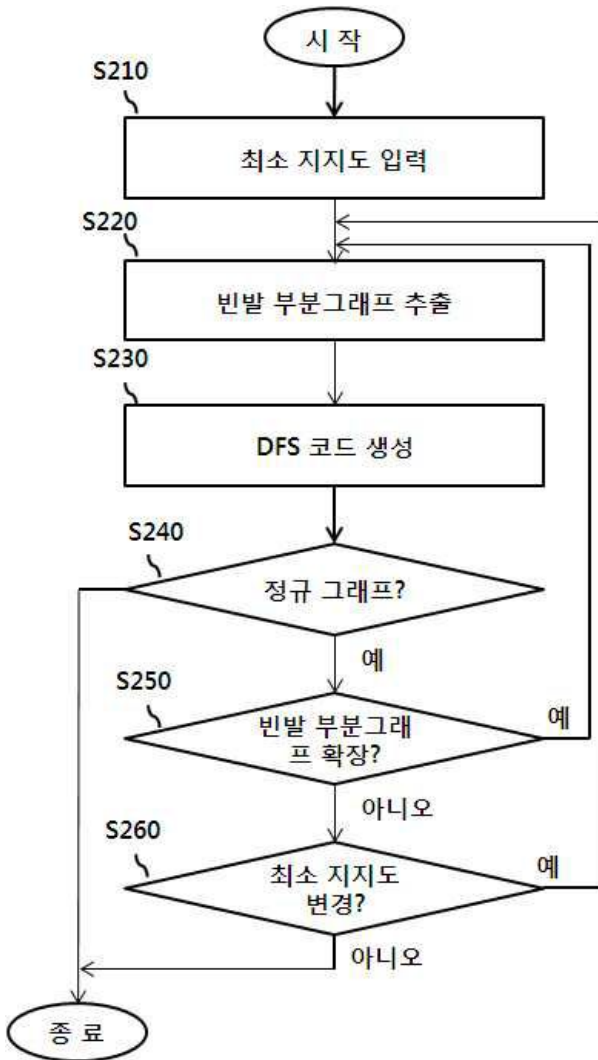
에지 코드	호프만 코드
X,1,Y	0
Y,2,Z	10
D,1,X	110
D,1,Y	1110

(c)

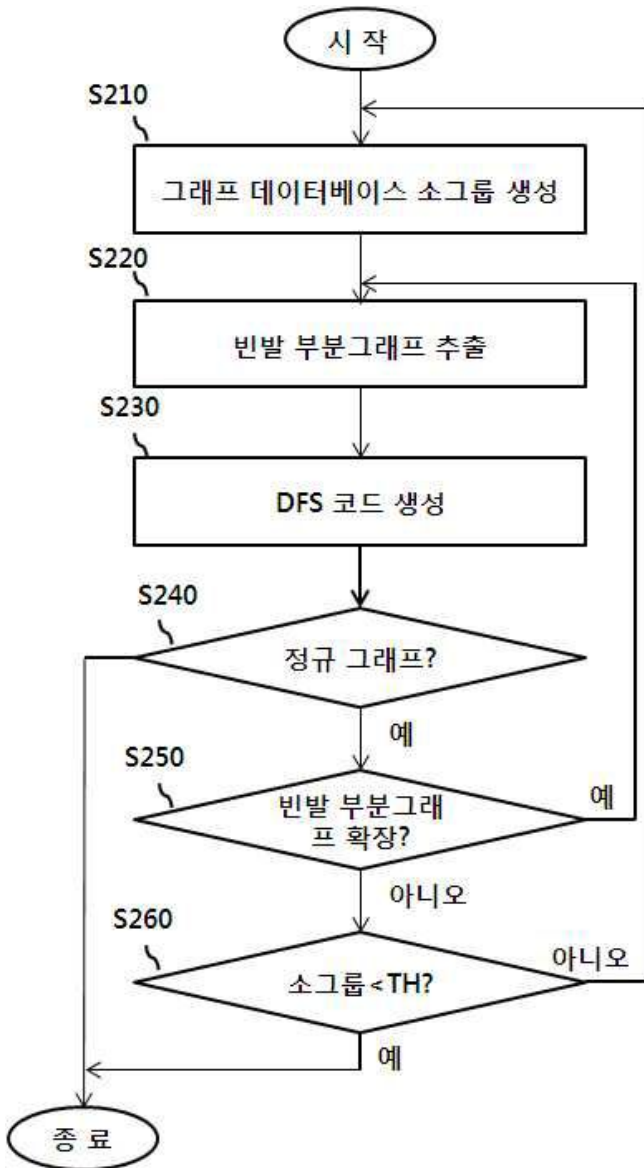
도면8



도면9



도면10



도면11

