



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2018-0099422
(43) 공개일자 2018년09월05일

- | | |
|---|---|
| <p>(51) 국제특허분류(Int. Cl.)
G06F 21/56 (2013.01) G06F 17/30 (2006.01)</p> <p>(52) CPC특허분류
G06F 21/56 (2013.01)
G06F 17/30705 (2013.01)</p> <p>(21) 출원번호 10-2017-0078565</p> <p>(22) 출원일자 2017년06월21일
심사청구일자 2017년06월21일</p> <p>(30) 우선권주장
1020170025313 2017년02월27일 대한민국(KR)</p> | <p>(71) 출원인
충남대학교산학협력단
대전광역시 유성구 대학로 99 (궁동, 충남대학교)</p> <p>(72) 발명자
김형식
대전광역시 유성구 대학로 99</p> <p>김기수
대전광역시 동구 계족로489번길 73, 115동 601호
(뒷면에 계속)</p> <p>(74) 대리인
홍성욱, 심경식</p> |
|---|---|

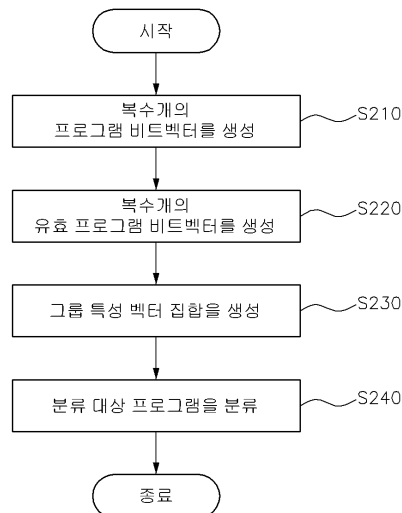
전체 청구항 수 : 총 13 항

(54) 발명의 명칭 프로그램 분류 방법 및 장치

(57) 요약

비트벡터 생성부가, 미리 저장된 복수 개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록(basic block)을 나타내는 복수 개의 프로그램 비트벡터(bit vector)를 복수 개의 프로그램 별로 각각 생성하는 단계, 유효 비트벡터 생성부가, 복수 개의 프로그램 비트벡터에 기초하여 비트벡터 마스크(mask)를 생성하고, 복수 개의 프로그램 비트벡터 각각에 비트벡터 마스크를 적용하여 복수 개의 유효 프로그램 비트벡터를 생성하는 단계, 그룹 특성 벡터 집합 생성부가, 복수 개의 유효 프로그램 비트벡터에 기설정된 조건을 적용하여 결정된 비트벡터인 복수 개의 그룹 특성 벡터를 포함하는 그룹 특성 벡터 집합을 생성하는 단계 및 프로그램 분류부가, 미리 저장된 적어도 하나의 분류 대상 프로그램과 복수 개의 그룹 특성 벡터를 비교하여 적어도 하나의 분류 대상 프로그램을 분류하는 단계를 포함하는, 프로그램 분류 방법 및 이를 위한 장치에 관한 것이다.

대표도 - 도2



(72) 발명자

윤정무

대전광역시 유성구 대학로145번길 22, 405호

이경준

대전광역시 중구 태평로 35, 215동 401호

이 발명을 지원한 국가연구개발사업

과제고유번호 B0101-16-0300

부처명 한국인터넷진흥원

연구관리전문기관 정보통신기술진흥센터

연구사업명 방송통신연구개발사업

연구과제명 사이버 공격의 사전 사후 대응을 위한 사이버 블랙박스 및 통합 사이버보안 상황분석 기술
개발

기 여 율 1/1

주관기관 충남대학교 산학협력단

연구기간 2016.03.01 ~ 2017.02.28

명세서

청구범위

청구항 1

비트벡터 생성부가, 미리 저장된 복수 개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록(basic block)을 나타내는 복수 개의 프로그램 비트벡터(bit vector)를 상기 복수 개의 프로그램 별로 각각 생성하는 단계;

유효 비트벡터 생성부가, 상기 복수 개의 프로그램 비트벡터에 기초하여 비트벡터 마스크(mask)를 생성하고, 상기 복수 개의 프로그램 비트벡터 각각에 상기 비트벡터 마스크를 적용하여 복수 개의 유효 프로그램 비트벡터를 생성하는 단계;

그룹 특성 벡터 집합 생성부가, 상기 복수 개의 유효 프로그램 비트벡터에 기설정된 조건을 적용하여 결정된 비트벡터인 복수 개의 그룹 특성 벡터를 포함하는 그룹 특성 벡터 집합을 생성하는 단계; 및

프로그램 분류부가, 미리 저장된 적어도 하나의 분류 대상 프로그램과 상기 복수 개의 그룹 특성 벡터를 비교하여 상기 적어도 하나의 분류 대상 프로그램을 분류하는 단계를 포함하는, 프로그램 분류 방법.

청구항 2

제1항에 있어서,

상기 복수 개의 프로그램 비트벡터를 상기 복수 개의 프로그램 별로 각각 생성하는 단계는,

상기 복수 개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록 각각에 해시함수를 적용하여 상기 모든 함수 또는 모든 기본 블록 각각을 임의의 비트열로 변환하는 단계;

상기 모든 함수 또는 모든 기본 블록 각각에 대응되는 상기 임의의 비트열 각각에서 기설정된 개수의 연속된 비트열인 연속 비트열을 각각 추출하는 단계;

추출된 상기 연속 비트열 각각을 기설정된 조건에 따라 변환하여 상기 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호를 생성하는 단계; 및

상기 복수 개의 프로그램 각각에 대응되며 기설정된 개수의 비트를 포함하는 복수 개의 비트벡터 상에서 상기 식별 번호에 각각 대응되는 비트의 비트값을 설정하여 상기 복수 개의 프로그램 비트벡터를 각각 생성하는 단계를 포함하는, 프로그램 분류 방법.

청구항 3

제1항에 있어서,

상기 복수 개의 유효 프로그램 비트벡터를 생성하는 단계는,

상기 복수 개의 프로그램 비트벡터 상호간에 서로 대응되는 비트의 비트값을 모두 합산하여 상기 모든 함수 또는 모든 기본 블록 각각의 빈도수를 나타내는 함수 빈도수 벡터를 생성하는 단계;

상기 함수 빈도수 벡터에 포함된 상기 모든 함수 또는 모든 기본 블록 각각의 빈도수에 기설정된 제1 임계값을 적용하여 상기 비트벡터 마스크를 생성하는 단계; 및

상기 복수 개의 프로그램 비트벡터 각각과 상기 비트벡터 마스크를 서로 AND 연산하여 상기 복수 개의 유효 프로그램 비트벡터를 생성하는 단계를 포함하는, 프로그램 분류 방법.

청구항 4

제3항에 있어서,

상기 비트벡터 마스크는,

상기 모든 함수 또는 모든 기본 블록 각각의 빈도수 중 상기 기설정된 제1 임계값보다 큰 빈도수에 대응되는 함

수 또는 기본 블록에 대한 비트의 비트값을 "1"로 설정하고, 상기 기설정된 제1 임계값보다 작은 빈도수에 대응되는 함수 또는 기본 블록에 대한 비트의 비트값을 "0"으로 설정한 비트벡터인, 프로그램 분류 방법.

청구항 5

제1항에 있어서,

상기 그룹 특성 벡터 집합이 공집합인 경우,

상기 그룹 특성 벡터 집합을 생성하는 단계는,

상기 복수 개의 유효 프로그램 비트벡터 중 선택된 하나의 유효 프로그램 비트벡터를 그룹 특성 벡터로 결정하여 상기 그룹 특성 벡터 집합에 저장하는 단계를 포함하는, 프로그램 분류 방법.

청구항 6

제1항에 있어서,

상기 그룹 특성 벡터 집합이 미리 저장된 적어도 하나의 그룹 특성 벡터를 포함하는 경우,

상기 그룹 특성 벡터 집합을 생성하는 단계는,

상기 적어도 하나의 그룹 특성 벡터 중 i 번째 그룹 특성 벡터와 상기 복수 개의 유효 프로그램 비트벡터 중 j 번째 유효 프로그램 비트벡터 상호 간의 제1 유사도를 산출하는 단계;

상기 제1 유사도에 기설정된 제2 임계값을 적용하여 상기 i 번째 그룹 특성 벡터와 상기 j 번째 유효 프로그램 비트벡터가 서로 유사한지 여부를 판단하는 제1 판단 단계; 및

상기 제1 판단 결과에 기초하여 상기 그룹 특성 벡터 집합에 저장될 그룹 특성 벡터를 결정하는 제1 결정 단계를 포함하는, 프로그램 분류 방법.

청구항 7

제6항에 있어서,

상기 i 번째 그룹 특성 벡터와 상기 j 번째 유효 프로그램 비트벡터가 서로 비유사한 경우,

상기 제1 결정 단계는,

상기 j 번째 유효 프로그램 비트벡터를 그룹 특성 벡터로 결정하여 상기 그룹 특성 벡터 집합에 저장하는 단계를 포함하는, 프로그램 분류 방법.

청구항 8

제6항에 있어서,

상기 i 번째 그룹 특성 벡터와 상기 j 번째 유효 프로그램 비트벡터가 서로 유사한 경우,

상기 제1 결정 단계는,

상기 i 번째 그룹 특성 벡터와 상기 j 번째 유효 프로그램 비트벡터를 서로 AND 연산하여 공통 비트벡터를 생성하는 단계;

상기 i 번째 그룹 특성 벡터와 상기 공통 비트벡터 상호 간의 제2 유사도 및 상기 j 번째 유효 프로그램 비트벡터와 상기 공통 비트벡터 상호 간의 제3 유사도를 각각 산출하는 단계;

상기 제2 유사도 및 상기 제3 유사도 각각에 상기 제2 임계값을 적용하여 상기 i 번째 그룹 특성 벡터와 상기 공통 비트벡터가 서로 유사한지 여부 및 상기 j 번째 유효 프로그램 비트벡터와 상기 공통 비트벡터가 서로 유사한지 여부를 각각 판단하는 제2 판단 단계; 및

상기 제2 판단 결과에 기초하여 상기 그룹 특성 벡터 집합에 저장될 그룹 특성 벡터를 결정하는 제2 결정 단계를 포함하는, 프로그램 분류 방법.

청구항 9

제8항에 있어서,

상기 i 번째 그룹 특성 벡터와 상기 공통 비트벡터가 서로 유사하고, 상기 j 번째 유효 프로그램 비트벡터와 상기 공통 비트벡터가 서로 유사한 경우,

상기 제2 결정 단계는,

상기 i 번째 그룹 특성 벡터를 상기 그룹 특성 벡터 집합에서 삭제하는 단계; 및

상기 공통 비트벡터를 그룹 특성 벡터로 결정하여 상기 그룹 특성 벡터 집합에 저장하는 단계를 포함하는, 프로그램 분류 방법.

청구항 10

제8항에 있어서,

상기 i 번째 그룹 특성 벡터와 상기 공통 비트벡터가 서로 비유사하고, 상기 j 번째 유효 프로그램 비트벡터와 상기 공통 비트벡터가 서로 유사한 경우,

상기 제2 결정 단계는,

상기 공통 비트벡터를 그룹 특성 벡터로 결정하여 상기 그룹 특성 벡터 집합에 저장하는 단계를 포함하는, 프로그램 분류 방법.

청구항 11

제8항에 있어서,

상기 i 번째 그룹 특성 벡터와 상기 공통 비트벡터가 서로 유사하고, 상기 j 번째 유효 프로그램 비트벡터와 상기 공통 비트벡터가 서로 비유사한 경우,

상기 제2 결정 단계는,

상기 i 번째 그룹 특성 벡터를 상기 그룹 특성 벡터 집합에서 삭제하는 단계; 및

상기 j 번째 유효 프로그램 비트벡터 및 상기 공통 비트벡터 각각을 그룹 특성 벡터로 결정하여 상기 그룹 특성 벡터 집합에 저장하는 단계를 포함하는, 프로그램 분류 방법.

청구항 12

제1항에 있어서,

상기 적어도 하나의 분류 대상 프로그램을 분류하는 단계는,

상기 복수 개의 그룹 특성 벡터 각각에 대한 상기 적어도 하나의 분류 대상 프로그램 각각의 포함률을 산출하는 단계; 및

상기 포함률에 기설정된 제3 임계값을 적용하여 상기 적어도 하나의 분류 대상 프로그램 각각이 속하는 프로그램 그룹을 결정하는 단계를 포함하는, 프로그램 분류 방법.

청구항 13

미리 저장된 복수 개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록(basic block)을 나타내는 복수 개의 프로그램 비트벡터(bit vector)를 상기 복수 개의 프로그램 별로 각각 생성하는 비트벡터 생성부;

상기 복수 개의 프로그램 비트벡터에 기초하여 비트벡터 마스크(mask)를 생성하고, 상기 복수 개의 프로그램 비트벡터 각각에 상기 비트벡터 마스크를 적용하여 복수 개의 유효 프로그램 비트벡터를 생성하는 유효 비트벡터 생성부;

상기 복수 개의 유효 프로그램 비트벡터에 기설정된 조건을 적용하여 결정된 비트벡터인 복수 개의 그룹 특성 벡터를 포함하는 그룹 특성 벡터 집합을 생성하는 그룹 특성 벡터 집합 생성부; 및

미리 저장된 적어도 하나의 분류 대상 프로그램과 상기 복수 개의 그룹 특성 벡터를 비교하여 상기 적어도 하나의 분류 대상 프로그램을 분류하는 프로그램 분류부를 포함하는, 프로그램 분류 장치.

발명의 설명

기술 분야

[0001] 본 발명은 프로그램에 포함된 함수 또는 기본 블록에 기초하여 프로그램을 분류하기 위한 프로그램 분류 방법 및 장치에 관한 것이다.

배경 기술

[0002] 컴퓨터 프로그래밍 기술 및 이동 통신 기술이 발전함에 따라 사용자들은 인터넷 환경하에서 다양한 컴퓨터 프로그램들에 손쉽게 접근할 수 있게 되었으며, 필요에 따른 각종 프로그램들을 다운로드하여 자신의 컴퓨터, 노트북, 스마트폰 등 각종 단말기에 저장할 수 있게 되었다.

[0003] 이러한, 인터넷 환경이 조성됨에 따라서 사용자들은 상술한 바와 같이 필요에 따른 각종 프로그램들을 다운로드할 수 있게 되었으나, 그 반대급부로 각종 위변조 프로그램, 악성 코드 등에 손쉽게 노출되는 문제가 발생하게 되었다.

[0004] 이러한 악성 코드 등은 백신 프로그램 등 각종 방어 프로그램이 업데이트됨에 대응하여 빠르게 변종 악성 코드로 변형되는 특징이 있으며, 이러한 변종 악성 코드에 대응하기 위해서는 해당 변종 악성 코드가 어떤 악성 코드로부터 변형된 악성 코드인지를 판단하는 것이 매우 중요하다.

[0005] 이를 위해, 종래에는 변종 악성 코드가 어떤 악성 코드로부터 변형된 악성 코드인지를 판단하기 위해서는 2개의 악성 코드 내에 포함된 함수 또는 기본 블록을 서로 일대일로 직접 비교하는 과정을 거쳐야 했다.

[0006] 그러나, 2 개의 악성 코드 각각에 포함된 함수 또는 기본 블록을 서로 일대일로 직접 비교하는 경우에 비교 대상이 되는 악성 코드 각각의 용량이 큰 상황에서 2개의 악성 코드를 비교하기 위해서는 과도하게 많은 시간이 요구되는 문제가 있다.

선행기술문헌

특허문헌

[0007] (특허문헌 0001) 한국 등록특허공보 제10-0572660호(2006.04.24.)

발명의 내용

해결하려는 과제

[0008] 본 발명의 목적은, 상기 문제점을 해결하기 위한 것으로 복수 개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록을 나타내는 복수 개의 프로그램 비트벡터를 생성하고, 생성된 프로그램 비트벡터에 비트벡터 마스크를 적용하여 복수 개의 유효 프로그램 비트벡터를 생성한 뒤, 생성된 복수 개의 유효 프로그램 비트벡터에 기초하여 그룹 특성 벡터 집합을 생성함으로써 분류 대상 프로그램이 어떤 그룹에 속하는지 여부를 분류하기 위한 것이다.

[0009] 본 발명이 해결하고자 하는 과제는 이상에서 언급한 과제(들)로 제한되지 않으며, 언급되지 않은 또 다른 과제(들)은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0010] 상기한 목적을 달성하기 위하여, 본 발명의 실시예에 따른 프로그램 분류 방법은, 비트벡터 생성부가, 미리 저장된 복수 개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록(basic block)을 나타내는 복수 개의 프로그램 비트벡터(bit vector)를 복수 개의 프로그램 별로 각각 생성하는 단계, 유효 비트벡터 생성부가, 복수 개의 프로그램 비트벡터에 기초하여 비트벡터 마스크(mask)를 생성하고, 복수 개의 프로그램 비트벡터 각각에 비트벡터 마스크를 적용하여 복수 개의 유효 프로그램 비트벡터를 생성하는 단계, 그룹 특성 벡터 집합 생성부가, 복수 개의 유효 프로그램 비트벡터에 기설정된 조건을 적용하여 결정된 비트벡터인 복수 개의 그룹 특성 벡터를 포함하는 그룹 특성 벡터 집합을 생성하는 단계 및 프로그램 분류부가, 미리 저장된 적어도 하나의 분류

대상 프로그램과 복수 개의 그룹 특성 벡터를 비교하여 적어도 하나의 분류 대상 프로그램을 분류하는 단계를 포함한다.

- [0011] 예컨대, 복수 개의 프로그램 비트벡터를 복수 개의 프로그램 별로 각각 생성하는 단계는, 복수 개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록 각각에 해시함수를 적용하여 모든 함수 또는 모든 기본 블록 각각을 임의의 비트열로 변환하는 단계, 모든 함수 또는 모든 기본 블록 각각에 대응되는 임의의 비트열 각각에서 기설정된 개수의 연속된 비트열인 연속 비트열을 각각 추출하는 단계, 추출된 연속 비트열 각각을 기설정된 조건에 따라 변환하여 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호를 생성하는 단계 및 복수 개의 프로그램 각각에 대응되며 기설정된 개수의 비트를 포함하는 복수 개의 비트벡터 상에서 식별 번호에 각각 대응되는 비트의 비트값을 설정하여 복수 개의 프로그램 비트벡터를 각각 생성하는 단계를 포함한다.
- [0012] 예를 들어, 복수 개의 유효 프로그램 비트벡터를 생성하는 단계는, 복수 개의 프로그램 비트벡터 상호간에 서로 대응되는 비트의 비트값을 모두 합산하여 모든 함수 또는 모든 기본 블록 각각의 빈도수를 나타내는 함수 빈도수 벡터를 생성하는 단계, 함수 빈도수 벡터에 포함된 모든 함수 또는 모든 기본 블록 각각의 빈도수에 기설정된 제1 임계값을 적용하여 비트벡터 마스크를 생성하는 단계 및 복수 개의 프로그램 비트벡터 각각과 비트벡터 마스크를 서로 AND 연산하여 복수 개의 유효 프로그램 비트벡터를 생성하는 단계를 포함한다.
- [0013] 예컨대, 비트벡터 마스크는, 모든 함수 또는 모든 기본 블록 각각의 빈도수 중 기설정된 제1 임계값보다 큰 빈도수에 대응되는 함수 또는 기본 블록에 대한 비트의 비트값을 "1"로 설정하고, 기설정된 제1 임계값보다 작은 빈도수에 대응되는 함수 또는 기본 블록에 대한 비트의 비트값을 "0"으로 설정한 비트벡터이다.
- [0014] 예를 들어, 그룹 특성 벡터 집합이 공집합인 경우, 그룹 특성 벡터 집합을 생성하는 단계는, 복수 개의 유효 프로그램 비트벡터 중 선택된 하나의 유효 프로그램 비트벡터를 그룹 특성 벡터로 결정하여 그룹 특성 벡터 집합에 저장하는 단계를 포함한다.
- [0015] 그룹 특성 벡터 집합이 미리 저장된 적어도 하나의 그룹 특성 벡터를 포함하는 경우, 그룹 특성 벡터 집합을 생성하는 단계는, 적어도 하나의 그룹 특성 벡터 중 i 번째 그룹 특성 벡터와 복수 개의 유효 프로그램 비트벡터 중 j 번째 유효 프로그램 비트벡터 상호 간의 제1 유사도를 산출하는 단계, 제1 유사도에 기설정된 제2 임계값을 적용하여 i 번째 그룹 특성 벡터와 j 번째 유효 프로그램 비트벡터가 서로 유사한지 여부를 판단하는 제1 판단 단계 및 제1 판단 결과에 기초하여 그룹 특성 벡터 집합에 저장될 그룹 특성 벡터를 결정하는 제1 결정 단계를 포함한다.
- [0016] 예컨대, i 번째 그룹 특성 벡터와 j 번째 유효 프로그램 비트벡터가 서로 비유사한 경우, 제1 결정 단계는, j 번째 유효 프로그램 비트벡터를 그룹 특성 벡터로 결정하여 그룹 특성 벡터 집합에 저장하는 단계를 포함한다.
- [0017] 일 실시예에 따라, i 번째 그룹 특성 벡터와 j 번째 유효 프로그램 비트벡터가 서로 유사한 경우, 제1 결정 단계는, i 번째 그룹 특성 벡터와 상기 j 번째 유효 프로그램 비트벡터를 서로 AND 연산하여 공통 비트벡터를 생성하는 단계, i 번째 그룹 특성 벡터와 공통 비트벡터 상호 간의 제2 유사도 및 j 번째 유효 프로그램 비트벡터와 공통 비트벡터 상호 간의 제3 유사도를 각각 산출하는 단계, 제2 유사도 및 제3 유사도 각각에 제2 임계값을 적용하여 i 번째 그룹 특성 벡터와 공통 비트벡터가 서로 유사한지 여부 및 j 번째 유효 프로그램 비트벡터와 공통 비트벡터가 서로 유사한지 여부를 각각 판단하는 제2 판단 단계 및 제2 판단 결과에 기초하여 그룹 특성 벡터 집합에 저장될 그룹 특성 벡터를 결정하는 제2 결정 단계를 포함한다.
- [0018] 예컨대, i 번째 그룹 특성 벡터와 공통 비트벡터가 서로 유사하고, j 번째 유효 프로그램 비트벡터와 공통 비트벡터가 서로 유사한 경우, 제2 결정 단계는, i 번째 그룹 특성 벡터를 그룹 특성 벡터 집합에서 삭제하는 단계 및 공통 비트벡터를 그룹 특성 벡터로 결정하여 그룹 특성 벡터 집합에 저장하는 단계를 포함한다.
- [0019] 예를 들어, i 번째 그룹 특성 벡터와 공통 비트벡터가 서로 비유사하고, j 번째 유효 프로그램 비트벡터와 공통 비트벡터가 서로 유사한 경우, 제2 결정 단계는, 공통 비트벡터를 그룹 특성 벡터로 결정하여 그룹 특성 벡터 집합에 저장하는 단계를 포함한다.
- [0020] 일 실시예에 따르면, i 번째 그룹 특성 벡터와 공통 비트벡터가 서로 유사하고, j 번째 유효 프로그램 비트벡터와 공통 비트벡터가 서로 비유사한 경우, 제2 결정 단계는, i 번째 그룹 특성 벡터를 그룹 특성 벡터 집합에서 삭제하는 단계 및 j 번째 유효 프로그램 비트벡터 및 공통 비트벡터 각각을 그룹 특성 벡터로 결정하여 그룹 특성 벡터 집합에 저장하는 단계를 포함한다.
- [0021] 예를 들어, 적어도 하나의 분류 대상 프로그램을 분류하는 단계는, 복수 개의 그룹 특성 벡터 각각에 대한 적어

도 하나의 분류 대상 프로그램 각각의 포함물을 산출하는 단계 및 포함물에 기설정된 제3 임계값을 적용하여 적어도 하나의 분류 대상 프로그램 각각이 속하는 프로그램 그룹을 결정하는 단계를 포함한다.

[0022] 상기한 목적을 달성하기 위하여 본 발명의 실시예에 따른 프로그램 분류 장치는 미리 저장된 복수 개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록을 나타내는 복수 개의 프로그램 비트벡터를 복수 개의 프로그램 별로 각각 생성하는 비트벡터 생성부, 복수 개의 프로그램 비트벡터에 기초하여 비트벡터 마스크를 생성하고, 복수 개의 프로그램 비트벡터 각각에 비트벡터 마스크를 적용하여 복수 개의 유효 프로그램 비트벡터를 생성하는 유효 비트벡터 생성부, 복수 개의 유효 프로그램 비트벡터에 기설정된 조건을 적용하여 결정된 비트벡터인 복수 개의 그룹 특성 벡터를 포함하는 그룹 특성 벡터 집합을 생성하는 그룹 특성 벡터 집합 생성부 및 미리 저장된 적어도 하나의 분류 대상 프로그램과 복수 개의 그룹 특성 벡터를 비교하여 적어도 하나의 분류 대상 프로그램을 분류하는 프로그램 분류부를 포함한다.

발명의 효과

[0023] 본 발명의 일 실시예에 따르면, 복수 개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록을 나타내는 복수 개의 프로그램 비트벡터를 생성하고, 생성된 프로그램 비트벡터에 비트벡터 마스크를 적용하여 복수 개의 유효 프로그램 비트벡터를 생성한 뒤, 생성된 복수 개의 유효 프로그램 비트벡터에 기초하여 그룹 특성 벡터 집합을 생성함으로써 분류 대상 프로그램이 어떤 그룹에 속하는지 여부를 분류함으로써, 분류 대상 프로그램이 속하는 그룹을 신속하게 결정할 수 있다.

[0024] 더 나아가, 본 발명의 일 실시예에 따르면, 분류 대상 프로그램이 악성코드인 경우, 특정 변종 악성코드가 속하는 그룹을 신속하게 결정함으로써 악성코드를 그룹으로 묶어 체계적으로 관리할 수 있는 효과가 있다.

도면의 간단한 설명

[0025] 도 1은 본 발명의 실시예에 따른 프로그램 분류 장치를 설명하기 위한 구성도이다.

도 2는 본 발명의 실시예에 따른 프로그램 분류 방법을 설명하기 위한 순서도이다.

도 3은 본 발명의 실시예에 따른 프로그램 분류 방법에서 복수 개의 프로그램 비트벡터를 생성하는 방법을 설명하기 위한 순서도이다.

도 4a 및 도 4b는 본 발명의 실시예에 따른 프로그램 분류 방법에서 복수 개의 프로그램 비트벡터를 생성하는 방법을 설명하기 위한 도면이다.

도 5는 본 발명의 실시예에 따른 프로그램 분류 방법에서 복수 개의 유효 프로그램 비트벡터를 생성하는 방법을 설명하기 위한 순서도이다.

도 6a 및 도 6b는 본 발명의 실시예에 따른 프로그램 분류 방법에서 복수 개의 유효 프로그램 비트벡터를 생성하는 방법을 설명하기 위한 도면이다.

도 7 및 도 8은 본 발명의 실시예에 따른 프로그램 분류 방법에서 그룹 특성 벡터 집합을 생성하는 방법을 설명하기 위한 순서도이다.

도 9는 본 발명의 실시예에 따른 프로그램 분류 방법에서 그룹 특성 벡터 집합을 생성하기 위해 제1 유사도, 제2 유사도 및 제3 유사도를 산출하는 방법을 설명하기 위한 도면이다.

도 10은 본 발명의 실시예에 따른 프로그램 분류 방법에서 분류 대상 프로그램을 분류하는 방법을 설명하기 위한 순서도이다.

도 11은 본 발명의 실시예에 따른 프로그램 분류 방법에서 분류 대상 프로그램을 분류하기 위한 포함물을 산출하는 방법을 설명하기 위한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0026] 이하, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 정도로 상세히 설명하기 위하여, 본 발명의 가장 바람직한 실시예를 첨부 도면을 참조하여 설명하기로 한다. 우선 각 도면의 구성요소들에 참조부호를 부가함에 있어서, 동일한 구성요소들에 대해서는 비록 다른 도면 상에 표시되더라도 가능한 한 동일한 부호를 가지도록 하고 있음에 유의해야 한다. 또한, 본 발명을 설명함에 있어, 관련된 공지 구성 또는 기능에 대한 구체적인 설명이 본 발명의 요지를 흐릴 수 있다고 판단되는 경우에

는 그 상세한 설명은 생략한다.

- [0027] 이하, 본 발명의 실시예에 따른 프로그램 분류 방법 및 장치를 첨부된 도면을 참조하여 상세하게 설명하면 아래와 같다.
- [0028] 우선 도 1을 참조하여, 본 발명의 실시예에 따른 프로그램 분류 장치(1000)를 설명한다.
- [0029] 도 1은 본 발명의 실시예에 따른 프로그램 분류 장치를 설명하기 위한 구성도이다.
- [0030] 도 1에 도시된 바와 같이, 본 발명의 실시예에 따른 프로그램 분류 장치(1000)는 비트벡터 생성부(1100), 유효 비트벡터 생성부(1200), 그룹 특성 벡터 집합 생성부(1300) 및 프로그램 분류부(1400)를 포함한다.
- [0031] 비트벡터 생성부(1100)는 미리 저장된 복수 개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록(basic block)을 나타내는 복수 개의 프로그램 비트벡터(bit vector)를 복수 개의 프로그램 별로 각각 생성한다.
- [0032] 유효 비트벡터 생성부(1200)는 복수 개의 프로그램 비트벡터에 기초하여 비트벡터 마스크(mask)를 생성하고, 복수 개의 프로그램 비트벡터 각각에 비트벡터 마스크를 적용하여 복수 개의 유효 프로그램 비트벡터를 생성한다.
- [0033] 일 실시예에 따르면, 유효 프로그램 비트벡터는 상술한 프로그램 비트벡터에 상술한 비트벡터 마스크를 적용하여 생성된 일종의 변형 프로그램 비트벡터를 의미할 수 있다.
- [0034] 그룹 특성 벡터 집합 생성부(1300)는 복수 개의 유효 프로그램 비트벡터에 기설정된 조건을 적용하여 결정된 비트벡터인 복수 개의 그룹 특성 벡터를 포함하는 그룹 특성 벡터 집합을 생성한다.
- [0035] 프로그램 분류부(1400)는 미리 저장된 적어도 하나의 분류 대상 프로그램과 복수 개의 그룹 특성 벡터를 비교하여 적어도 하나의 분류 대상 프로그램을 분류한다.
- [0036] 본 발명의 실시예에 따른 프로그램 분류 장치(1000)의 각각의 구성들에 대한 보다 구체적인 설명은 이하 도 2 내지 도 11을 참조하여 후술하도록 하며, 중복되는 설명은 생략한다.
- [0037] 이제 도 2를 참조하여, 본 발명의 실시예에 따른 프로그램 분류 방법을 설명한다.
- [0038] 도 2는 본 발명의 실시예에 따른 프로그램 분류 방법을 설명하기 위한 순서도이다.
- [0039] 도 2에 도시된 바와 같이, 본 발명의 실시예에 따른 프로그램 분류 방법은 복수 개의 프로그램 비트벡터를 생성하는 단계(S210), 복수 개의 유효 프로그램 비트벡터를 생성하는 단계(S220), 그룹 특성 벡터 집합을 생성하는 단계(S230) 및 분류 대상 프로그램을 분류하는 단계(S240)를 포함한다.
- [0040] S210 단계는, 비트벡터 생성부(1100)가, 미리 저장된 복수 개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록을 나타내는 복수 개의 프로그램 비트벡터를 복수 개의 프로그램 별로 각각 생성하는 단계를 의미한다.
- [0041] 예컨대, 미리 저장된 복수 개의 프로그램 각각은 함수 또는 기본 블록 단위로 구분할 수 있는 프로그램인 어셈블리어 기반 프로그램 또는 고급 언어 기반 프로그램을 의미할 수 있다.
- [0042] 예를 들어, 미리 저장된 복수 개의 프로그램 각각이 고급 언어 중 C언어에 기반한 프로그램인 경우, 미리 저장된 복수 개의 프로그램 각각에 포함된 모든 함수는 "printf("Hello, World!");"과 같이 함수명과 인자값을 모두 포함하는 함수 각각을 의미할 수 있다.
- [0043] 예를 들어, 미리 저장된 복수 개의 프로그램 각각에 포함된 모든 기본 블록은 함수보다 작은 식별 단위로써, 실행 흐름 단위로 함수를 분할한 엔트리 외에는 들어오는 분기가 없고, 출구 외에는 나가는 분기가 없는 직선 코드열을 의미할 수 있다.
- [0044] 예를 들어, 미리 저장된 복수 개의 프로그램 각각이 고급 언어 중 C언어에 기반한 프로그램인 경우, 미리 저장된 복수 개의 프로그램 각각에 포함된 특정 함수의 변수 a가 1 또는 2에서 정의된 경우에서 특정 함수가 "if(a == 1)"인 경우, 특정 함수에 포함된 기본 블록은 a가 1인 경우와 a가 1이 아닌 경우(a가 2인 경우) 각각을 의미하는 2개의 기본 블록을 의미할 수 있다.
- [0045] S220 단계는 유효 비트벡터 생성부(1200)가, 복수 개의 프로그램 비트벡터에 기초하여 비트벡터 마스크를 생성하고, 복수 개의 프로그램 비트벡터 각각에 비트벡터 마스크를 적용하여 복수 개의 유효 프로그램 비트벡터를 생성하는 단계를 의미할 수 있다.

- [0046] 일 실시예에 따르면, S220 단계는 생략될 수 있으며, S220 단계가 생략되는 경우 S230 단계 및 S240 단계는 S210 단계를 통해 생성된 복수 개의 프로그램 비트벡터에 기초하여 수행될 수도 있으나, 본 발명은 이에 한정되지 않는다.
- [0047] S230 단계는 그룹 특성 벡터 집합 생성부(1300)가, 복수 개의 유효 프로그램 비트벡터에 기설정된 조건을 적용하여 결정된 비트벡터인 복수 개의 그룹 특성 벡터를 포함하는 그룹 특성 벡터 집합을 생성하는 단계를 의미할 수 있다.
- [0048] S240 단계는 프로그램 분류부(1400)가, 미리 저장된 적어도 하나의 분류 대상 프로그램과 복수 개의 그룹 특성 벡터를 비교하여 적어도 하나의 분류 대상 프로그램을 분류하는 단계를 의미할 수 있다.
- [0049] 이제 도 3, 도 4a 및 도 4b를 동시에 참조하여 비트벡터 생성부(1100)에 의해 수행되는 S210 단계를 설명한다.
- [0050] 도 3은 본 발명의 실시예에 따른 프로그램 분류 방법에서 복수 개의 프로그램 비트벡터를 생성하는 방법을 설명하기 위한 순서도이며, 도 4a 및 도 4b는 본 발명의 실시예에 따른 프로그램 분류 방법에서 복수 개의 프로그램 비트벡터를 생성하는 방법을 설명하기 위한 도면이다.
- [0051] 도 3에 도시된 바와 같이, S210 단계는 모든 함수 또는 모든 기본 블록을 임의의 비트열로 변환하는 단계(S211), 임의의 비트열 각각에서 연속 비트열을 추출하는 단계(S213), 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호를 생성하는 단계(S215) 및 복수의 비트벡터 상에서 식별 번호에 대응되는 비트의 비트값을 설정하여 복수 개의 프로그램 비트벡터를 각각 생성하는 단계(S217)를 포함한다.
- [0052] S211 단계는 복수 개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록 각각에 해시함수를 적용하여 모든 함수 또는 모든 기본 블록 각각을 임의의 비트열로 변환하는 단계를 의미할 수 있다.
- [0053] 예컨대, 해시함수는 그 크기가 일정하되 임의의 값을 가지는 출력값을 생성하는 함수로, 해시함수에 동일한 입력값을 넣는 경우, 항상 같은 출력값이 나오지만 출력값만을 이용해서 입력값을 역산할 수는 없는 특징이 있다.
- [0054] 이때, 해시함수는 메시지 다이제스트(Message Digest), 시큐어 해시 알고리즘(Secure Hash Algorithm), HAS-160 및 SHA-256을 비롯한 각종 해시함수가 사용될 수 있으며, 본 발명은 특정 해시함수의 종류에 한정되지 않는다.
- [0055] 예를 들어, 도 4a에 도시된 바와 같이 특정 프로그램(1)에 포함된 모든 함수 중 하나의 함수인 함수 1에 해시함수를 적용하면 함수 1에 대응되는 해시값이 생성될 수 있으며, 이때 생성된 해시값은 임의의 비트열로 정의할 수 있다.
- [0056] 예를 들어, 도 4a에 도시된 바와 같이 특정 프로그램(1)에 포함된 특정 함수인 함수 1에 해시함수를 적용하는 경우를 예로 들어 S211 단계를 설명하면, S211 단계의 연산 결과값으로 "0101010110101101101010101010101001101010101010101010101010101010"와 같은 해시값이 생성될 수 있으며, 이때 생성된 해시값을 임의의 비트열이라 할 수 있다.
- [0057] 이때, 복수 개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록 각각에 해시함수를 적용하면, 복수 개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록 각각은 임의의 비트열로 표현될 수 있다.
- [0058] S213 단계는 모든 함수 또는 모든 기본 블록 각각에 대응되는 임의의 비트열 각각에서 기설정된 개수의 연속된 비트열인 연속 비트열을 각각 추출하는 단계를 의미할 수 있다.
- [0059] 예를 들어, 도 4a에 도시된 바와 같이, 임의의 비트열인 "0101010110101101101010101010101001101010101010101010101010101010"의 최초의 비트값인 "0"을 0번째 비트라 정의할 때, S213 단계에서 24번째 내지 39번째로 연속되는 16개의 비트를 연속 비트열로 추출한 결과 "0101010110110100"이 추출될 수 있다.
- [0060] 이때, S213 단계에서 추출되는 연속된 비트열의 기설정된 개수는 미리 설정되어 비트벡터 생성부(1100)에 저장될 수 있으며, 본 발명은 특정 개수의 비트열을 추출하는 것으로 한정되지 않는다.
- [0061] 일 실시예에 따르면, S213 단계에서 비트벡터 생성부(1100)는 기설정된 개수의 연속된 비트열을 추출하는 대신에, 기설정된 개수의 불연속한 비트열을 추출할 수도 있으며, 본 발명은 연속된 비트열을 추출하는 것으로 한정되지 않는다.
- [0062] S215 단계는 추출된 연속 비트열 각각을 기설정된 조건에 따라 변환하여 모든 함수 또는 모든 기본 블록 각각에

대응되는 식별 번호를 생성하는 단계를 의미할 수 있다.

- [0063] 예컨대, S215 단계는 추출된 연속 비트열 각각을 10진수로 변환하여 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호를 생성하는 단계를 의미할 수도 있으나, 본 발명은 이에 한정되지 않는다.
- [0064] 예컨대, 도 4a에 도시된 바와 같이 연속 비트열 "0101010110110100"에 대응되는 10진수인 21,940를 S211, S213 및 S215 단계를 거쳐 특정 함수인 함수 1에 대응되는 식별 번호로 생성할 수 있다.
- [0065] 다시 말해, 복수 개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록 각각에 S211 단계, S213 단계 및 S215 단계를 각각 적용하면, 복수 개의 프로그램 각각은 복수 개의 식별 번호로 표현될 수 있게 된다.
- [0066] S217 단계는 복수 개의 프로그램 각각에 대응되며 기설정된 개수의 비트를 포함하는 복수 개의 비트벡터 상에서 식별 번호에 각각 대응되는 비트의 비트값을 설정하여 복수 개의 프로그램 비트벡터를 각각 생성하는 단계를 의미할 수 있다.
- [0067] 예컨대, 복수 개의 비트벡터는 각각이 포함하는 모든 비트의 비트값이 "0"으로 초기화된 상태일 수 있다.
- [0068] 도 4a에 도시된 상술한 예시에서 특정 프로그램(1)에 포함된 함수인 함수 1에 대응되는 식별 번호는 21,940으로 생성되었으며, S217 단계에서 비트벡터 생성부(1100)는 특정 프로그램(1)에 대응되는 특정 비트벡터(2) 상에서 21,940 번째 비트의 비트값을 "1"로 설정할 수 있으며, 이러한 과정은 도 4b에 도시된 바와 같다.
- [0069] S217 단계에서 비트벡터 생성부(1100)가 특정 프로그램(1)에 포함된 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호에 대응되는 비트의 비트값을 "1"로 설정하면, 특정 프로그램(1)에 대응되는 프로그램 비트벡터가 생성될 수 있다.
- [0070] 상기한 S211 단계, S213 단계, S215 단계 및 S217 단계의 과정을 복수 개의 프로그램 각각에 대하여 수행하면, 복수 개의 프로그램 각각에 대응되는 복수 개의 프로그램 비트벡터가 생성될 수 있다.
- [0071] 상기한 S211 단계, S213 단계, S215 단계 및 S217 단계를 거쳐 생성된 복수 개의 프로그램 비트벡터는 복수 개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록의 존재 여부를 나타내는 비트벡터를 의미할 수 있다.
- [0072] 예를 들어, 특정 프로그램 비트벡터의 1번째 비트, 5번째 비트 및 7번째 비트의 비트값이 "1"인 경우, 특정 프로그램은 식별번호 1, 식별번호 5 및 식별번호 7에 대응되는 함수 또는 기본 블록을 포함함을 의미하게 된다.
- [0073] 이제, 도 5, 도 6a 및 도 6b를 참조하여, 유효 비트벡터 생성부(1200)에 의해 수행되는 S220 단계를 설명한다.
- [0074] 도 5는 본 발명의 실시예에 따른 프로그램 분류 방법에서 복수 개의 유효 프로그램 비트벡터를 생성하는 방법을 설명하기 위한 순서도이며, 도 6a 및 도 6b는 본 발명의 실시예에 따른 프로그램 분류 방법에서 복수 개의 유효 프로그램 비트벡터를 생성하는 방법을 설명하기 위한 도면이다.
- [0075] 도 5에 도시된 바와 같이, S220 단계는 함수 빈도수 벡터를 생성하는 단계(S221), 함수 빈도수 벡터에 제1 임계값을 적용하여 비트벡터 마스크를 생성하는 단계(S223) 및 복수 개의 프로그램 비트벡터와 비트벡터 마스크를 서로 AND 연산하여 복수 개의 유효 프로그램 비트벡터를 생성하는 단계(S225)를 포함한다.
- [0076] S221 단계는, 복수 개의 프로그램 비트벡터 상호간에 서로 대응되는 비트의 비트값을 모두 합산하여 모든 함수 또는 모든 기본 블록 각각의 빈도수를 나타내는 함수 빈도수 벡터를 생성하는 단계를 의미할 수 있다.
- [0077] 상술한 바와 같이, S210 단계를 통해 생성된 복수 개의 프로그램 비트벡터 각각은 대응되는 복수 개의 프로그램에 포함된 모든 함수 또는 모든 기본 블록을 나타내게 되며, 복수 개의 프로그램 비트벡터 상호간에 서로 대응되는 비트의 비트값을 모두 합산하여 생성된 함수 빈도수 벡터에 포함된 각각의 블록들은 대응되는 함수 또는 기본 블록의 출현 빈도수를 의미하게 된다.
- [0078] 예를 들어 도 6a에 도시된 바와 같이, 복수 개의 프로그램 비트벡터(61, 62, 63)의 서로 대응되는 비트의 비트값을 모두 합산하여 생성된 함수 빈도수 벡터(64)의 각각의 값은 복수 개의 프로그램 상에서 동일한 식별 번호를 가지는 함수 또는 기본 블록의 출현 빈도수를 각각 의미하게 된다.
- [0079] 예를 들어 도 6a에 도시된 바와 같이, 제1 프로그램 비트벡터(61)의 i 번째 비트의 비트값이 "1"이고, $i+1$ 번째 비트의 비트값이 "0"이며, j 번째 비트의 비트값이 "0"이고, $j+1$ 번째 비트의 비트값이 "1"이고, $j+2$ 번째 비트의 비트값이 "1"인 경우를 예로 든다.

- [0080] 상술한 예시에서, 제2 프로그램 비트벡터(62)는 i 번째 비트의 비트값이 "0"이고, $i+1$ 번째 비트의 비트값이 "1"이며, j 번째 비트의 비트값이 "0"이고, $j+1$ 번째 비트의 비트값이 "1"이고, $j+2$ 번째 비트의 비트값이 "1"이라 할 수 있다.
- [0081] 마찬가지로, 제 n 프로그램 비트벡터(63)는 i 번째 비트의 비트값이 "0"이고, $i+1$ 번째 비트의 비트값이 "0"이며, j 번째 비트의 비트값이 "1"이고, $j+1$ 번째 비트의 비트값이 "0"이고, $j+2$ 번째 비트의 비트값이 "0"이라 할 수 있다.
- [0082] 이 경우, 제1 번째 프로그램부터 제 n 번째 프로그램까지의 복수 개의 프로그램(61, 62, 63)의 i 번째 비트의 비트값을 모두 합산하여 함수 빈도수 벡터(64) 상의 i 번째 블록의 빈도수가 "9"로 생성되었으며, 이는 복수 개의 프로그램(61, 62, 63)에서 식별 번호 i 에 대응되는 함수 또는 기본 블록이 총 9회 출현하였음을 의미하게 된다.
- [0083] 마찬가지로, 제1 번째 프로그램부터 제 n 번째 프로그램까지의 복수 개의 프로그램(61, 62, 63)의 $i+1$ 번째 비트의 비트값을 모두 합산하여 함수 빈도수 벡터(64) 상의 $i+1$ 번째 블록의 빈도수가 "15"로 생성되었으며, 이는 복수 개의 프로그램(61, 62, 63)에서 식별 번호 $i+1$ 에 대응되는 함수 또는 기본 블록이 총 15회 출현하였음을 의미하게 된다.
- [0084] 동일하게, 제1 번째 프로그램부터 제 n 번째 프로그램까지의 복수 개의 프로그램(61, 62, 63)의 j 번째 비트의 비트값을 모두 합산하여 함수 빈도수 벡터(64) 상의 j 번째 블록의 빈도수가 "7"로 생성되었으며, 이는 복수 개의 프로그램(61, 62, 63)에서 식별 번호 j 에 대응되는 함수 또는 기본 블록이 총 7회 출현하였음을 의미하게 된다.
- [0085] 또한, 제1 번째 프로그램부터 제 n 번째 프로그램까지의 복수 개의 프로그램(61, 62, 63)의 $j+1$ 번째 비트의 비트값을 모두 합산하여 함수 빈도수 벡터(64) 상의 $j+1$ 번째 블록의 빈도수가 "33"으로 생성되었으며, 이는 복수 개의 프로그램(61, 62, 63)에서 식별 번호 $j+1$ 에 대응되는 함수 또는 기본 블록이 총 33회 출현하였음을 의미하게 된다.
- [0086] 마찬가지로, 제1 번째 프로그램부터 제 n 번째 프로그램까지의 복수 개의 프로그램(61, 62, 63)의 $j+2$ 번째 비트의 비트값을 모두 합산하여 함수 빈도수 벡터(64) 상의 $j+2$ 번째 블록의 빈도수가 "30"으로 생성되었으며, 이는 복수 개의 프로그램(61, 62, 63)에서 식별 번호 $j+2$ 에 대응되는 함수 또는 기본 블록이 총 30회 출현하였음을 의미하게 된다.
- [0087] 즉, S221 단계를 거쳐 생성된 함수 빈도수 벡터(64)은 복수 개의 프로그램에 포함된 각각의 함수 또는 기본 블록 별로 그 출현 빈도수를 나타내는 일종의 테이블을 의미할 수도 있으나, 본 발명은 이에 한정되지 않는다.
- [0088] S223 단계는 함수 빈도수 벡터에 포함된 모든 함수 또는 모든 기본 블록 각각의 빈도수에 기설정된 제1 임계값을 적용하여 비트벡터 마스크를 생성하는 단계를 의미한다.
- [0089] 예를 들어, 도 6a에 도시된 바와 같이, 함수 빈도수 벡터(64)의 i 번째 블록은 "9"이고, $i+1$ 번째 블록은 "15"이고, j 번째 블록은 "7"이고, $j+1$ 번째 블록은 "33"이며, $j+2$ 번째 블록은 "30"이고, 기설정된 제1 임계값은 10인 경우를 예로 들면, 함수 빈도수 벡터(64)에 기설정된 제1 임계값을 적용하여 생성된 비트벡터 마스크(65)는 i 번째 비트의 비트값이 "0"이고, $i+1$ 번째 비트의 비트값이 "1"이고, j 번째 비트의 비트값이 "0"이고, $j+1$ 번째 비트의 비트값이 "1"이고, $j+2$ 번째 비트의 비트값이 "1"인 비트벡터로써 생성되게 된다.
- [0090] 이때, 기설정된 제1 임계값이 10인 것은 일 실시예로써 본 발명의 기설정된 제1 임계값은 특정 수치로 한정되지 않는다.
- [0091] 다시 말해, 비트벡터 마스크(65)의 각각의 비트는 복수 개의 프로그램 상에서 기설정된 제1 임계값 이상 출현한 함수 또는 기본 블록은 "1"로 설정하고, 복수 개의 프로그램 상에서 기설정된 제1 임계값 미만으로 출현한 함수 또는 기본 블록은 "0"으로 설정하는 비트벡터를 의미할 수 있다.
- [0092] 즉, 비트벡터 마스크(65)는, 모든 함수 또는 모든 기본 블록 각각의 빈도수 중 기설정된 제1 임계값보다 큰 빈도수에 대응되는 함수 또는 기본 블록에 대한 비트의 비트값을 "1"로 설정하고, 기설정된 제1 임계값보다 작은 빈도수에 대응되는 함수 또는 기본 블록에 대한 비트의 비트값을 "0"으로 설정한 비트벡터를 의미할 수 있다.
- [0093] 예를 들어, 도 6a에 도시된 비트벡터 마스크(65)는 복수 개의 프로그램 비트벡터(61, 62, 63)상에서 식별 번호 i 에 대응되는 함수 또는 기본 블록은 10회 미만으로 출현하였고, 식별 번호 $i+1$ 에 대응되는 함수 또는 기본 블록은 10회 이상 출현하였으며, 식별 번호 j 에 대응되는 함수 또는 기본 블록은 10회 미만으로 출현하였고, 식별 번호 $j+1$ 에 대응되는 함수 또는 기본 블록은 10회 이상 출현하였으며, 식별 번호 $j+2$ 에 대응되는 함수 또는 기

본 블록은 10회 이상 출현하였음을 의미하게 된다.

- [0094] 이때, 복수 개의 프로그램 상에서 기설정된 횟수 이상 출현하여, 비트벡터 마스크(65) 상에서 "1"로 표현되는 함수 또는 기본 블록을 공통 함수 또는 공통 기본 블록으로 정의할 수 있다.
- [0095] S225 단계는, 복수 개의 프로그램 비트벡터 각각과 비트벡터 마스크를 서로 AND 연산하여 복수 개의 유효 프로그램 비트벡터를 생성하는 단계를 의미할 수 있다.
- [0096] 예를 들어, 도 6b에 도시된 바와 같이, 복수 개의 프로그램 비트벡터 중 i 번째 프로그램 비트벡터인 제 i 프로그램 비트벡터(66)의 i 번째 비트의 비트값이 "1"이고, $i+1$ 번째 비트의 비트값이 "1"이며, j 번째 비트의 비트값이 "1"이고, $j+1$ 번째 비트의 비트값이 "1"이고, $j+2$ 번째 비트의 비트값이 "1"이며, 비트벡터 마스크(65)의 i 번째 비트의 비트값이 "0"이고, $i+1$ 번째 비트의 비트값이 "1"이며, j 번째 비트의 비트값이 "0"이고, $j+1$ 번째 비트의 비트값이 "1"이고, $j+2$ 번째 비트의 비트값이 "1"인 경우, 제 i 유효 프로그램 비트벡터(67)는 i 번째 비트의 비트값이 "0"이고, $i+1$ 번째 비트의 비트값이 "1"이며, j 번째 비트의 비트값이 "0"이고, $j+1$ 번째 비트의 비트값이 "1"이고, $j+2$ 번째 비트의 비트값이 "1"로 결정될 수 있다.
- [0097] 이는, AND 연산의 특징에 기인한 것으로, 제 i 프로그램 비트벡터(66)와 비트벡터 마스크(65)의 서로 대응되는 비트의 비트값이 모두 "1"인 경우 제 i 유효 프로그램 비트벡터(67)의 해당되는 비트의 비트값은 "1"로 설정되고, 제 i 프로그램 비트벡터(66)와 비트벡터 마스크(65)의 서로 대응되는 비트의 비트값 중 어느 하나가 "1"이 아닌 경우, 제 i 유효 프로그램 비트벡터(67)의 해당되는 비트의 비트값은 "0"로 설정되기 때문이다.
- [0098] 그 결과, 제 i 유효 프로그램 비트벡터(67)는 제 i 프로그램에 포함된 모든 함수 또는 모든 기본 블록 중 비트벡터 마스크(65)를 통해 정의되는 공통함수 만을 포함하도록 설정되게 된다.
- [0099] 상술한 과정을 복수 개의 프로그램 비트벡터 각각에 대하여 수행하면, 복수 개의 프로그램에 대응되는 복수 개의 유효 프로그램 비트벡터가 생성될 수 있다.
- [0100] 이제 도 7 및 도 8을 참조하여, 그룹 특성 벡터 집합 생성부(1300)에 의해 수행되는 S230 단계를 설명한다.
- [0101] 예컨대, S230 단계는 그룹 특성 벡터 집합이 공집합인 경우와, 공집합이 아닌 경우로 나누어 설명될 수 있으며, 이하 그룹 특성 벡터 집합이 공집합인 경우와 공집합이 아닌 경우로 나누어 S230 단계를 설명한다.
- [0102] 다시 말해, S230 단계는, 그 집합이 공집합인 경우와 그렇지 않은 경우로 구분하여 그룹 특성 벡터의 추가와 제거를 반복할 수 있다.
- [0103] 이때, 그룹 특성 벡터는 적어도 하나의 분류 대상 프로그램을 분류하기 위한 복수의 프로그램 그룹 각각의 특성을 나타내는 비트벡터를 의미할 수 있으며, 그룹 특성 벡터 집합은 적어도 하나의 그룹 특성 벡터를 포함하는 집합을 의미할 수 있다.
- [0104] 예를 들어, S230 단계에서 그룹 특성 벡터 집합은 초기에 공집합으로 정의될 수 있으며, 그룹 특성 벡터 집합이 공집합인 경우, S230 단계는 복수 개의 유효 프로그램 비트벡터 중 선택된 하나의 유효 프로그램 비트벡터를 그룹 특성 벡터로 결정하여 그룹 특성 벡터 집합에 저장하는 단계를 포함할 수 있다.
- [0105] 즉, 그룹 특성 벡터 집합이 공집합인 경우, 임의로 선택된 최초의 1개의 유효 프로그램 비트벡터는 그 자체가 그룹 특성 벡터로써 결정되어 그룹 특성 벡터 집합에 저장될 수 있다.
- [0106] 이제, 도 7 및 도 8을 참조하여 그룹 특성 벡터 집합에 적어도 하나의 그룹 특성 벡터가 저장되어 그룹 특성 벡터 집합이 공집합이 아닌 경우에서 S230 단계를 설명한다.
- [0107] 도 7에 도시된 바와 같이, 그룹 특성 벡터 집합이 미리 저장된 적어도 하나의 그룹 특성 벡터를 포함하는 경우, S230 단계는 S710 단계, S720 단계 및 S730 단계를 포함할 수 있다.
- [0108] S710 단계는, 적어도 하나의 그룹 특성 벡터 중 i 번째 그룹 특성 벡터와 복수 개의 유효 프로그램 비트벡터 중 j 번째 유효 프로그램 비트벡터 상호 간의 제1 유사도를 산출하는 단계를 의미할 수 있다.
- [0109] 이때, 제1 유사도를 산출하는 보다 구체적인 방법은 이하 도 9를 참조하여 후술하도록 하며, 중복되는 설명은 생략한다.
- [0110] S720 단계는 제1 유사도에 기설정된 제2 임계값을 적용하여 i 번째 그룹 특성 벡터와 j 번째 유효 프로그램 비트벡터가 서로 유사한지 여부를 판단하는 제1 판단 단계를 의미할 수 있다.

- [0111] 보다 상세하게, S720 단계에서 제1 유사도가 제2 임계값보다 크거나 같은 경우 i 번째 그룹 특성 벡터와 j 번째 유효 프로그램 비트벡터는 서로 유사한 것으로 판단될 수 있고, 제1 유사도가 제2 임계값보다 작은 경우 i 번째 그룹 특성 벡터와 j 번째 유효 프로그램 비트벡터는 서로 유사하지 않은 것으로 판단될 수 있다.
- [0112] 예를 들어, 제1 유사도가 40%이며 기설정된 제2 임계값이 70%인 경우, 제1 유사도는 제2 임계값보다 작기 때문에 i 번째 그룹 특성 벡터와 j 번째 유효 프로그램 비트벡터는 서로 유사하지 않은 것으로 판단될 수 있다.
- [0113] 그 뒤, S230 단계는 상술한 S720 단계에서의 제1 판단 결과에 기초하여 그룹 특성 벡터 집합에 저장될 그룹 특성 벡터를 결정하는 제1 결정 단계(미도시)를 포함할 수 있다.
- [0114] 만약, S720 단계에서 판단 결과, 제1 유사도가 제2 임계값보다 작아 i 번째 그룹 특성 벡터와 j 번째 유효 프로그램 비트벡터가 서로 비유사한 경우, 제1 결정 단계(미도시)는, j 번째 유효 프로그램 비트벡터를 그룹 특성 벡터로 결정하여 그룹 특성 벡터 집합에 저장하는 단계(S730)를 포함할 수 있다.
- [0115] 다시 말해, S730 단계에서 그룹 특성 벡터 집합에 이미 저장된 i 번째 그룹 특성 벡터와 j 번째 유효 프로그램 비트벡터가 서로 비유사하다면, j 번째 유효 프로그램 비트벡터는 프로그램을 분류하기 위한 프로그램 그룹의 특성 벡터로써 활용될 수 있기 때문에, j 번째 유효 프로그램 비트벡터는 그룹 특성 벡터로 결정되어 그룹 특성 벡터 집합에 저장될 수 있다.
- [0116] 한편, S720 단계에서 판단 결과, 제1 유사도가 제2 임계값보다 크거나 같아 i 번째 그룹 특성 벡터와 j 번째 유효 프로그램 비트벡터가 서로 유사한 경우 도 8에 도시된 S810 단계가 수행될 수 있다.
- [0117] 도 8에 도시된 바와 같이, S720 단계의 판단 결과 i 번째 그룹 특성 벡터와 j 번째 유효 프로그램 비트벡터가 서로 유사한 경우, 제1 결정 단계(미도시)는, 아래와 같은 S810 단계, S820 단계, S830 단계, S840 단계 및 S850 단계를 포함할 수 있다.
- [0118] S810 단계는 i 번째 그룹 특성 벡터와 j 번째 유효 프로그램 비트벡터를 서로 AND 연산하여 공통 비트벡터를 생성하는 단계를 의미할 수 있다.
- [0119] 다시 말해, 공통 비트벡터는 그룹 특성 벡터와 유효 프로그램 비트벡터를 AND 연산하여 생성된 비트벡터를 의미할 수 있다.
- [0120] 이때, 두 개의 비트벡터를 서로 AND 연산하는 구체적인 방법은 상기 도 6b에서 설명된 바와 같으므로 그 구체적인 설명은 생략한다.
- [0121] S820 단계는, i 번째 그룹 특성 벡터와 공통 비트벡터 상호 간의 제2 유사도 및 j 번째 유효 프로그램 비트벡터와 공통 비트벡터 상호 간의 제3 유사도를 각각 산출하는 단계를 의미할 수 있다.
- [0122] 이때, 제2 유사도 및 제3 유사도를 산출하는 보다 구체적인 방법은 이하 도 9를 참조하여 후술하도록 하며, 중복되는 설명은 생략한다.
- [0123] S830 단계, S840 단계 및 S850 단계는 제2 유사도 및 제3 유사도 각각에 제2 임계값을 적용하여 i 번째 그룹 특성 벡터와 공통 비트벡터가 서로 유사한지 여부 및 j 번째 유효 프로그램 비트벡터와 공통 비트벡터가 서로 유사한지 여부를 각각 판단하는 제2 판단 단계를 의미할 수 있다.
- [0124] 예를 들어, S830 단계에서 판단 결과 제2 유사도가 제2 임계값보다 크거나 같아 i 번째 그룹 특성 벡터와 공통 비트벡터가 서로 유사한 것으로 판단되는 경우 S840 단계가 수행될 수 있다.
- [0125] 반면, S830 단계에서 판단 결과 제2 유사도가 제2 임계값보다 작아 i 번째 그룹 특성 벡터와 공통 비트벡터가 서로 유사하지 않은 것으로 판단되는 경우 S850 단계가 수행될 수 있다.
- [0126] 예컨대, S840 단계에서 판단 결과 제3 유사도가 제2 임계값보다 크거나 같아 j 번째 유효 프로그램 비트벡터와 공통 비트벡터가 서로 유사한 것으로 판단되는 경우 S841 단계가 수행될 수 있다.
- [0127] 반면, S840 단계에서 판단 결과 제3 유사도가 제2 임계값보다 작아 j 번째 유효 프로그램 비트벡터와 공통 비트벡터가 서로 유사하지 않은 것으로 판단되는 경우 S845 단계가 수행될 수 있다.
- [0128] 마찬가지로, S850 단계에서 판단 결과 제3 유사도가 제2 임계값보다 크거나 같아 j 번째 유효 프로그램 비트벡터와 공통 비트벡터가 서로 유사한 것으로 판단되는 경우 S851 단계가 수행될 수 있다.
- [0129] 이때, 제2 유사도 및 제3 유사도 각각에 적용되는 제2 임계값은 상기한 제1 유사도에 적용되는 제2 임계값과 서

로 동일한 임계값일 수 있다.

- [0130] 그 뒤, S230 단계는 제2 판단 결과에 기초하여 그룹 특성 벡터 집합에 저장될 그룹 특성 벡터를 결정하는 제2 결정 단계(미도시)를 포함할 수 있다.
- [0131] S830 단계의 판단 결과, i 번째 그룹 특성 벡터와 공통 비트벡터가 서로 유사하고, S840 단계의 판단 결과 j 번째 유효 프로그램 비트벡터와 공통 비트벡터가 서로 유사한 경우, 제2 결정 단계(미도시)는, 아래 S841 단계 및 S843 단계를 포함할 수 있다.
- [0132] S841 단계는 i 번째 그룹 특성 벡터를 그룹 특성 벡터 집합에서 삭제하는 단계를 의미할 수 있다.
- [0133] S843 단계는 공통 비트벡터를 그룹 특성 벡터로 결정하여 그룹 특성 벡터 집합에 저장하는 단계를 의미할 수 있다.
- [0134] S841 단계 및 S843 단계의 수행 결과, 공통 비트벡터와 유사한 i 번째 그룹 특성 벡터는 그룹 특성 벡터 집합에서 삭제되고, 공통 비트벡터와 유사한 j 번째 유효 프로그램 비트벡터는 그룹 특성 벡터 집합에 저장되지 않기 때문에, 그룹 특성 벡터 집합에는 공통 비트벡터만이 그룹 특성 벡터로 결정되어 저장되게 된다.
- [0135] S830 단계의 판단 결과, i 번째 그룹 특성 벡터와 공통 비트벡터가 서로 비유사하고, S850 단계의 판단 결과, j 번째 유효 프로그램 비트벡터와 공통 비트벡터가 서로 유사한 경우, 제2 결정 단계(미도시)는, 아래 S851 단계를 포함할 수 있다.
- [0136] S851 단계는 공통 비트벡터를 그룹 특성 벡터로 결정하여 상기 그룹 특성 벡터 집합에 저장하는 단계를 의미할 수 있다.
- [0137] S851 단계의 수행 결과, 공통 비트벡터와 비유사한 i 번째 그룹 특성 벡터는 그룹 특성 벡터 집합에 남아있고, 공통 비트벡터와 유사한 j 번째 유효 프로그램 비트벡터는 그룹 특성 벡터 집합에 저장되지 않기 때문에, 그룹 특성 벡터 집합에는 i 번째 그룹 특성 벡터가 저장되며, 공통 비트벡터가 그룹 특성 벡터로 결정되어 저장되게 된다.
- [0138] S830 단계의 판단 결과, i 번째 그룹 특성 벡터와 공통 비트벡터가 서로 유사하고, S840 단계의 판단 결과, j 번째 유효 프로그램 비트벡터와 공통 비트벡터가 서로 비유사한 경우, 제2 결정 단계(미도시)는, 아래 S845 단계 및 S847 단계를 포함할 수 있다.
- [0139] S845 단계는 i 번째 그룹 특성 벡터를 그룹 특성 벡터 집합에서 삭제하는 단계를 의미할 수 있다.
- [0140] S847 단계는 j 번째 유효 프로그램 비트벡터 및 공통 비트벡터 각각을 그룹 특성 벡터로 결정하여 그룹 특성 벡터 집합에 저장하는 단계를 의미할 수 있다.
- [0141] S845 단계 및 S847 단계의 수행 결과, 공통 비트벡터와 유사한 i 번째 그룹 특성 벡터는 그룹 특성 벡터 집합에서 삭제되고, 공통 비트벡터와 비유사한 j 번째 유효 프로그램 비트벡터는 그룹 특성 벡터로 결정되어 그룹 특성 벡터 집합에 저장되기 때문에, 그룹 특성 벡터 집합에는 공통 비트벡터 및 j 번째 유효 프로그램 비트벡터 각각이 그룹 특성 벡터로 결정되어 저장되게 된다.
- [0142] 상기한 바와 같이, S230 단계에서 복수 개의 그룹 특성 벡터는 서로 유사하지 않도록 결정되어 그룹 특성 벡터 집합에 저장되기 때문에, 각각의 그룹 특성 벡터는 S240 단계에서 프로그램을 분류하기 위한 프로그램 그룹을 대표하는 특성 벡터로써 기능하게 된다.
- [0143] 이제 도 9를 참조하여, S230 단계에서 그룹 특성 벡터를 결정하기 위해 활용하는 제1 유사도, 제2 유사도 및 제3 유사도를 산출하는 방법을 설명한다.
- [0144] 도 9는 본 발명의 실시예에 따른 프로그램 분류 방법에서 그룹 특성 벡터 집합을 생성하기 위해 제1 유사도, 제2 유사도 및 제3 유사도를 산출하는 방법을 설명하기 위한 도면이다.
- [0145] 예를 들어, S230 단계에서 그룹 특성 벡터 집합 생성부(1300)는 상기한 제1 유사도, 제2 유사도 및 제3 유사도를 산출하기 위하여, 서로 다른 두 개의 비트벡터의 서로 대응되는 비트의 비트값을 서로 비교할 수 있다.
- [0146] 예를 들어, 그룹 특성 벡터 집합 생성부(1300)가 i 번째 그룹 특성 벡터와 j 번째 유효 프로그램 비트벡터의 서로 대응되는 비트를 비교하는 경우 제1 유사도가 산출될 수 있다.
- [0147] 예컨대, 그룹 특성 벡터 집합 생성부(1300)가 i 번째 그룹 특성 벡터와 공통 비트벡터의 서로 대응되는 비트를

비교하는 경우 제2 유사도가 산출될 수 있다.

- [0148] 예컨대, 그룹 특성 벡터 집합 생성부(1300)가 j 번째 유효 프로그램 비트벡터와 공통 비트벡터의 서로 대응되는 비트를 비교하는 경우 제3 유사도가 산출될 수 있다.
- [0149] 계속 도 9를 참조하여, 제1 유사도, 제2 유사도 및 제3 유사도를 산출하는 방법을 설명하기 위하여, 두 개의 비트벡터 간의 유사도를 산출하는 방법을 설명한다.
- [0150] 도 9에 도시된 바와 같이, 제1 비트벡터(91)는 i 번째 비트의 비트값이 "1", i+1 번째 비트의 비트값이 "1", j 번째 비트의 비트값이 "1", j+1 번째 비트의 비트값이 "1", j+2 번째 비트의 비트값이 "0"인 비트벡터로 "...11...110..."과 같이 표현될 수 있다.
- [0151] 마찬가지로, 제2 비트벡터(92)는 i 번째 비트의 비트값이 "0", i+1 번째 비트의 비트값이 "1", j 번째 비트의 비트값이 "0", j+1 번째 비트의 비트값이 "1", j+2 번째 비트의 비트값이 "1"인 비트벡터로 "...01...011..."과 같이 표현될 수 있다.
- [0152] 이 경우, S230 단계에서 그룹 특성 벡터 집합 생성부(1300)는 제1 유사도, 제2 유사도 또는 제3 유사도를 각각 산출하기 위하여, 제1 비트벡터(91) 및 제2 비트벡터(92)를 서로 대응되는 비트 별로 비교하여, 서로 일치하는 비트값을 가지는 비트의 개수에 기초하여 제1 비트벡터(91) 및 제2 비트벡터(92) 상호간의 유사도를 산출할 수 있다.
- [0153] 보다 상세하게, S230 단계에서 그룹 특성 벡터 집합 생성부(1300)는 제1 유사도, 제2 유사도 및 제3 유사도를 산출하기 위하여, 제1 비트벡터(91) 및 제2 비트벡터(92)에서 서로 대응되는 비트의 비트값이 모두 "1"인 비트의 개수를 서로 대응되는 비트의 비트값 중 적어도 하나가 "1"인 비트의 개수로 나누어 두 개의 비트벡터(91, 92) 상호간의 유사도를 산출할 수 있다.
- [0154] 예를 들어, 제1 비트벡터(91) 및 제2 비트벡터(92) 각각의 i번째 비트, i+1번째 비트, j번째 비트, j+1번째 비트, j+2번째 비트에 기초하여, 제1 비트벡터(91) 및 제2 비트벡터(92) 상호간의 유사도를 산출하면, 제1 비트벡터(91) 및 제2 비트벡터(92)에서 서로 대응되는 비트의 비트값이 모두 "1"인 비트는 i+1번째 비트 및 j+1번째 비트이며, 그 개수는 2개임을 확인할 수 있다.
- [0155] 한편, 제1 비트벡터(91) 및 제2 비트벡터(92)에서 서로 대응되는 비트의 비트값 중 적어도 하나가 "1"인 비트는 i번째 비트, i+1번째 비트, j번째 비트, j+1번째 비트, j+2번째 비트이며, 그 개수는 5개임을 확인할 수 있다.
- [0156] 그 결과, 제1 비트벡터(91) 및 제2 비트벡터(92) 상호간의 유사도는 2/5로 산출될 수 있다.
- [0157] 예를 들어 상술한 유사도 산출 방법은, 제1 비트벡터(91)와 제2 비트벡터(92)에서 "1"의 비트값을 가지는 비트의 교집합의 개수에서 "1"의 비트값을 가지는 비트의 합집합의 개수를 나누는 일종의 변형된 자카드 인덱스(Jaccard Index) 방법을 의미할 수 있으나, 본 발명은 이에 한정되지 않는다.
- [0158] 상술한 바와 같이, 제1 비트벡터(91)가 i 번째 그룹 특성 벡터이고, 제2 비트벡터(92)가 j 번째 유효 프로그램 비트벡터인 경우 산출되는 유사도는 제1 유사도를 의미할 수 있다.
- [0159] 상기한 바와 같이, 제1 비트벡터(91)가 i 번째 그룹 특성 벡터이고, 제2 비트벡터(92)가 공통 비트벡터인 경우 산출되는 유사도는 제2 유사도를 의미할 수 있다.
- [0160] 마찬가지로, 제1 비트벡터(91)가 j 번째 유효 프로그램 비트벡터고, 제2 비트벡터(92)가 공통 비트벡터인 경우 산출되는 유사도는 제3 유사도를 의미할 수 있다.
- [0161] 예를 들어, 상기한 제1 유사도, 제2 유사도 및 제3 유사도 각각은 상술한 방법 외에도 각종 유사도 산출 방법을 활용하여 산출될 수 있으며, 본 발명은 특정 유사도 산출 방법으로 한정되지 않는다.
- [0162] 이제 도 10을 참조하여, 프로그램 분류부(1400)에 의해 수행되는 S240 단계를 설명한다.
- [0163] 도 10은 본 발명의 실시예에 따른 프로그램 분류 방법에서 분류 대상 프로그램을 분류하는 방법을 설명하기 위한 순서도이다.
- [0164] 도 10에 도시된 바와 같이, S240 단계는 복수의 그룹 특성 벡터 각각에 대한 분류 대상 프로그램 각각의 포함률을 산출하는 단계(S241) 및 포함률에 제3 임계값을 적용하여 분류 대상 프로그램이 속하는 프로그램 그룹을 결정하는 단계(S243)를 포함한다.

- [0165] S241 단계는, 복수 개의 그룹 특성 벡터 각각에 대한 적어도 하나의 분류 대상 프로그램 각각의 포함률을 산출하는 단계를 의미할 수 있다.
- [0166] 이때, 포함률은 상술한 분류 대상 프로그램이 상술한 그룹 특성 벡터에 포함되는 정도를 나타내는 값을 의미할 수 있다.
- [0167] 예를 들어, S241 단계에서 적어도 하나의 분류 대상 프로그램의 복수 개의 그룹 특성 벡터 각각에 대한 포함률을 산출하기 위하여, 적어도 하나의 분류 대상 프로그램 각각은 분류 대상 프로그램 비트벡터로 변환될 수 있다.
- [0168] 이때, 적어도 하나의 분류 대상 프로그램에 기초하여 적어도 하나의 분류 대상 프로그램 비트벡터를 생성하는 구체적인 방법은, 상기한 도 3에서 설명된 S211 단계, S213 단계, S215 단계 및 S217 단계에서 설명된 바와 같으므로 그 구체적인 설명은 생략한다.
- [0169] 이때, S241 단계에서 산출하는 포함률은 이하 도 11을 참조하여 후술하도록 하며, 중복되는 설명은 생략한다.
- [0170] S243 단계는 포함률에 기설정된 제3 임계값을 적용하여 적어도 하나의 분류 대상 프로그램 각각이 속하는 프로그램 그룹을 결정하는 단계를 의미할 수 있다.
- [0171] 예를 들어, 제1 그룹 특성 벡터가 제1 프로그램 그룹의 특성을 대표하고, 제2 그룹 특성 벡터가 제2 프로그램 그룹의 특성을 대표하며, 제3 그룹 특성 벡터가 제3 프로그램 그룹의 특성을 대표하고, 기설정된 제3 임계값은 70%인 경우를 예로 들어 S241 단계 및 S243 단계를 설명한다.
- [0172] S241 단계에서, 특정 분류 대상 프로그램의 특정 분류 대상 프로그램 비트벡터의 제1 그룹 특성 벡터에 대한 포함률이 50%로 산출되고, 제2 그룹 특성 벡터에 대한 포함률이 80%로 산출되며, 제3 그룹 특성 벡터에 대한 포함률이 30%로 산출된 경우를 가정한다.
- [0173] 상술한 예시의 경우, S243 단계에서 프로그램 분류부(1400)는 특정 분류 대상 프로그램의 제2 그룹 특성 벡터에 대한 포함률이 80%이므로, 특정 분류 대상 프로그램은 제2 프로그램 그룹에 속함을 결정할 수 있다.
- [0174] 이제, 도 11을 참조하여, S241 단계에서 포함률을 산출하는 방법을 설명한다.
- [0175] 도 11은 본 발명의 실시예에 따른 프로그램 분류 방법에서 분류 대상 프로그램을 분류하기 위한 포함률을 산출하는 방법을 설명하기 위한 도면이다.
- [0176] 도 11에 도시된 바와 같이, 제1 그룹 특성 벡터(111)는 i 번째 비트의 비트값이 "1", $i+1$ 번째 비트의 비트값이 "1", j 번째 비트의 비트값이 "1", $j+1$ 번째 비트의 비트값이 "1", $j+2$ 번째 비트의 비트값이 "0"인 비트벡터로 "...11...110..."과 같이 표현될 수 있다.
- [0177] 마찬가지로, 제1 분류 대상 프로그램 비트벡터(112)는 i 번째 비트의 비트값이 "0", $i+1$ 번째 비트의 비트값이 "1", j 번째 비트의 비트값이 "0", $j+1$ 번째 비트의 비트값이 "1", $j+2$ 번째 비트의 비트값이 "1"인 비트벡터로 "...01...011..."과 같이 표현될 수 있다.
- [0178] 상술한 예시에서, 프로그램 분류부(1400)는 제1 그룹 특성 벡터(111) 및 제1 분류 대상 프로그램 비트벡터(112)에서 서로 대응되는 비트의 비트값이 모두 "1"인 비트의 개수를 제1 그룹 특성 벡터(111)의 비트 중 비트값이 "1"인 비트의 개수로 나누어 제1 분류 대상 프로그램 비트벡터(112)의 제1 그룹 특성 벡터(111)에 대한 포함률을 산출할 수 있다.
- [0179] 예를 들어, 제1 그룹 특성 벡터(111) 및 제1 분류 대상 프로그램 비트벡터(112) 각각의 i 번째 비트, $i+1$ 번째 비트, j 번째 비트, $j+1$ 번째 비트, $j+2$ 번째 비트에 기초하여 제1 분류 대상 프로그램 비트벡터(112)의 제1 그룹 특성 벡터(111)에 대한 포함률을 산출하면, 제1 그룹 특성 벡터(111) 및 제1 분류 대상 프로그램 비트벡터(112)에서 서로 대응되는 비트의 비트값이 모두 "1"인 비트는 $i+1$ 번째 비트 및 $j+1$ 번째 비트이며, 그 개수는 2개임을 확인할 수 있다.
- [0180] 한편, 제1 그룹 특성 벡터(111)의 비트 중 비트값이 "1"인 비트는 i 번째 비트, $i+1$ 번째 비트, j 번째 비트, $j+1$ 번째 비트이며, 그 개수는 4개임을 확인할 수 있다.
- [0181] 그 결과, 제1 분류 대상 프로그램 비트벡터(112)의 제1 그룹 특성 벡터(111)에 대한 포함률은 $2/4$ 로 산출될 수 있다.

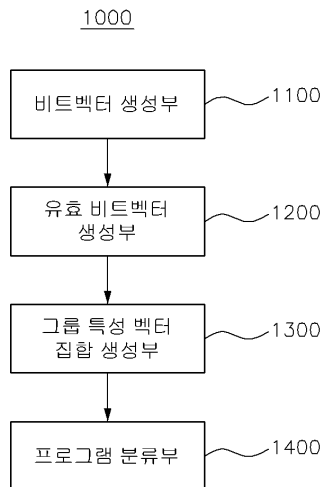
[0182] 이상에서 본 발명에 따른 바람직한 실시예에 대해 설명하였으나, 다양한 형태로 변형이 가능하며, 본 기술분야에서 통상의 지식을 가진 자라면 본 발명의 특허청구범위를 벗어남이 없이 다양한 변형예 및 수정예를 실시할 수 있을 것으로 이해된다.

부호의 설명

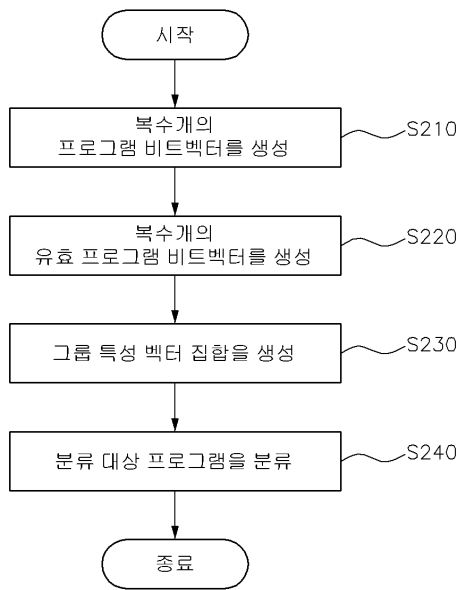
[0183] 1: 프로그램 2: 비트벡터
 61: 제1 프로그램 비트벡터 62: 제2 프로그램 비트벡터
 63: 제n 프로그램 비트벡터 64: 함수 빈도수 벡터
 65: 비트벡터 마스크 66: 제i 프로그램 비트벡터
 67: 제i 유효 프로그램 비트벡터
 91: 제1 비트벡터 92: 제2 비트벡터
 111: 제1 그룹 특성 벡터 112: 제1 분류 대상 프로그램 비트벡터
 1000: 프로그램 분류 장치 1100: 비트벡터 생성부
 1200: 유효 비트벡터 생성부 1300: 그룹 특성 벡터 집합 생성부
 1400: 프로그램 분류부

도면

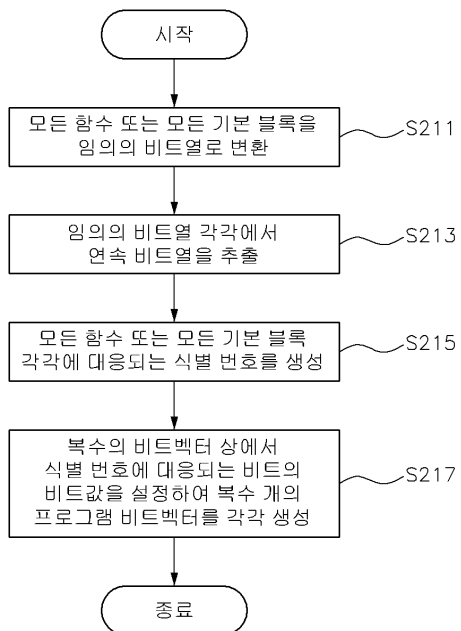
도면1



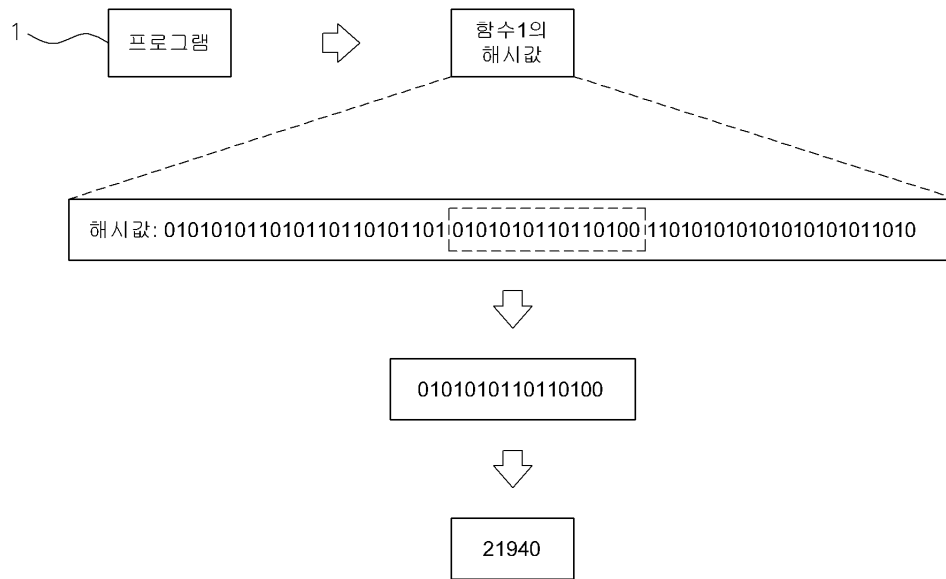
도면2



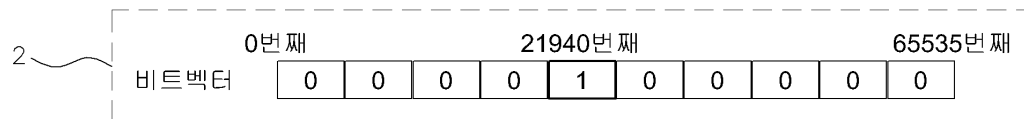
도면3



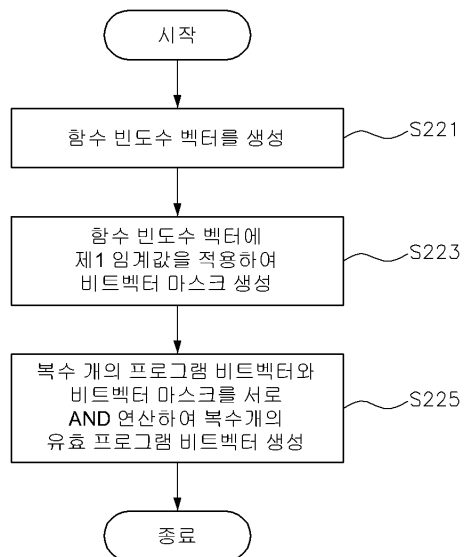
도면4a



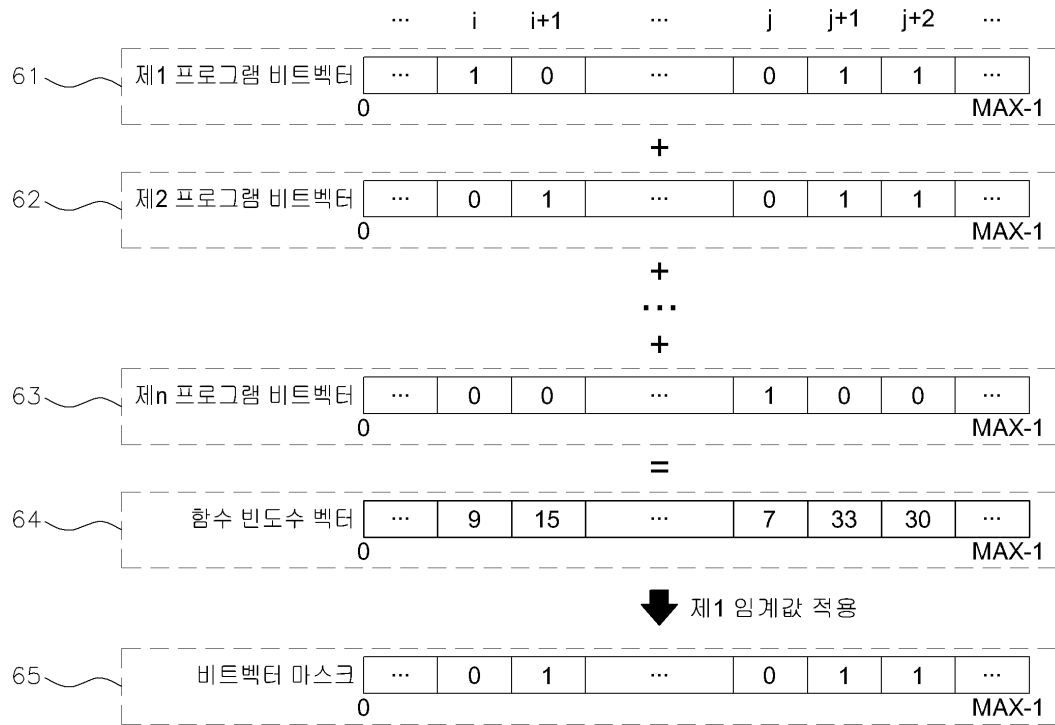
도면4b



도면5



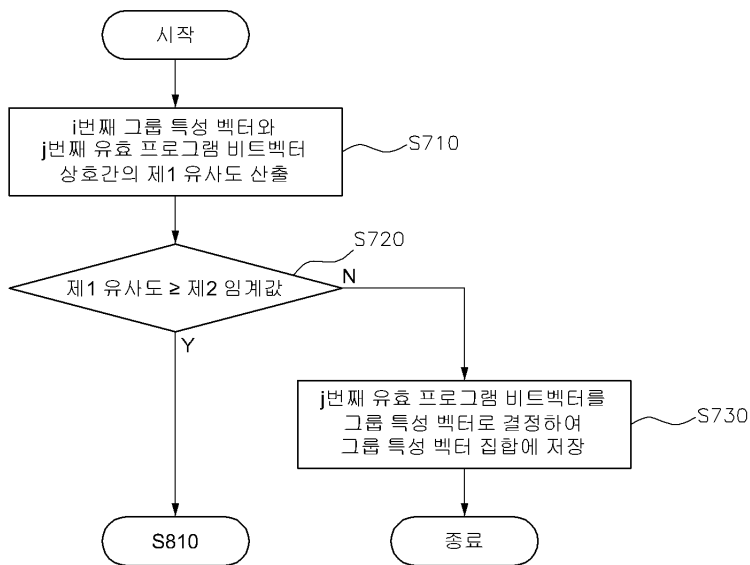
도면6a



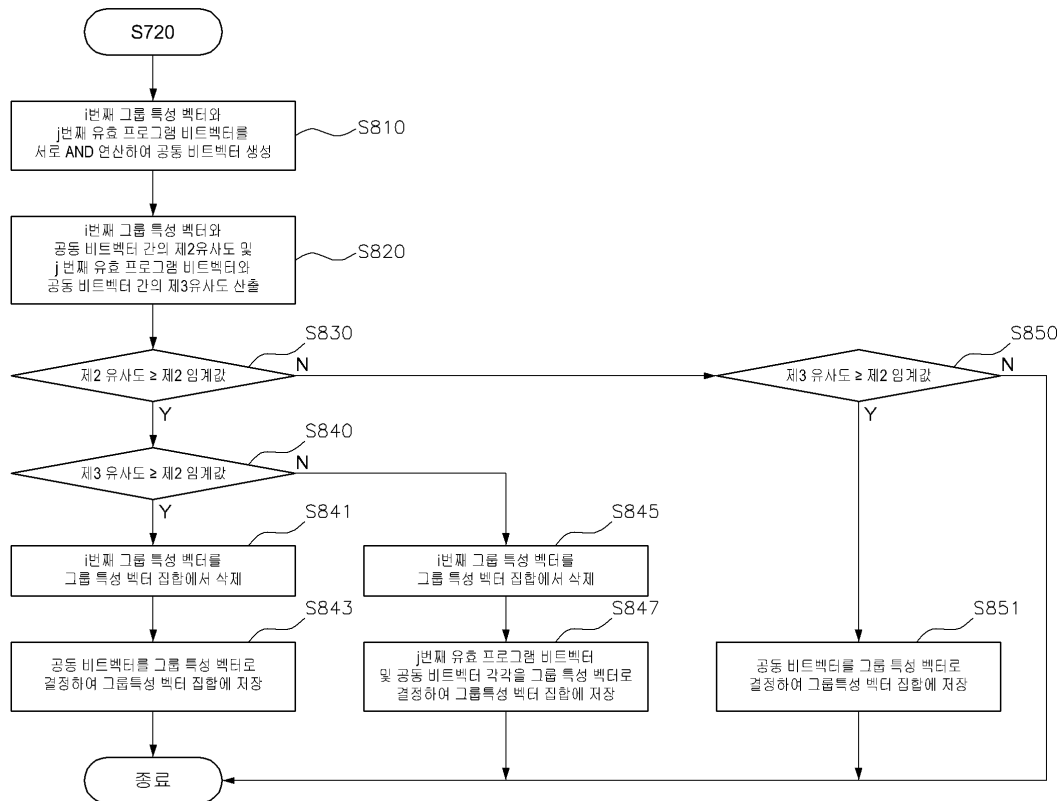
도면6b



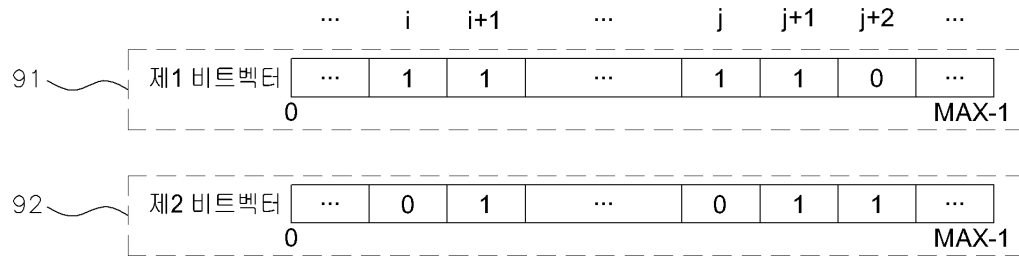
도면7



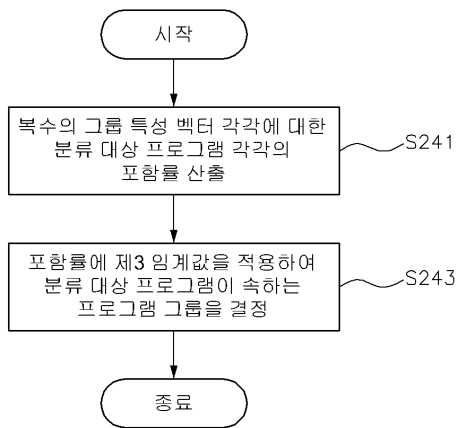
도면8



도면9



도면10



도면11

