(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2018-0098925
(43) 공개일자 2018년09월05일

(51) 국제특허분류(Int. Cl.)
G06F 21/57 (2013.01) *G06F 21/12* (2013.01)
(52) CPC특허분류
G06F 21/57 (2013.01)
G06F 21/12 (2013.01)
(21) 출원번호 10-2017-0025717
(22) 출원일자 2017년02월27일
심사청구일자 2017년02월27일

(71) 출원인
충남대학교산학협력단
대전광역시 유성구 대학로 99 (궁동, 충남대학교)
(72) 발명자
김형식
대전광역시 유성구 대학로 99
윤정무
대전광역시 유성구 대학로145번길 22, 405호
(뒷면에 계속)
(74) 대리인
홍성욱, 심경식

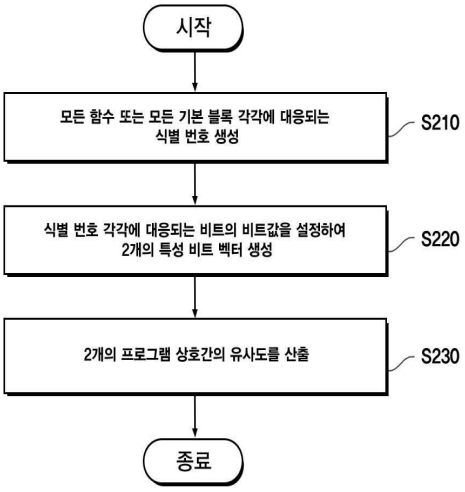
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 **프로그램 유사도 산출 방법 및 장치**

(57) 요 약

식별 번호 생성부가, 2개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록(basic block) 각각을 기설정된 조건에 따라 변환하여 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호를 생성하는 단계, 특성 비트 벡터 생성부가, 2개의 프로그램 각각에 대응되며 기설정된 개수의 비트를 포함하는 2개의 비트 벡터 상에서 식별 번호에 각각 대응되는 비트의 비트값을 설정하여 2개의 특성 비트 벡터를 각각 생성하는 단계 및 프로그램 유사도 산출부가, 2개의 특성 비트 벡터를 서로 대응되는 비트 별로 서로 비교하여 2개의 프로그램 상호간의 유사도를 산출하는 단계를 포함하는, 프로그램 유사도 산출 방법 및 이를 위한 장치에 관한 것이다.

대 표 도 - 도2



(52) CPC특허분류
G06F 2221/033 (2013.01)
(72) 발명자
김기수
대전광역시 동구 계족로489번길 73, 115동 601호

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

명 세 서

청구범위

청구항 1

식별 번호 생성부가, 2개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록(basic block) 각각을 기설정된 조건에 따라 변환하여 상기 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호를 생성하는 단계;

특성 비트 벡터 생성부가, 상기 2개의 프로그램 각각에 대응되며 기설정된 개수의 비트를 포함하는 2개의 비트 벡터(bit vector) 상에서 상기 식별 번호에 각각 대응되는 비트의 비트값을 설정하여 2개의 특성 비트 벡터를 각각 생성하는 단계; 및

프로그램 유사도 산출부가, 상기 2개의 특성 비트 벡터를 서로 대응되는 비트 별로 서로 비교하여 상기 2개의 프로그램 상호간의 유사도를 산출하는 단계를 포함하는, 프로그램 유사도 산출 방법.

청구항 2

제1항에 있어서,

상기 2개의 프로그램 각각이 기계어로 구성된 2개의 실행 프로그램인 경우,

상기 식별 번호를 부여하는 단계 이전에,

프로그램 변환부가, 상기 2개의 실행 프로그램 각각을 함수 또는 기본 블록으로 구성되는 어셈블리언어 기반 프로그램 또는 고급 언어 기반 프로그램으로 변환하는 단계를 더 포함하는, 프로그램 유사도 산출 방법.

청구항 3

제1항에 있어서,

상기 식별 번호를 생성하는 단계는,

상기 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; 및

추출된 상기 연속 비트열 각각을 10진수로 변환하여 상기 식별 번호를 생성하는 단계를 포함하는, 프로그램 유사도 산출 방법.

청구항 4

제1항에 있어서,

상기 2개의 특성 비트 벡터를 각각 생성하는 단계는,

상기 식별 번호 각각에 대응되는 상기 비트의 비트값을 '1'로 설정하는 단계를 포함하는, 프로그램 유사도 산출 방법.

청구항 5

제1항에 있어서,

상기 2개의 프로그램 상호간의 유사도를 산출하는 단계는,

상기 2개의 특성 비트 벡터를 서로 대응되는 비트 별로 순차적으로 비교하여, 서로 일치하는 비트값을 가지는 비트의 개수에 기초하여 상기 2개의 프로그램 상호간의 유사도를 산출하는 단계를 포함하는, 프로그램 유사도 산출 방법.

청구항 6

제1항에 있어서,

상기 2개의 프로그램 상호간의 유사도를 산출하는 단계는,

상기 서로 대응되는 비트의 비트값이 모두 '1'인 비트의 개수를 상기 서로 대응되는 비트의 비트값 중 적어도 하나가 '1'인 비트의 개수로 나누어 상기 2개의 프로그램 상호간의 유사도를 산출하는 단계를 포함하는, 프로그램 유사도 산출 방법.

청구항 7

2개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록(basic block) 각각을 기설정된 조건에 따라 변환하여 상기 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호를 생성하는 식별 번호 생성부;

상기 2개의 프로그램 각각에 대응되며 기설정된 개수의 비트를 포함하는 2개의 비트 벡터 상에서 상기 식별 번호에 각각 대응되는 비트의 비트값을 설정하여 2개의 특성 비트 벡터를 각각 생성하는 특성 비트 벡터 생성부; 및

상기 2개의 특성 비트 벡터를 서로 대응되는 비트 별로 서로 비교하여 상기 2개의 프로그램 상호간의 유사도를 산출하는 프로그램 유사도 산출부를 포함하는, 프로그램 유사도 산출 장치.

청구항 8

제7항에 있어서,

상기 2개의 프로그램 각각이 기계어로 구성된 2개의 실행 프로그램인 경우,

상기 2개의 실행 프로그램 각각을 함수 또는 기본 블록으로 구성되는 어셈블리언어 기반 프로그램 또는 고급 언어 기반 프로그램으로 변환하는 프로그램 변환부를 더 포함하는, 프로그램 유사도 산출 장치.

청구항 9

제7항에 있어서,

상기 식별 번호 생성부는,

상기 모든 함수 또는 모든 기본 블록 각각에 해시함수를 적용하여 상기 모든 함수 또는 모든 기본 블록 각각을 임의의 비트열로 변환하고, 상기 모든 함수 또는 모든 기본 블록 각각에 대응되는 상기 임의의 비트열 각각에서 기설정된 개수의 연속된 비트열인 연속 비트열을 각각 추출하고, 추출된 상기 연속 비트열 각각을 10진수로 변환하여 상기 식별 번호를 생성하는, 프로그램 유사도 산출 장치.

청구항 10

제7항에 있어서,

상기 특성 비트 벡터 생성부는,

상기 식별 번호 각각에 대응되는 상기 비트의 비트값을 '1'로 설정하는, 프로그램 유사도 산출 장치.

청구항 11

제7항에 있어서,

상기 프로그램 유사도 산출부는,

상기 2개의 특성 비트 벡터를 서로 대응되는 비트 별로 순차적으로 비교하여, 서로 일치하는 비트값을 가지는 비트의 개수에 기초하여 상기 2개의 프로그램 상호간의 유사도를 산출하는, 프로그램 유사도 산출 장치.

청구항 12

제7항에 있어서,

상기 프로그램 유사도 산출부는,

상기 서로 대응되는 비트의 비트값이 모두 '1'인 비트의 개수를 상기 서로 대응되는 비트의 비트값 중 적어도 하나가 '1'인 비트의 개수로 나누어 상기 2개의 프로그램 상호간의 유사도를 산출하는, 프로그램 유사도 산출 장치.

발명의 설명

기 술 분 야

[0001] 본 발명은 2 개의 프로그램 상호간의 유사도를 산출하는 프로그램 유사도 산출 방법 및 장치에 관한 것이다.

배 경 기 술

[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, 스마트 폰 등 각종 단말기에 저장할 수 있게 되었다.

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

[0004] 종래에는, 특정 프로그램이 위변조된 프로그램인지 여부, 특정 프로그램이 악성 코드인지 여부 등을 판단하기 위해서는 2개의 프로그램 내에 포함된 함수 또는 기본 블록을 서로 일대일로 직접 비교하는 과정을 거쳐야 했다.

[0005] 그러나, 2 개의 프로그램 각각에 포함된 함수 또는 기본 블록을 서로 일대일로 직접 비교하는 경우에 비교 대상이 되는 프로그램 각각의 용량이 큰 상황에서 2개의 프로그램을 비교하기 위해서는 과도하게 많은 시간이 요구되는 문제가 있다.

[0006] 또한, 상술한 경우에서 비교하는 2개의 프로그램의 용량에 따라 프로그램을 함수 또는 기본 블록 단위로 서로 비교하는데 걸리는 시간이 변하게 되므로, 비교에 걸리는 전체 시간을 예측할 수 없는 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 2개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록(basic block) 각각에 대응되는 식별 번호를 생성하고, 기설정된 크기의 비트 벡터 상에서 식별 번호 각각에 대응되는 비트의 비트값을 설정하여 2개의 프로그램 각각에 대응되는 2개의 특성 비트 벡터를 생성한 뒤, 2개의 특성 비트 벡터를 서로 비교하여 2개의 프로그램 상호간의 유사도를 산출하기 위함이다.

[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 유사도 산출 방법은 식별 번호 생성부가, 2개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록 각각을 기설정된 조건에 따라 변환하여 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호를 생성하는 단계, 특성 비트 벡터 생성부가, 2개의 프로그램 각각에 대응되며 기설정된 개수의 비트를 포함하는 2개의 비트 벡터(bit vector) 상에서 식별 번호에 각각 대응

되는 비트의 비트값을 설정하여 2개의 특성 비트 벡터를 각각 생성하는 단계 및 프로그램 유사도 산출부가, 2개의 특성 비트 벡터를 서로 대응되는 비트 별로 서로 비교하여 2개의 프로그램 상호간의 유사도를 산출하는 단계를 포함한다.

[0011] 예컨대, 본 발명의 실시예에 따른 프로그램 유사도 산출 방법은 2개의 프로그램 각각이 기계어로 구성된 2개의 실행 프로그램인 경우, 식별 번호를 부여하는 단계 이전에, 프로그램 변환부가, 2개의 실행 프로그램 각각을 함수 또는 기본 블록으로 구성되는 어셈블리언어 기반 프로그램 또는 고급 언어 기반 프로그램으로 변환하는 단계를 더 포함한다.

[0012] 예를 들어, 식별 번호를 생성하는 단계는, 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 및 추출된 연속 비트열 각각을 10진수로 변환하여 식별 번호를 생성하는 단계를 포함한다.

[0013] 예컨대, 2개의 특성 비트 벡터를 각각 생성하는 단계는, 식별 번호 각각에 대응되는 비트의 비트값을 '1'로 설정하는 단계를 포함한다.

[0014] 예를 들어, 2개의 프로그램 상호간의 유사도를 산출하는 단계는, 2개의 특성 비트 벡터를 서로 대응되는 비트 별로 순차적으로 비교하여, 서로 일치하는 비트값을 가지는 비트의 개수에 기초하여 2개의 프로그램 상호간의 유사도를 산출하는 단계를 포함한다.

[0015] 일 실시예에 따르면, 2개의 프로그램 상호간의 유사도를 산출하는 단계는, 서로 대응되는 비트의 비트값이 모두 '1'인 비트의 개수를 서로 대응되는 비트의 비트값 중 적어도 하나가 '1'인 비트의 개수로 나누어 2개의 프로그램 상호간의 유사도를 산출하는 단계를 포함한다.

[0016] 상기한 목적을 달성하기 위하여, 본 발명의 실시예에 따른, 프로그램 유사도 산출 장치는 2개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록 각각을 기설정된 조건에 따라 변환하여 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호를 생성하는 식별 번호 생성부, 2개의 프로그램 각각에 대응되며 기설정된 개수의 비트를 포함하는 2개의 비트 벡터 상에서 식별 번호에 각각 대응되는 비트의 비트값을 설정하여 2개의 특성 비트 벡터를 각각 생성하는 특성 비트 벡터 생성부 및 2개의 특성 비트 벡터를 서로 대응되는 비트 별로 서로 비교하여 2개의 프로그램 상호간의 유사도를 산출하는 프로그램 유사도 산출부를 포함한다.

[0017] 예컨대, 본 발명의 실시예에 따른 프로그램 유사도 산출 장치는, 2개의 프로그램 각각이 기계어로 구성된 2개의 실행 프로그램인 경우, 2개의 실행 프로그램 각각을 함수 또는 기본 블록으로 구성되는 어셈블리언어 기반 프로그램 또는 고급 언어 기반 프로그램으로 변환하는 프로그램 변환부를 더 포함한다.

[0018] 일 실시예에 따르면, 식별 번호 생성부는, 모든 함수 또는 모든 기본 블록 각각에 해시함수를 적용하여 모든 함수 또는 모든 기본 블록 각각을 임의의 비트열로 변환하고, 모든 함수 또는 모든 기본 블록 각각에 대응되는 임의의 비트열 각각에서 기설정된 개수의 연속된 비트열인 연속 비트열을 각각 추출하고, 추출된 연속 비트열 각각을 10진수로 변환하여 식별 번호를 생성한다.

[0019] 일 실시예에 따라, 특성 비트 벡터 생성부는, 식별 번호 각각에 대응되는 비트의 비트값을 '1'로 설정한다.

[0020] 예를 들어, 프로그램 유사도 산출부는, 2개의 특성 비트 벡터를 서로 대응되는 비트 별로 순차적으로 비교하여, 서로 일치하는 비트값을 가지는 비트의 개수에 기초하여 2개의 프로그램 상호간의 유사도를 산출한다.

[0021] 예컨대, 프로그램 유사도 산출부는, 서로 대응되는 비트의 비트값이 모두 '1'인 비트의 개수를 서로 대응되는 비트의 비트값 중 적어도 하나가 '1'인 비트의 개수로 나누어 2개의 프로그램 상호간의 유사도를 산출한다.

발명의 효과

[0022] 본 발명의 일 실시예에 따르면, 2개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록을 2개의 특성 비트 벡터로 표현하여 서로 비교함으로써 유사도를 산출하기 때문에, 모든 함수 또는 모든 기본 블록을 직접 비교하는 것에 비해 더욱 신속하게 2개의 프로그램 상호간의 유사도를 산출할 수 있다.

[0023] 나아가, 본 발명의 일 실시예에 따르면, 2개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록을 기설정된 크기의 2개의 특성 비트 벡터로 표현하기 때문에, 2개의 프로그램 상호간의 유사도를 산출하기 위해 걸리는 소요 시간을 보다 정확하게 예측할 수 있다.

도면의 간단한 설명

[0024] 도 1은 본 발명의 실시예에 따른 프로그램 유사도 산출 장치를 설명하기 위한 구성도이다.

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

도 3은 본 발명의 실시예에 따른 프로그램 유사도 산출 방법에서 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호 생성 단계를 설명하기 위한 순서도이다.

도 4는 본 발명의 실시예에 따른 프로그램 유사도 산출 방법 및 장치에서 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호를 생성하는 방법을 설명하기 위한 도면이다.

도 5는 본 발명의 실시예에 따른 프로그램 유사도 산출 방법 및 장치에서 2개의 특성 비트 벡터를 생성하는 방법을 설명하기 위한 도면이다.

도 6은 본 발명의 실시예에 따른 프로그램 유사도 산출 방법 및 장치에서 2개의 특성 비트 벡터에 기초하여 2개의 프로그램 상호간의 유사도를 산출하는 방법을 설명하기 위한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0025] 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 도면상에 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 경우에는 그 상세한 설명은 생략한다.

[0026] 이하, 본 발명의 실시예에 따른 프로그램 유사도 산출 방법 및 장치를 첨부된 도면을 참조하여 상세하게 설명하면 아래와 같다.

[0027] 이제, 도 1을 참조하여 본 발명의 실시예에 따른 프로그램 유사도 산출 장치를 설명한다.

[0028] 도 1은 본 발명의 실시예에 따른 프로그램 유사도 산출 장치를 설명하기 위한 구성도이다.

[0029] 도 1에 도시된 바와 같이, 본 발명의 실시예에 따른 프로그램 유사도 산출 장치(100)는 식별 번호 생성부(110), 특성 비트 벡터 생성부(120) 및 프로그램 유사도 산출부(130)를 포함한다.

[0030] 예컨대, 도면에 도시되지는 않았으나, 본 발명의 실시예에 따른 프로그램 유사도 산출 장치(100)는 프로그램 변환부(미도시)를 더 포함할 수 있으나, 본 발명은 이에 한정되지 않는다.

[0031] 이하, 본 발명의 실시예에 따른 프로그램 유사도 산출 장치(100)가 상호간에 유사도를 산출하는 대상이 제1 프로그램(1) 및 제2 프로그램(2)을 2개의 프로그램(1, 2)으로 정의하여 설명한다.

[0032] 예컨대, 2개의 프로그램(1, 2) 각각이 기계어로 구성된 2개의 실행 프로그램인 경우, 프로그램 변환부(미도시)는 2개의 실행 프로그램 각각을 함수 또는 기본 블록으로 구성되는 어셈블리언어 기반 프로그램 또는 고급 언어 기반 프로그램으로 변환한다.

[0033] 예컨대, 프로그램 변환부(미도시)는 2개의 프로그램(1, 2) 각각이 "0" 또는 "1"로 표현되는 기계어로 구성된 2개의 실행 프로그램인 경우, 기계어를 디스어셈블(disassemble)하여 2개의 실행 프로그램 각각을 함수 또는 기본 블록으로 구분할 수 있는 언어인 어셈블리언어 기반 프로그램으로 변환할 수 있다.

[0034] 예를 들면, 본 발명의 실시예에 따른 프로그램 유사도 산출 장치(100)는 어셈블리언어 기반 프로그램 또는 고급 언어 기반 프로그램에 기초하여 2개의 프로그램(1, 2) 상호간의 유사도를 산출할 수 있기 때문에, 2개의 프로그램(1, 2) 각각이 이미 어셈블리언어 기반 프로그램인 경우 프로그램 변환부(미도시)의 동작은 생략될 수 있다.

[0035] 예컨대, 프로그램 변환부(미도시)는 2개의 프로그램(1, 2) 각각이 이미 어셈블리언어 기반 프로그램인 경우에도, 어셈블리언어 기반 프로그램을 더 변환하여 C언어, JAVA언어를 비롯한 고급 언어 기반 프로그램으로 상술한 2개의 프로그램(1, 2) 각각을 더 변환할 수도 있으나, 본 발명은 이에 한정되지 않는다.

[0036] 일 실시예에 따르면, 프로그램 변환부(미도시)는 2개의 프로그램(1, 2) 각각이 이미 고급 언어 기반 프로그램인 경우에도, 고급 언어 기반 프로그램을 더 변환하여 상술한 2개의 프로그램(1, 2) 각각을 어셈블리언어 기반 프

로그램으로 더 변환할 수도 있으나, 본 발명은 이에 한정되지 않는다.

[0037] 즉, 2개의 프로그램(1, 2) 각각이 기계어에 기반한 프로그램인 경우, 프로그램 변환부(미도시)는 2개의 프로그램(1, 2) 각각을 어셈블리언어 기반 프로그램 또는 고급 언어 기반 프로그램으로 변환할 수 있다.

[0038] 한편, 2개의 프로그램(1, 2) 각각이 이미 어셈블리언어 기반 프로그램 또는 고급 언어 기반 프로그램인 경우, 프로그램 변환부(미도시)는 생략될 수 있다.

[0039] 식별 번호 생성부(110)는 2개의 프로그램 각각에 포함된 모든 함수 또는 모든 기본 블록 각각을 기설정된 조건에 따라 변환하여 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호를 생성한다.

[0040] 예컨대, 2개의 프로그램(1, 2) 각각은 함수 또는 기본 블록 단위로 구분할 수 있는 프로그램인 어셈블리언어 기반 프로그램 또는 고급 언어 기반 프로그램을 의미할 수 있다.

[0041] 예를 들어, 2개의 프로그램(1, 2) 각각이 고급 언어 중 C언어에 기반한 프로그램인 경우, 2개의 프로그램(1, 2) 각각에 포함된 모든 함수는 "printf("Hello, World!")"과 같이 함수명과 인자값을 모두 포함하는 함수 각각을 의미할 수 있다.

[0042] 예를 들어, 2개의 프로그램(1, 2) 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.

[0043] 예를 들어, 2개의 프로그램(1, 2) 각각이 고급 언어 중 C언어에 기반한 프로그램인 경우, 2개의 프로그램(1, 2) 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.

[0044] 예컨대, 식별 번호 생성부(110)는, 모든 함수 또는 모든 기본 블록 각각에 해시함수를 적용하여 모든 함수 또는 모든 기본 블록 각각을 임의의 비트열로 변환하고, 모든 함수 또는 모든 기본 블록 각각에 대응되는 임의의 비트열 각각에서 기설정된 개수의 연속된 비트열인 연속 비트열을 각각 추출하고, 추출된 연속 비트열 각각을 10진수로 변환하여 식별 번호를 생성한다.

[0045] 특성 비트 벡터 생성부(120)는 2개의 프로그램(1, 2) 각각에 대응되며 기설정된 개수의 비트를 포함하는 2개의 비트 벡터 상에서 식별 번호에 각각 대응되는 비트의 비트값을 설정하여 2개의 특성 비트 벡터를 각각 생성한다.

[0046] 예컨대, 상술한 2개의 비트 벡터 각각은 복수 개의 비트를 포함하며, 각각의 비트는 비트 벡터 상에 포함된 하나의 비트에 대응되며 각각의 비트는 특정 식별 번호에 대응되는 비트가 사용되면 '1'의 비트값을 가지고 특정 식별 번호에 대응되는 비트가 사용되지 않으면 '0'의 비트값을 가질 수 있다.

[0047] 예컨대, 상술한 2개의 비트 벡터 각각은 모든 비트의 비트값이 '0'으로 초기화될 수 있다.

[0048] 예컨대, 특성 비트 벡터 생성부(120)는, 식별 번호 각각에 대응되는 비트의 비트값을 '1'로 설정할 수 있다.

[0049] 즉, 특성 비트 벡터 생성부(120)는 모든 비트의 비트값이 '0'으로 초기화된 기설정된 크기의 비트 벡터 상에서 식별 번호에 대응되는 비트의 비트값을 '1'로 변경하여, 특성 비트 벡터를 생성할 수 있다.

[0050] 프로그램 유사도 산출부(130)는 2개의 특성 비트 벡터를 서로 대응되는 비트 별로 서로 비교하여 2개의 프로그램(1, 2) 상호간의 유사도를 산출한다.

[0051] 예컨대, 프로그램 유사도 산출부(130)는, 2개의 특성 비트 벡터를 서로 대응되는 비트 별로 순차적으로 비교하여, 서로 일치하는 비트값을 가지는 비트의 개수에 기초하여 2개의 프로그램(1, 2) 상호간의 유사도를 산출할 수 있다.

[0052] 일 실시예에 따라, 프로그램 유사도 산출부(130)는, 서로 대응되는 비트의 비트값이 모두 '1'인 비트의 개수를 서로 대응되는 비트의 비트값 중 적어도 하나가 '1'인 비트의 개수로 나누어 2개의 프로그램(1, 2) 상호간의 유사도를 산출할 수 있다.

[0053] 즉, 프로그램 유사도 산출부(130)는 2개의 프로그램(1, 2) 상호간의 유사도를 산출하기 위하여, 2개의 프로그램(1, 2) 각각에 포함된 모든 함수 또는 모든 기본 블록의 존재 여부를 표시하는 2개의 특성 비트 벡터를 서로 비

교함으로써, 2개의 프로그램(1, 2) 상호간의 유사도를 산출할 수 있다.

[0054] 본 발명의 실시예에 따른, 프로그램 유사도 산출 장치(100)의 각각의 구성에 대한 보다 구체적인 설명은 이하 도 2 내지 도 6을 참조하여 후술하도록 하며, 중복되는 설명은 생략한다.

[0055] 이제 도 2를 참조하여, 본 발명의 실시예에 따른 프로그램 유사도 산출 방법을 설명한다.

[0056] 도 2는 본 발명의 실시예에 따른 프로그램 유사도 산출 방법을 설명하기 위한 순서도이다.

[0057] 도 2에 도시된 바와 같이, 본 발명의 실시예에 따른 프로그램 유사도 산출 방법은 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호 생성 단계(S210), 식별 번호 각각에 대응되는 비트의 비트값을 설정하여 2개의 특성 비트 벡터 생성 단계(S220) 및 2개의 프로그램 상호간의 유사도를 산출 단계(S230)를 포함한다.

[0058] 일 실시예에 따르면, 도면에 도시되지는 않았으나, 본 발명의 실시예에 따른 프로그램 유사도 산출 방법은, 프로그램 변환 단계(미도시)를 더 포함할 수도 있다.

[0059] 예컨대, 프로그램 변환 단계(미도시)는 유사도 산출의 대상이 되는 2개의 프로그램(1, 2) 각각이 기계어로 구성된 2개의 실행 프로그램인 경우, S210 단계 이전에, 수행될 수 있다.

[0060] 이는, 기계어는 "0" 또는 "1"로 구성되기 때문이며, S210 단계는 2개의 프로그램(1, 2) 각각에 포함된 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호를 생성하기 위하여, 2개의 프로그램(1, 2) 각각에 함수 또는 기본 블록으로 표현되어야 수행될 수 있기 때문이다.

[0061] 예컨대, 2개의 프로그램(1, 2) 각각이 어셈블리언어에 기반하거나 고급 언어에 기반하는 경우, 프로그램 변환 단계(미도시)는 생략될 수 있다.

[0062] 보다 상세하게, 프로그램 변환 단계(미도시)는 프로그램 변환부(미도시)가, 2개의 실행 프로그램 각각을 함수 또는 기본 블록으로 구성되는 어셈블리언어 기반 프로그램 또는 고급 언어 기반 프로그램으로 변환하는 단계를 의미할 수 있다.

[0063] S210 단계는, 식별 번호 생성부(110)가, 2개의 프로그램(1, 2) 각각에 포함된 모든 함수 또는 모든 기본 블록 각각을 기설정된 조건에 따라 변환하여 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호를 생성하는 단계를 의미할 수 있다.

[0064] 예컨대, S210 단계의 수행 결과 2개의 프로그램(1, 2) 각각에 포함된 모든 함수 또는 모든 기본 블록 각각은 대응되는 식별 번호로 변환되게 되며, 그 결과 2개의 프로그램(1, 2) 각각은 복수 개의 식별 번호의 집합으로 표현될 수 있게 된다.

[0065] 이제 도 3 및 도 4를 동시에 참조하여, S210 단계를 설명한다.

[0066] 도 3은 본 발명의 실시예에 따른 프로그램 유사도 산출 방법에서 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호 생성 단계를 설명하기 위한 순서도이다.

[0067] 도 4는 본 발명의 실시예에 따른 프로그램 유사도 산출 방법 및 장치에서 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호를 생성하는 방법을 설명하기 위한 도면이다.

[0068] 도 3에 도시된 바와 같이, S210 단계는 모든 함수 또는 모든 기본 블록 각각을 임의의 비트열로 변환 단계(S211), 임의의 비트열 각각에서 기설정된 개수의 연속 비트열 추출 단계(S212) 및 연속 비트열 각각을 10진수로 변환하여 식별 번호 생성 단계(S213)를 포함한다.

[0069] S211 단계는, 모든 함수 또는 모든 기본 블록 각각에 해시함수를 적용하여 모든 함수 또는 모든 기본 블록 각각을 임의의 비트열로 변환하는 단계를 의미할 수 있다.

[0070] 예컨대, 해시함수의 출력값은 그 크기가 일정하되 임의의 값을 가지며, 해시함수에 동일한 입력값을 넣는 경우, 항상 같은 출력값이 나오지만 출력값만을 이용해서 입력값을 역산할 수는 없게 된다.

[0071] 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.

[0072] 도 4에 도시된 바와 같이, S211 단계에서 제1 프로그램(1) 또는 제2 프로그램(2) 각각에 포함된 특정 함수 또는 특정 기본 블록에 대해 해시함수를 적용하는 경우 특정 함수 또는 특정 기본 블록에 대응되는 해시값이 생성될

수 있으며, 이때 생성된 해시값은 임의의 비트열로 정의한다.

[0073] 예를 들어, S211 단계에서 제1 프로그램(1)이 어셈블리언어에 기반한 프로그램인 경우, 함수명과 인자값을 포함하는 특정 함수인 "mov ebp, esp"에 해시함수를 적용하는 경우를 예로들어 S211 단계를 설명하면, S211 단계의 연산 결과값으로"01010101101011011010110101010101101101001101010101010101011010"와 같은 해시값이 생성될 수 있으며, 이때 생성된 해시값을 임의의 비트열이라 할 수 있다.

[0074] 이 때, 2개의 프로그램(1, 2) 각각에 포함된 모든 함수 또는 모든 기본 블록에 대해 해시함수를 적용한다면, 2개의 프로그램(1, 2) 각각에 포함된 모든 함수 또는 모든 기본 블록 각각은 임의의 비트열로 표현될 수 있다.

[0075] S212 단계는, 모든 함수 또는 모든 기본 블록 각각에 대응되는 임의의 비트열 각각에서 기설정된 개수의 연속된 비트열인 연속 비트열을 각각 추출하는 단계를 의미할 수 있다.

[0076] 예를 들어, 도 4에 도시된 바와 같이, 임의의 비트열인 "01010101101011011010110101010101101101001101010101010101011010"의 최초의 비트값인 "0"을 0번째 비트라 정의할 때, S212 단계에서 24번째 내지 39번째로 연속되는 16개의 비트를 연속 비트열로 추출한 결과 "0101010110110100"이 추출될 수 있다.

[0077] 이때, S212 단계에서 추출되는 연속된 비트열의 기설정된 개수는 미리 설정되어 식별 번호 생성부(110)에 저장될 수 있으며, 본 발명은 특정 개수의 비트열을 추출하는 것으로 한정되지 않는다.

[0078] 일 실시예에 따르면, S212 단계에서 식별 번호 생성부(110)는 기설정된 개수의 연속된 비트열을 추출하는 대신에, 기설정된 개수의 불연속한 비트열을 추출할 수도 있으며, 본 발명은 연속된 비트열을 추출하는 것으로 한정되지 않는다.

[0079] S213 단계는, 추출된 연속 비트열 각각을 10진수로 변환하여 식별 번호를 생성하는 단계를 의미할 수 있다.

[0080] 예컨대, 도 4에 도시된 바와 같이 연속 비트열 "0101010110110100"의 10진수 변환 값은 21,940이기 때문에, S211, S212 및 S213 단계를 거쳐 특정 함수인 "mov ebp, esp"에 대응되는 식별 번호는 21,940으로 생성될 수 있다.

[0081] 다시 말해, 2개의 프로그램(1, 2) 각각에 포함된 모든 함수 또는 모든 기본 블록 각각에 S211, S212 및 S213 단계를 적용하면, 2개의 프로그램(1, 2) 각각은 복수 개의 식별 번호로 표현될 수 있게 된다.

[0082] 이제 도 2 및 도 5를 동시에 참조하여, S220 단계를 설명한다.

[0083] 도 5는 본 발명의 실시예에 따른 프로그램 유사도 산출 방법 및 장치에서 2개의 특성 비트 벡터를 생성하는 방법을 설명하기 위한 도면이다.

[0084] S220 단계는, 특성 비트 벡터 생성부(120)가, 2개의 프로그램(1, 2) 각각에 대응되며 기설정된 개수의 비트를 포함하는 2개의 비트 벡터 상에서 식별 번호에 각각 대응되는 비트의 비트값을 설정하여 2개의 특성 비트 벡터를 각각 생성하는 단계를 의미할 수 있다.

[0085] 예컨대, 2개의 비트 벡터는 각각이 포함하는 모든 비트의 비트값이 '0'으로 초기화된 상태일 수 있다.

[0086] 도 5에 도시된 바와 같이, 상술한 예시에서 S211, S212 및 S213 단계를 통해 제1 프로그램(1)에 포함된 함수인 "mov ebp, esp"에 대응되는 식별 번호는 21,940으로 생성되었으며, S220 단계에서 특성 비트 벡터 생성부(120)는 제1 프로그램(1)에 대응되는 비트 벡터 상에서 21,940 번째 비트의 비트값을 '1'로 설정할 수 있다.

[0087] S220 단계에서, 특성 비트 벡터 생성부(120)가 제1 프로그램(1)에 포함된 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호에 대응되는 비트의 비트값을 설정하면, 제1 프로그램(1)에 대응되는 제1 특성 비트 벡터(미도시)가 생성될 수 있다.

[0088] 마찬가지로, S220 단계에서 특성 비트 벡터 생성부(120)가 제2 프로그램(2)에 포함된 모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호에 대응되는 비트의 비트값을 설정하면, 제2 프로그램(2)에 대응되는 제2 특성 비트 벡터(미도시)가 생성될 수 있다.

[0089] S220 단계의 수행결과, 제1 프로그램(1)은 제1 프로그램(1)에 포함된 모든 함수 또는 모든 기본 블록 각각의 식별 번호에 기초하여, 제1 프로그램(1)에 포함된 모든 함수 또는 모든 기본 블록의 존재 여부를 나타내는 제1 특성 비트 벡터(미도시)로 변환될 수 있으며, 마찬가지로 제2 프로그램(2) 또한 제2 특성 비트 벡터(미도시)로 변

환될 수 있다.

[0090] 다시 말해 S220 단계는, 식별 번호 각각에 대응되는 비트의 비트값을 '1'로 설정하는 단계를 포함할 수 있다.

[0091] 이제 도 2 및 도 6을 동시에 참조하여, S230 단계를 설명한다.

[0092] S230 단계는, 프로그램 유사도 산출부(130)가, 2개의 특성 비트 벡터를 서로 대응되는 비트 별로 서로 비교하여 2개의 프로그램 상호간의 유사도를 산출하는 단계를 의미할 수 있다.

[0093] 도 6에 도시된 바와 같이, 제1 프로그램(1, 프로그램 A)과 제2 프로그램(2, 프로그램 B) 상호간의 유사도를 산출하기 위하여, S230 단계에서 프로그램 유사도 산출부(130)는 제1 프로그램(1, 프로그램 A)에 대응되는 제1 특성 비트 벡터(미도시)와 제2 프로그램(2, 프로그램 B)에 대응되는 제2 특성 비트 벡터(미도시) 각각의 비트를 서로 비교할 수 있다.

[0094] 예컨대, 제1 특성 비트 벡터(미도시)는 제1 프로그램(1, 프로그램 A)의 우측 화살표 영역에 도시된 특성 비트 벡터를 의미할 수 있다.

[0095] 예를 들어, 제2 특성 비트 벡터(미도시)는 제2 프로그램(2, 프로그램 B)의 우측 화살표 영역에 도시된 특성 비트 벡터를 의미할 수 있다.

[0096] 이제, S230 단계에 대한 보다 구체적인 설명에 앞서, 도 6을 참조하여 제1 특성 비트 벡터(미도시) 및 제2 특성 비트 벡터(미도시) 각각의 의미에 대해 설명한다.

[0097] 예를 들어, 제1 프로그램(1, 프로그램 A)의 우측 화살표 영역에 도시된 제1 특성 비트 벡터(미도시)가 0번째 비트부터 9번째 비트까지 총 10개의 비트를 가지는 경우를 가정한다.

[0098] 이 경우, 제1 특성 비트 벡터(미도시)는 0번째 비트의 비트값이 '0', 1번째 비트의 비트값이 '1', 2번째 비트의 비트값이 '1', 3번째 비트의 비트값이 '0', 4번째 비트의 비트값이 '0', 5번째 비트의 비트값이 '1', 6번째 비트의 비트값이 '0', 7번째 비트의 비트값이 '1', 8번째 비트의 비트값이 '0', 9번째 비트의 비트값이 '0'인 비트 벡터로 '0110010100'과 같이 표현될 수 있다.

[0099] 이 경우, 제1 특성 비트 벡터(미도시)의 의미는 제1 프로그램(1, 프로그램 A)이 식별 번호 0, 3, 4, 6, 8, 9 각각에 대응되는 함수 또는 기본 블록을 포함하지 않으며, 식별 번호 1, 2, 5, 7 각각에 대응되는 함수 또는 기본 블록을 포함한다는 의미일 수 있다.

[0100] 마찬가지로, 제2 프로그램(2, 프로그램 B)의 우측 화살표 영역에 도시된 제2 특성 비트 벡터(미도시)가 0번째 비트부터 9번째 비트까지 총 10개의 비트를 가지는 경우를 가정한다.

[0101] 이 경우, 제2 특성 비트 벡터(미도시)는 0번째 비트의 비트값이 '0', 1번째 비트의 비트값이 '1', 2번째 비트의 비트값이 '1', 3번째 비트의 비트값이 '0', 4번째 비트의 비트값이 '1', 5번째 비트의 비트값이 '1', 6번째 비트의 비트값이 '0', 7번째 비트의 비트값이 '0', 8번째 비트의 비트값이 '1', 9번째 비트의 비트값이 '0'인 비트 벡터로 '0110110010'과 같이 표현될 수 있다.

[0102] 이 경우, 제2 특성 비트 벡터(미도시)의 의미는 제2 프로그램(2, 프로그램 B)이 식별 번호 0, 3, 6, 7, 9 각각에 대응되는 함수 또는 기본 블록을 포함하지 않으며, 식별 번호 1, 2, 4, 5, 8 각각에 대응되는 함수 또는 기본 블록을 포함한다는 의미일 수 있다.

[0103] 예컨대, S230 단계는, 2개의 특성 비트 벡터를 서로 대응되는 비트 별로 순차적으로 비교하여, 서로 일치하는 비트값을 가지는 비트의 개수에 기초하여 2개의 프로그램(1, 2) 상호간의 유사도를 산출하는 단계를 포함할 수 있다.

[0104] 이제 도 6의 상술한 예시를 참조하여, S230 단계의 제1 실시예를 설명한다.

[0105] 상술한 예시의 상황에서, 제1 특성 비트 벡터(미도시)는 '0110010100'과 같이 표현되며, 제2 특성 비트 벡터(미도시)는 '0110110010'과 같이 표현되며, S230 단계에서 프로그램 유사도 산출부(130)는 상술한 제1 특성 비트 벡터(미도시) 및 제2 특성 비트 벡터(미도시)를 서로 동일한 비트끼리 비교함으로써, 2개의 프로그램(1, 2) 상호간의 유사도를 산출할 수 있다.

[0106] 상술한 예시의 경우에서 제1 특성 비트 벡터(미도시) 및 제2 특성 비트 벡터(미도시)를 비교하면, 2개의 특성 비트 벡터는 서로 0, 1, 2, 3, 5, 6, 9 번째 비트가 서로 동일하며, 4, 7, 8 번째 비트가 서로 상이함을 확인할

수 있다.

[0107] 이 경우, S230 단계에서 프로그램 유사도 산출부(130)는 전체 비트의 개수인 10개와 서로 동일한 비트의 개수인 7개의 비율에 기초하여 제1 특성 비트 벡터(미도시)와 제2 특성 비트 벡터(미도시) 상호간의 유사도를 산출할 수 있으며, 그 결과 제1 프로그램(1, 프로그램 A)와 제2 프로그램(2, 프로그램 B) 상호간의 유사도가 산출될 수 있다.

[0108] 상술한 실시예의 경우, 제1 프로그램(1, 프로그램 A)와 제2 프로그램(2, 프로그램 B) 상호간의 유사도는 7/10인 70%일 수 있으나, 본 발명은 이에 한정되지 않는다.

[0109] 한편, S230 단계는, 서로 대응되는 비트의 비트값이 모두 '1'인 비트의 개수를 상기 서로 대응되는 비트의 비트값 중 적어도 하나가 '1'인 비트의 개수로 나누어 2개의 프로그램(1, 2) 상호간의 유사도를 산출하는 단계를 포함할 수 있다.

[0110] 다시 도 6의 상술한 예시를 참조하여, S230 단계의 제2 실시예를 설명한다.

[0111] 상술한 예시의 경우에서, 제1 특성 비트 벡터(미도시)에 포함된 각각의 비트와 제2 특성 비트 벡터(미도시)에 포함된 각각의 비트 모두에서 비트값이 '1'인 비트는 1, 2, 5번째 비트로, 서로 대응되는 비트의 비트값이 모두 '1'인 비트의 개수는 3개임을 확인할 수 있다.

[0112] 같은 예시의 경우에서, 제1 특성 비트 벡터(미도시)에 포함된 각각의 비트와 제2 특성 비트 벡터(미도시)에 포함된 각각의 비트 중 어느 하나의 비트값이 '1'인 비트는 1, 2, 4, 5, 7, 8 번째 비트로, 서로 대응되는 비트의 비트값 중 적어도 하나가 '1'인 비트의 개수는 6개임을 확인할 수 있다.

[0113] 그 결과, 제1 특성 비트 벡터(미도시)와 제2 특성 비트 벡터(미도시) 상호간의 유사도는 3/6으로 나타날 수 있으며, 제1 프로그램(1) 및 제2 프로그램(2) 상호간의 유사도 또한 3/6로 나타날 수 있게 된다.

[0114] 본 발명의 제2 실시예에 따른 유사도 산출 방법은, 제1 특성 비트 벡터(미도시)와 제2 특성 비트 벡터(미도시)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.

[0115] 예를 들어, S230 단계에서 프로그램 유사도 산출부(130)는 상술한 2가지 유사도 산출 방법 외에도 각종 유사도 산출 방법을 활용하여 제1 프로그램(1) 및 제2 프로그램(2) 상호간의 유사도를 산출할 수 있으며, 본 발명은 특정 유사도 산출 방법으로 한정되지 않는다.

[0116] 본 발명의 실시예에 따른, 프로그램 유사도 산출 방법 및 장치에 따르면, 제1 프로그램(1) 및 제2 프로그램(2)의 실제 크기에 관계없이, 동일한 크기의 특성 비트 벡터로 제1 프로그램(1) 및 제2 프로그램(2)을 표현하여 유사도를 산출할 수 있기 때문에, 유사도 산출에 소요되는 시간을 정확하게 예측할 수 있으며, 더 나아가 제1 프로그램(1) 및 제2 프로그램(2) 각각에 포함된 함수 또는 기본 블록을 직접 비교하지 않고, 특정 함수 또는 특정 기본 블록의 존재 여부를 나타내는 제1 특성 비트 벡터(미도시) 및 제2 특성 비트 벡터(미도시)를 비교함으로써, 제1 프로그램(1) 및 제2 프로그램(2) 상호간의 유사도를 산출하기 때문에, 보다 신속하게 2개의 프로그램(1, 2) 상호간의 유사도를 산출할 수 있다.

[0117] 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

[0118] 1: 제1 프로그램

2: 제2 프로그램

100: 프로그램 유사도 산출 장치

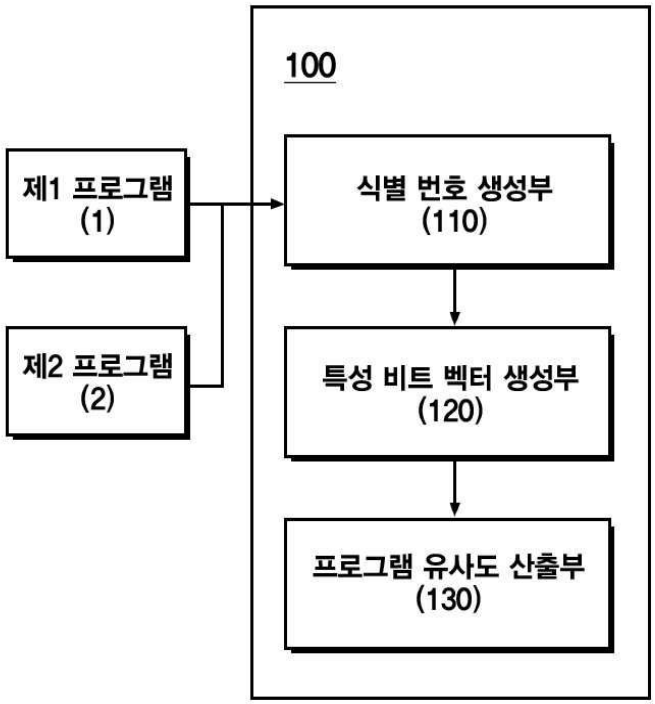
110: 식별 번호 생성부

120: 특성 비트 벡터 생성부

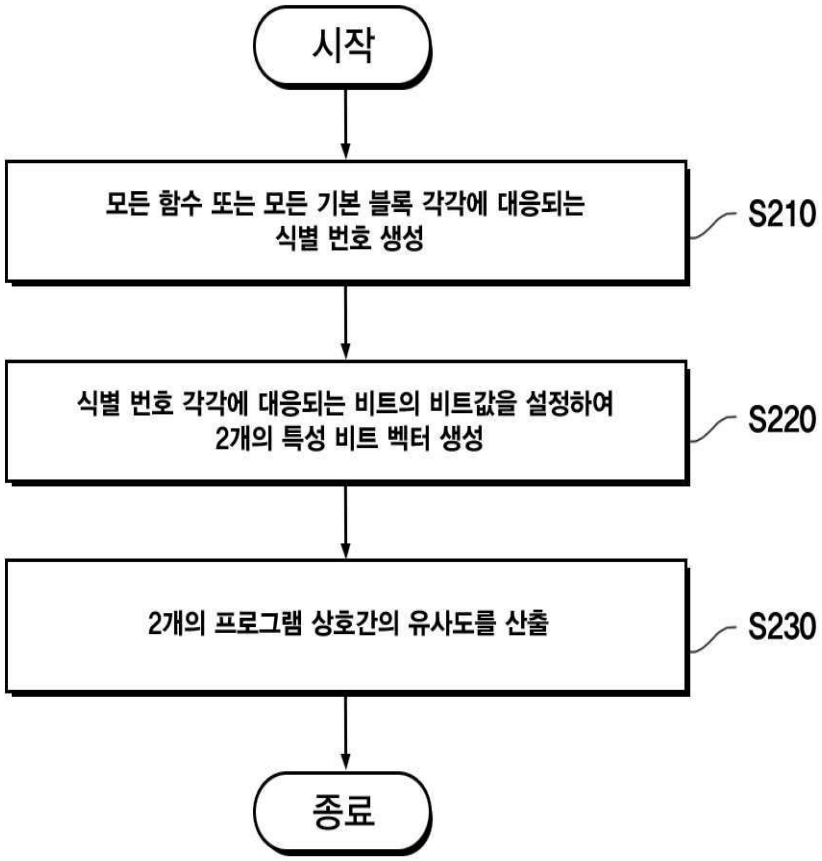
130: 프로그램 유사도 산출부

도면

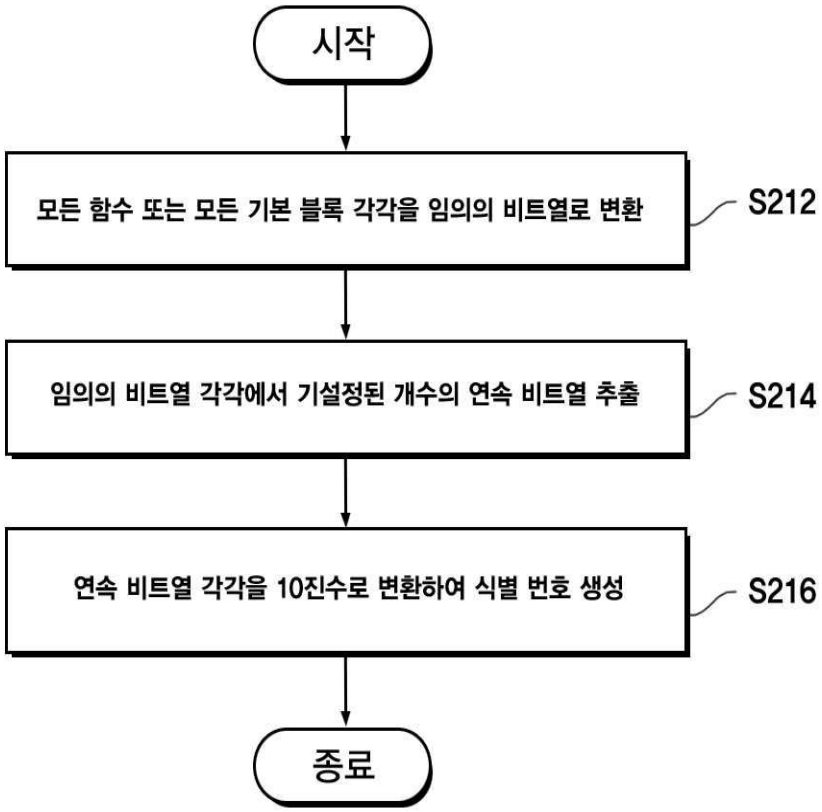
도면1


100
제1 프로그램
(1)
제2 프로그램
(2)
식별 번호 생성부
(110)
특성 비트 벡터 생성부
(120)
프로그램 유사도 산출부
(130)

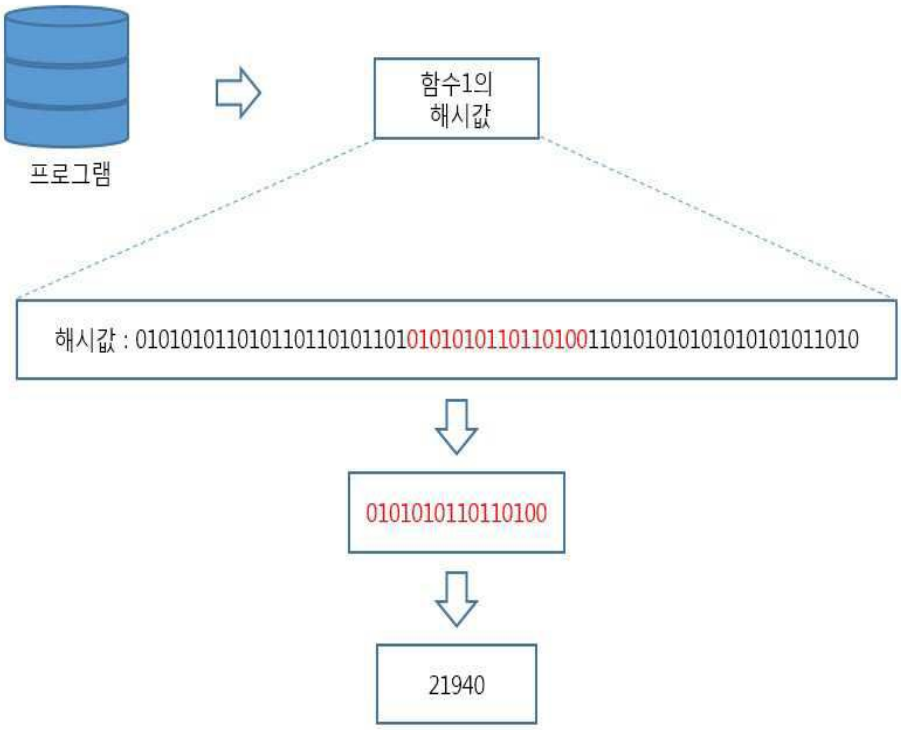
도면2


시작
모든 함수 또는 모든 기본 블록 각각에 대응되는 식별 번호 생성
S210
식별 번호 각각에 대응되는 비트의 비트값을 설정하여 2개의 특성 비트 벡터 생성
S220
2개의 프로그램 상호간의 유사도를 산출
S230
종료

도면3


시작
모든 함수 또는 모든 기본 블록 각각을 임의의 비트열로 변환
S212
임의의 비트열 각각에서 기설정된 개수의 연속 비트열 추출
S214
연속 비트열 각각을 10진수로 변환하여 식별 번호 생성
S216
종료

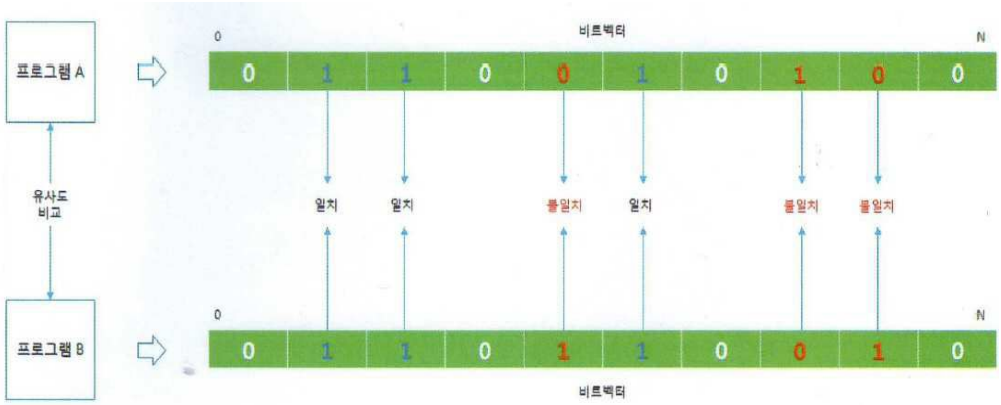
도면4


프로그램
함수1의
해시값
해시값 : 01010101101011011010110101010101101101001101010101010101011010
0101010110110100
21940

도면5


0번째
21940번째
65535번째
비트벡터
0 0 0 0 1 0 0 0 0 0

도면6


프로그램 A
유사도 비교
프로그램 B
비트벡터
0
N
0 1 1 0 0 1 0 1 0 0
일치
일치
불일치
일치
불일치
불일치
0 1 1 0 1 1 0 0 1 0
비트벡터