(12) 특허협력조약에 의하여 공개된 국제출원

(19) 세계지식재산권기구
국제사무국

(43) 국제공개일
2022 년 9 월 1 일 (01.09.2022)

WIPO | PCT

(10) 국제공개번호
WO 2022/182185 A1

(51) 국제특허분류:
G06F 11/36 (2006.01) *G06N 20/20* (2019.01)

(21) 국제출원번호: PCT/KR2022/002759

(22) 국제출원일: 2022 년 2 월 25 일 (25.02.2022)

(25) 출원언어: 한국어

(26) 공개언어: 한국어

(30) 우선권정보:
10-2021-0026393 2021 년 2 월 26 일 (26.02.2021) KR

(71) 출원인: 고려대학교 산학협력단 (**KOREA UNIVERSITY RESEARCH AND BUSINESS FOUNDATION**) [KR/KR]; 02841 서울특별시 성북구 안암로 145 (안암동 5가), Seoul (KR).

(72) 발명자: 전동훈 (**JUN, Donghun**); 05507 서울특별시 송파구 올림픽로 435, 206동 2501호, Seoul (KR). 오학주 (**OH, Hakjoo**); 06001 서울특별시 강남구 압구정로 151, 122동 1002호, Seoul (KR). 전민석 (**JEON, Min-Seok**); 05507 서울특별시 송파구 올림픽로 435, 206동 2501호, Seoul (KR).

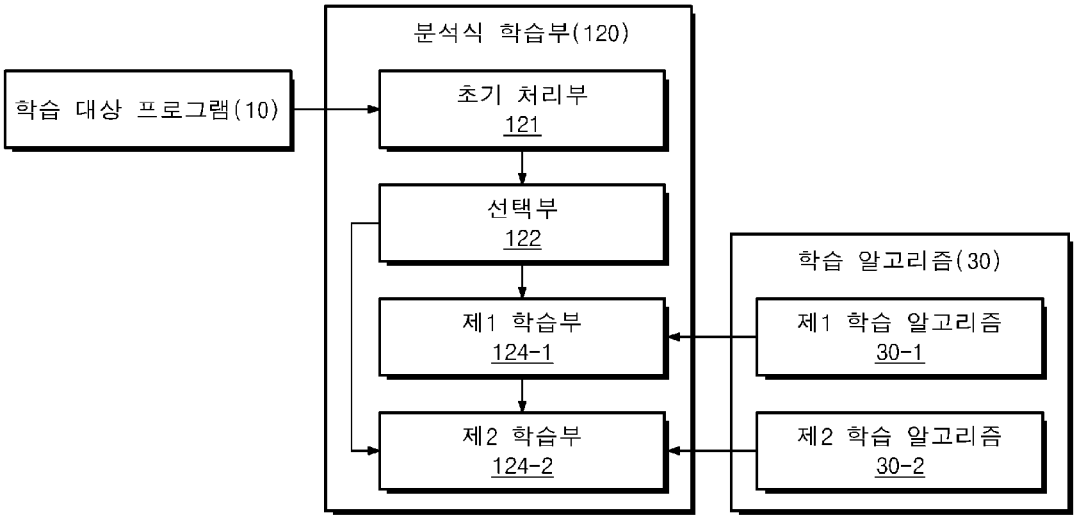
(74) 대리인: 김등용 (**KIM, Deungyong**); 08378 서울특별시 구로구 디지털로34길 55, 417호 동진국제특허법률사무소, Seoul (KR).

(81) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 국내 권리의 보호를 위하여): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC,

(54) **Title:** PROGRAM ANALYSIS DEVICE AND METHOD

(54) **발명의 명칭:** 프로그램 분석 장치 및 방법

[도3]



10 ... Program being learned
30 ... Learning algorithm
30-1 ... First learning algorithm
30-2 ... Second learning algorithm
120 ... Analysis equation learning unit
121 ... Initial processing unit
122 ... Selection unit
124-1 ... First learning unit
124-2 ... Second learning unit

(57) **Abstract:** The present invention relates to a program analysis device and method, wherein the program analysis device may comprise: a selection unit for selecting one of top-down processing or bottom-up processing for at least one analysis equation among a plurality of analysis equations; a first learning unit which, when the selection unit selects the top-down processing, performs top-down processing for the at least one analysis equation on the basis of a first learning algorithm in response to the selection, and thereby acquires at least one first learned analysis equation; and a second learning unit which, when the selection unit selects the bottom-up processing, performs bottom-up processing for the at least one analysis equation on the basis of a second learning algorithm in response to the selection, and thereby acquires at least one second learned analysis equation.

WO 2022/182185 A1

[다음 쪽 계속]

SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 역내 권리의 보호를 위하여): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 유라시아 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 유럽 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

공개:

— 국제조사보고서와 함께 (조약 제21조(3))

(57) 요약서: 프로그램 분석 장치 및 방법에 관한 것으로, 프로그램 분석 장치는 복수의 분석식 중 적어도 하나의 분석식에 대해서 하향식 처리 및 상향식 처리 중 어느 하나를 선택하는 선택부, 상기 선택부가 하향식 처리를 선택하는 경우, 이에 응하여 제1 학습 알고리즘을 기반으로 상기 적어도 하나의 분석식에 대한 하향식 처리를 수행하여 적어도 하나의 제1 학습된 분석식을 획득하는 제1 학습부 및 상기 선택부가 상향식 처리를 선택하는 경우, 이에 응하여 제2 학습 알고리즘을 기반으로 상기 적어도 하나의 분석식에 대한 하향식 처리를 수행하여 적어도 하나의 제2 학습된 분석식을 획득하는 제2 학습부를 포함할 수 있다.

명세서

발명의 명칭: 프로그램 분석 장치 및 방법

기술분야

[1] 프로그램 분석 장치 및 방법에 관한 것이다.

배경기술

[2] 프로그램 분석 방법은 크게 정적 분석 방법(static analysis) 및 동적 분석(dynamic analysis) 방법으로 구분될 수 있다. 정적 분석 방법은 프로그램을 실행시키지 않고 프로그램의 소스 코드를 분석하여 규칙 상의 오류나 보안 취약점 등을 검증 및 분석하는 방법이고, 동적 분석 방법은 프로그램을 실행시키면서 프로그램을 분석하는 방법이다.

[3] 포인터 분석(pointer analysis)은 프로그램 정적 분석 방법의 한 가지로, 프로그램 실행 전에 프로그램 실행 과정에서 포인터 변수나 표현식 등이 지시할 수 있는 모든 객체(들)을 파악하기 위해 이용된다. 이러한 포인터 분석의 결과는 버그 검출(Bug detecting), 프로그램 검증(Program Verification) 또는 프로그램 자동 수정(Automatic Program Repair) 등에 사용될 수 있다. 신속하고 정확한 포인터 분석을 위해서는 적절한 고성능의 분석 전략이 마련되어야 한다. 예를 들어, 문맥(context)의 고려 없이 분석을 수행하는 경우, 프로그램의 실제 실행 시에는 발생하지 않는 다수의 허위 알람이 분석 과정에서 검출될 수 있다. 이들 허위 알람은 개발자의 업무를 과중하게 하여 프로그램 개발의 자원 및 시간을 낭비하게 되는 원인이 되고 있다. 반대로 문맥을 고려하여 분석을 수행하는 경우, 검출되는 허위 알람은 상당히 감소하고 이와 더불어 분석의 정확성 및 실용성은 개선되는 장점이 있으나 반대로 프로그램의 분석에 소요되는 시간이 상당히 증가하게 되는 문제점도 존재하였다.

발명의 상세한 설명

기술적 과제

[4] 정확하면서도 보다 신속하게 저비용으로 프로그램에 대한 정적 분석을 자동으로 수행할 수 있는 프로그램 분석 장치 및 방법을 제공하는 것을 해결하고자 하는 과제로 한다.

과제 해결 수단

[5] 상술한 과제를 해결하기 위하여 프로그램 분석 장치 및 방법이 제공된다.

[6] 프로그램 분석 장치는 복수의 분석식 중 적어도 하나의 분석식에 대해서 하향식 처리 및 상향식 처리 중 어느 하나를 선택하는 선택부, 상기 선택부가 상기 하향식 처리를 선택하는 경우, 이에 응하여 제1 학습 알고리즘을 기반으로 상기 적어도 하나의 분석식에 대한 하향식 처리를 수행하여 적어도 하나의 제1 학습된 분석식을 획득하는 제1 학습부 및 상기 선택부가 상기 상향식 처리를 선택하는 경우, 이에 응하여 제2 학습 알고리즘을 기반으로 상기 적어도 하나의

분석식에 대한 하향식 처리를 수행하여 적어도 하나의 제2 학습된 분석식을 획득하는 제2 학습부를 포함할 수 있다.

[7] 프로그램 분석 방법은, 복수의 분석식 중 적어도 하나의 분석식에 대해서 하향식 처리 및 상향식 처리 중 어느 하나를 선택하는 단계 및 상기 선택부가 상기 상향식 처리를 선택하는 경우, 이에 응하여 제2 학습 알고리즘을 기반으로 상기 적어도 하나의 분석식에 대한 하향식 처리를 수행하여 적어도 하나의 제2 학습된 분석식을 획득하는 단계를 포함할 수 있다.

발명의 효과

[8] 상술한 프로그램 분석 장치 및 방법에 의하면, 주어진 적어도 하나 이상의 프로그램을 이용해 분석에 사용할 분석식들을 자동으로 생성할 수 있게 되고, 이 과정에서 생성된 분석식들을 적용하여 다른 프로그램들을 분석할 때 신속하고 정확한 분석을 할 수 있게 된다. 이에 따라 분석에 대한 비용(자원이나 노동력 등) 및 소요 시간을 보다 절감할 수 있는 효과를 얻을 수 있다.

[9] 상술한 프로그램 분석 장치 및 방법에 의하면, 프로그램 분석에 있어서 상대적으로 깊고 세밀하게 분석할 부분과 상대적으로 얕고 대략적으로 분석할 부분의 선별을 위한 다양한 분석 깊이(depth)의 분석식을 자동으로 생성할 수 있게 된다.

[10] 상술한 프로그램 분석 장치 및 방법에 의하면, 모든 문맥을 세밀하게 분석하는 수준으로 정확하면서도, 모든 문맥을 대략적으로 분석하는 수준만큼 신속하게 프로그램을 분석할 수 있는 장점도 얻을 수 있다.

[11] 상술한 프로그램 분석 장치 및 방법에 의하면, 효과적인 전략을 통해 빠르고 정확한 정적 분석을 수행할 수 있게 되는 장점이 존재한다.

도면의 간단한 설명

[12] 도 1은 프로그램 분석 장치의 일 실시예에 따른 블록도이다.

[13] 도 2는 순차적으로 분석식을 학습하는 과정의 일례를 설명하기 위한 도면이다.

[14] 도 3은 분석식 학습부의 일 실시예에 대한 블록도이다.

[15] 도 4는 분석식 학습부의 동작에 대응하는 프로그램 코드의 일례를 도시한 도면이다.

[16] 도 5는 제1 학습부의 동작에 대응하는 프로그램 코드의 일례를 도시한 도면이다.

[17] 도 6은 제2 학습부의 동작에 대응하는 프로그램 코드의 일례를 도시한 도면이다.

[18] 도 7은 분석부의 동작의 일례를 설명하기 위한 도면이다.

[19] 도 8은 프로그램 분석 방법의 일 실시예에 대한 흐름도이다.

발명의 실시를 위한 형태

[20] 이하 명세서 전체에서 동일 참조 부호는 특별한 사정이 없는 한 동일 구성요소를 지칭한다. 이하에서 사용되는 '부'가 부가된 용어는, 소프트웨어 또는

하드웨어로 구현될 수 있으며, 실시예에 따라 하나의 '부'가 하나의 물리적 또는 논리적 부품으로 구현되거나, 복수의 '부'가 하나의 물리적 또는 논리적 부품으로 구현되거나, 하나의 '부'가 복수의 물리적 또는 논리적 부품들로 구현되는 것도 가능하다. 명세서 전체에서 어떤 부분이 다른 부분과 연결되어 있다고 할 때, 이는 어떤 부분과 다른 부분에 따라서 물리적 연결을 의미할 수도 있고, 또는 전기적으로 연결된 것을 의미할 수도 있다. 또한, 어떤 부분이 다른 부분을 포함한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 부분 이외의 또 다른 부분을 제외하는 것이 아니며, 설계자의 선택에 따라서 또 다른 부분을 더 포함할 수 있음을 의미한다. 제1이나 제2 등의 용어는 하나의 부분을 다른 부분으로부터 구별하기 위해 사용되는 것으로, 특별한 기재가 없는 이상 이들이 순차적인 표현을 의미하는 것은 아니다. 또한 단수의 표현은 문맥상 명백하게 예외가 있지 않는 한, 복수의 표현을 포함할 수 있다.

[21] 이하 도 1 내지 도 7을 참조하여 프로그램 분석 장치의 일 실시예에 대해서 설명하도록 한다.

[22] 도 1은 프로그램 분석 장치의 일 실시예에 따른 블록도이다.

[23] 도 1에 도시된 바를 참조하면, 프로그램 분석 장치(100)는, 입출력부(101), 프로세서(110) 및 저장부(190)를 포함할 수 있으며, 이들 입출력부(101), 프로세서(110) 및 저장부(190) 중 적어도 둘은 명령/지시나 데이터 등을 전기적 신호의 형태로 전달할 수 있도록 마련된 것일 수 있다.

[24] 입출력부(101)는, 사용자, 설계자 또는 외부의 다른 전자 장치(미도시) 등으로부터 프로그램 분석 장치(100)의 동작에 필요한 명령/지시, 데이터 또는 프로그램(앱, 애플리케이션 또는 소프트웨어로 지칭될 수 있음) 등을 수신하거나 및/또는 프로세서(110)의 동작에 따른 학습 결과, 분석 결과 또는 프로그램 등을 사용자나 설계자에게 시각적 또는 청각적으로 출력하거나 또는 외부의 다른 전자 장치로 전기적 신호의 형태로 유선 또는 무선 통신 네트워크를 통해 전달할 수 있다. 예를 들어, 입출력부(101)는 저장부(190)에 저장된 학습 대상 프로그램(10), 분석 대상 프로그램(20), 적어도 하나의 학습 알고리즘(30) 또는 학습용 특질(feature, a1, a2, …, an) 등을 사용자나 다른 전자 장치로부터 동시에 또는 순차적으로 수신하거나 및/또는 학습된 분석식(80)이나 학습 결과(90) 등을 사용자에게 출력하거나 다른 전자 장치로 전달할 수 있다. 입출력부(101)는 프로그램 분석 장치(100)와 일체형으로 마련된 것일 수도 있고, 물리적으로 분리 가능하게 마련된 것일 수도 있다. 입출력부(101)는, 예를 들어, 키보드, 마우스, 태블릿, 터치 스크린, 터치 패드, 트랙 볼, 트랙패드, 스캐너 장치, 영상 촬영 모듈, 동작 감지 센서, 감압 센서, 근접 센서 및/또는 마이크로 폰 등의 입력 장치를 포함할 수도 있고, 디스플레이, 프린터 장치, 스피커 장치 및/또는 영상 출력 단자 등의 출력 장치를 포함할 수도 있다. 또한 실시예에 따라, 입출력부(101)는, 외부의 다른 장치(일례로 휴대용 메모리 장치 등)로부터 데이터 등의 수신이 가능한 데이터 입출력 단자나, 외부의 다른 장치와 유무선 통신 네트워크를 통해

연결되는 통신 모듈(일례로 랜 카드, 근거리 통신 모듈 또는 이동통신 모듈 등) 등을 포함할 수도 있다. 입출력부(101)는 필요에 따라 생략 가능하다.

[25] 프로세서(110)는 분석 대상 프로그램(20)에 대한 분석 수행을 위한 적어도 하나의 분석식(80)을 획득하거나 및/또는 획득한 적어도 하나의 분석식(80)을 이용하여 분석 대상 프로그램(20)에 대한 분석(일례로 정적 분석)을 수행할 수 있다. 실시예에 따라서, 프로세서(110)는 적어도 하나의 분석식(80) 획득 과정만을 수행할 수도 있다. 이 경우, 획득된 분석식(80)은 입출력부(101) 등을 통해 다른 전자 장치로 전송된 후, 다른 전자 장치에 의해 프로그램의 분석에 이용될 수도 있다. 또한, 프로세서(110)는, 적어도 하나의 분석식(80) 기반의 프로그램(20) 분석 과정만을 수행할 수도 있다. 이때, 적어도 하나의 분석식(80)은 다른 전자 장치에 의해 획득된 것일 수도 있다.

[26] 일 실시예에 의하면, 프로세서(110)는 적어도 하나의 학습 알고리즘(30)을 기반으로 하나 이상의 학습된 분석식(80)을 획득하는 분석식 학습부(120) 및 학습 알고리즘(30)에 의해 획득된 적어도 하나의 학습된 분석식(80)을 기반으로 분석을 수행하는 분석부(140) 중 적어도 하나를 포함할 수 있다. 여기서, 분석식 학습부(120) 및 분석부(140)는 실시예에 따라 논리적으로 분리된 것일 수도 있고, 물리적으로 분리된 것일 수도 있으며, 논리적으로 분리된 경우, 분석식 학습부(120) 및 분석부(140)는 각각 프로그램 코드로 구현되되, 각각의 프로그램 코드는 하나 또는 둘 이상의 프로그램 패키지에 더불어 또는 분리되어 포함된 것일 수 있다. 물리적으로 분리된 경우, 분석식 학습부(120) 및 분석부(140)는 각각 별도의 반도체 칩을 채용하여 구현된 것일 수도 있다. 분석식 학습부(120) 및 분석부(140) 각각의 상세한 동작 및 기능에 대한 자세한 설명은 후술한다. 프로세서(110)는, 예를 들어, 중앙 처리 장치(CPU, Central Processing Unit), 마이크로 컨트롤러 유닛(MCU, Micro Controller Unit), 애플리케이션 프로세서(AP, Application Processor), 마이컴(Micom, Micro Processor), 전자 제어 유닛(ECU, Electronic Controlling Unit) 및/또는 각종 연산 처리 및 제어 신호의 생성이 가능한 다른 전자 장치 등을 기반으로 구현될 수 있다. 이들 장치는, 예를 들어, 하나 또는 둘 이상의 반도체 칩과 관련 부품 등을 채용하여 제작 가능하다.

[27] 저장부(190)는 프로그램 분석 장치(100)의 동작에 필요한 데이터나 프로그램 등을 일시적으로 또는 비일시적으로 저장할 수 있다. 예를 들어, 저장부(190)는, 분석식 학습부(120)에 의해 분석식(80)의 학습 및 생성을 위해 이용되는 적어도 하나의 학습 대상 프로그램(10)과, 분석부(140)에 의해 오류의 존재 여부 등이 분석되는 분석 대상 프로그램(20)과, 분석식(80)의 학습을 위해 이용되는 소정의 학습 알고리즘(30)과, 학습 결과 획득된 적어도 하나의 학습된 분석식(80)과, 분석 대상 프로그램(20)에 대한 분석 결과(90) 및/또는 분석식(80)의 학습 및 생성을 위해 이용되는 적어도 하나의 특질(a1, a2, …, an) 등을 일시적으로 또는 비일시적으로 저장할 수 있다. 실시예에 따라 적어도 하나의 학습 대상 프로그램(10)과 분석 대상 프로그램(20)은 동일할 수도 있고 상이할 수도 있다.

적어도 하나의 학습 대상 프로그램(10)이 분석 대상 프로그램(20)인 경우, 분석 대상 프로그램(20)은 생략 가능하다. 실시예에 따라서, 저장부(190)는 이들(10 내지 90) 중 일부의 데이터나 알고리즘을 저장하지 않을 수도 있고, 및/또는 이들 외에도 다른 더 다양한 데이터나 알고리즘을 저장할 수도 있다. 또한, 저장부(190)는, 프로세서(110)의 동작을 위한 적어도 하나의 설정이나 명령/지시를 저장하거나, 프로세서(110)의 동작을 위한 프로그램 또는 프로그램 패키지를 저장할 수 있다. 프로세서(110)의 동작을 위한 프로그램 또는 프로그램 패키지는, 설계자에 의해 직접 작성되거나 또는 수정된 후 저장부(190)에 저장된 것일 수도 있고, 메모리 장치나 디브이디 등의 기록 매체를 통해 저장부(190)에 저장된 것일 수도 있으며, 및/또는 유선 또는 무선 통신 네트워크를 통해 접속 가능한 전자 소프트웨어 유통망을 통하여 획득 또는 갱신된 것일 수도 있다. 상술한 저장부(190)는 예를 들어, 주기억장치 및 보조기억장치 중 적어도 하나를 포함할 수 있다. 주기억장치는 롬(ROM) 및/또는 램(RAM) 등을 포함할 수 있고, 보조기억장치는, 플래시 메모리 장치, SD(Secure Digital) 카드, 솔리드 스테이트 드라이브(SSD, Solid State Drive), 하드 디스크 드라이브(HDD, Hard Disc Drive), 자기 드럼, 컴팩트 디스크(CD), 디브이디(DVD) 및/또는 플로피 디스크 등과 같이 데이터를 영구적 또는 반영구적으로 저장 가능한 적어도 하나의 기록 매체를 이용하여 구현될 수 있다.

[28] 상술한 프로그램 분석 장치(100)는, 하나의 정보 처리 장치 또는 둘 이상의 동일 또는 상이한 정보 처리 장치를 조합하여 구현될 수 있다. 여기서, 정보 처리 장치는, 예를 들어, 데스크톱 컴퓨터, 랩톱 컴퓨터, 스마트 폰, 태블릿 피씨, 스마트 시계, 두부 장착형 디스플레이(HMD: Head Mounted Display) 장치, 내비게이션 장치, 휴대용 게임기, 개인용 디지털 보조기(PDA: Personal Digital Assistant), 디지털 텔레비전, 셋 톱 박스, 인공 지능 음향 재생 장치(인공 지능 스피커), 가전 기기(냉장고 또는 세탁기 등), 유인 이동체(승용차, 버스나 이륜차 등의 차량 등), 무인 이동체(로봇 청소기 등), 유인 비행체, 무인 비행체(드론 등), 가정용 또는 산업용 로봇, 산업용 기계, 전자 칠판, 전자 광고판 또는 자동 입출금기(ATM: Automated Teller Machine) 등을 포함할 수 있으나, 이에 한정되는 것은 아니다. 설계자나 사용자 등은 상황이나 조건에 따라서 상술한 장치 이외에도 정보의 연산 처리 및 제어가 다양한 장치 중 적어도 하나를 상술한 프로그램 분석 장치(100)로 고려 및 적용할 수 있다.

[29] 또한, 프로그램 분석 장치(100)는, 외부의 다른 장치와 상호 통신을 수행하여, 다른 장치로부터 학습 대상 프로그램(10), 분석 대상 프로그램(20) 또는 학습 알고리즘(30)을 포함하는 프로그램 등을 수신하거나, 및/또는 분석 결과나 프로그램 등을 다른 장치로 전송할 수 있도록 마련되되, 프로그램 분석 장치(100)와 다른 장치와의 통신은 유선 통신 네트워크, 무선 통신 네트워크 또는 이들의 조합을 기반으로 수행될 수 있다. 여기서, 무선 통신 네트워크는 근거리 통신 네트워크 기술 및 원거리 통신 네트워크 기술 중 적어도 하나를 기반으로

하되, 예를 들어, 근거리 통신 네트워크 기술은 와이파이(Wi-Fi), 와이파이 다이렉트(Wi-Fi Direct), 블루투스(Bluetooth), 지그비(zigbee) 통신, 캔(CAN) 통신, 알에프아이디(RFID: Radio-Frequency IDentification) 및/또는 엔에프씨(NFC: Near Field Communication) 등을 포함할 수 있고, 원거리 통신 네트워크 기술은, 예를 들어, 3GPP, 3GPP2, 와이브로 또는 와이맥스 계열 등의 이동 통신 표준을 포함할 수 있다.

[30] 이하 도 2 내지 도 6을 참조하여, 분석식 학습부(120)의 동작을 보다 구체적으로 설명하도록 한다.

[31] 도 2는 순차적으로 분석식을 학습하는 과정의 일례를 설명하기 위한 도면이다.

[32] 일 실시예에 의하면, 프로세서(110)의 분석식 학습부(120)는, 도 2에 도시된 바와 같이, 하나 또는 둘 이상의 학습 전의 분석식 $f_i(f_0, f_1, f_2$ 등)가 주어지면(이 경우, 각 분석식 f_0, f_1, f_2는, 일례로 false 등의 기 정의된 초기값으로 설정되어 있을 수도 있음), 이들 각각의 분석식 $f_i(f_0, f_1, f_2$ 등)에 대한 학습을 적어도 일 회 수행하여 각각의 분석식 $f_i(f_0, f_1, f_2$ 등)가 적어도 하나의 특질$(a_1, a_2, \ldots, a_n)$을 포함하여 형성되도록 할 수 있다.

[33] 여기서, 분석식 f_i은 분석 깊이가 i만큼 주어져야 하는 프로그램 구성 요소 (예: 함수)를 나타낸다. 예를 들어, 분석식 f_i는 i의 분석 깊이로 분석 하게 될 함수의 집합을 의미할 수 있다. 각각의 분석식 f_0, f_1, f_2은 서로 간에 공통되는 함수가 존재하지 않으며, 예를 들어, 분석식 f_0에 포함된 함수는 다른 분석식 f_1 및 f_2에 포함되지 않는다는 것을 의미한다($f_0 \cap f_1 \cap f_2 = \varphi$).

[34] 분석식 f_i는, 예를 들어, 논리식 모델(Disjunctive model)을 이용하여 제작된 논리식을 포함할 수 있다. 구체적으로 분석식 $f_i(f_0, f_1, f_2$ 등)은 특질$(a_1, a_2, \ldots, a_n)$이 주어졌을 때, 하기의 수학식 1과 같이 하나 이상의 특질들$(a_1, a_2, \ldots, a_n)$ 중 하나의 특질$(a_1, a_2, \ldots, a_n)$을 단독으로 또는 둘 이상의 특질$(a_1, a_2, \ldots, a_n)$을 조합해서 생성될 수도 있다.

[35] [수식1]

$$f \rightarrow true \mid false \mid a_i \in features \mid \neg f \mid f_1 \wedge f_2 \mid f_1 \vee f_2$$

[36] 여기서 특질$(a_1, a_2, \ldots, a_n)$은 프로그램의 구성 요소(예: 자바의 메소드(method))등을 포함할 수도 있다. 이하 같다)가 어떠한 특질을 가지고 있는지 여부를 확인 또는 질의하는 함수로 하기의 수학식 2와 같이 정의될 수 있다.

[37] [수식2]

$$a_i(P) : M_P \rightarrow \{true, false\}$$

[38] 구체적으로 예를 들어, "출력 값(output)이 void"라는 특질 a에 대해서, $[[a]]_p$는 프로그램 P에서 출력 값이 void인 메소드의 집합을 의미한다. 수식이나 기호가 부가된 특질(즉, 복합 특질)은, 각각의 개별 특질만으로 표현하기 어려운 함수의 성질을 나타낼 수 있다. 이는 선언문의 형태로 표현 가능하다. 예를 들어, 복합

특질 $[[\neg a]]_p$는 특질 a를 가지고 있지 않은 함수의 집합(일례로 출력 값이 void가 아닌 함수의 집합)을 의미하도록 정의된 것일 수 있고, $[[a_i \wedge a_j]]$는 a_i와 a_j 두가지 특질을 모두 갖는 함수의 집합을 의미하도록 정의된 것일 수 있으며, 및/또는 $[[a_i \vee a_j]]$는 a_i와 a_j 두가지 특질 중 적어도 하나를 갖는 함수의 집합을 의미하도록 정의된 것일 수 있다. 이들 외에도 설계자나 사용자 등은 임의적 선택에 따라 새로운 복합 특질이 더 정의할 수 있으며, 또한 상술한 각각의 복합 특질을 상술한 바와 상이하게 정의할 수도 있다.

[39] 일 실시예에 의하면, 프로세서(110)는 모든 또는 일부의 분석식 f_0, f_1, f_2에 대한 학습을 동시에 수행할 수도 있다. 다른 실시예에 의하면, 프로세서(110)는 각각의 분석식 f_0, f_1, f_2에 대한 학습을 순차적으로 하나씩 수행할 수도 있다. 구체적으로 예를 들어, 프로세서(110)는 어느 하나의 분석식 f_0에 대한 학습을 수행하고, 다른 하나의 분석식 f_2에 대한 학습을 수행하고, 마지막으로 또 다른 하나의 분석식 f_1에 대한 학습을 수행할 수도 있다. 각각의 분석식 f_0, f_1, f_2에 대한 학습을 순차적으로 수행하는 것은 학습을 수행할 탐색 공간을 감소시키는 효과가 있다. 예를 들어, 학습을 통해 분석식 f_0 내지 f_k(k는 0 이상의 정수)를 획득하되, 어느 하나의 분석식 f_a의 연산 및 획득을 위해 탐색해야 하는 공간이 |S|인 경우에 있어서, (k+1)개의 분석식을 한 번에 구하기 위한 탐색 공간은 $|S|^{k+1}$이다. 그런데, 이러한 탐색 공간 $|S|^{k+1}$은 지나치게 범위가 넓어 최적의 분석식 검출을 상당히 어렵게 만든다. 그러나, 이와 상이하게 한 번에 하나의 분석식만을 구하는 과정을 (k+1) 회 반복하여 하나 이상의 분석식 f_0 내지 f_k를 순차적으로 획득하는 경우에는, 탐색 공간은 (k+1)*|S|로 주어지게 된다. 따라서, 분석식 획득을 위한 탐색 공간이 상당히 감소하게 되고, 결과적으로 탐색 효율이 상승하고, 자원, 시간 및 비용이 절감될 수 있게 된다.

[40] 도 3은 분석식 학습부의 일 실시예에 대한 블록도이고, 도 4는 분석식 학습부의 동작에 대응하는 프로그램 코드의 일례를 도시한 도면이다. 도 4에서 procedure LEARN(F, P, K, A) 함수(메소드)는 분석식 학습부(120)의 동작을 나타내는 함수의 프로그램 코드의 일 작성례로, 여기서, F는 학습에 사용할 정적분석기를, P는 프로그램을, k는 최대 분석 깊이를, A는 특질(a_1, a_2, …, a_n)의 집합을 의미한다.

[41] 상술한 분석식 f_0 내지 f_k의 획득을 위해서 분석식 학습부(120)는 도 3에 도시된 바와 같이, 초기 처리부(121), 선택부(122), 제1 학습부(124-1) 및 제2 학습부(124-2)를 포함할 수 있다.

[42] 초기 처리부(121)는, 먼저 도 4에 도시된 바와 같이 학습 대상 프로그램(10)에 관련하여 학습될 모든 또는 일부의 분석식 f_0 내지 f_k를 미리 정의된 값(일례로 false)으로 초기화시킬 수 있다(제2 줄). 또한, 초기 처리부(121)는 함수 내에서 이용될 적어도 하나의 변수(top, bot)에 미리 정의된 초기값을 할당할 수 있다(제3 및 4줄). 예를 들어, 변수 top 및 bot 각각이 k 및 0으로 초기화될 수 있다. 여기서, 변수 top은 제1 학습부(124-1)에 의해 수행되는 각각의 하향식 처리(top-down)를

식별하기 위한 변수로, 하향식 처리가 수행될 때마다 1씩 감소하도록 마련된 것일 수 있다(제10줄 참조). 또한, 변수 bot는 제2 학습부(124-2)에 의해 수행되는 각각의 상향식 처리(bottom-up)를 식별하기 위한 변수로, 상향식 처리가 수행될 때마다 1씩 증가하도록 설계될 수 있다(제14줄 참조).

[43] 이와 동시에 또는 순차적으로 선택부(122)는 학습 대상이 되는 분석식 f_0 내지 f_k 중에서 아직 학습 처리가 수행되지 않아 획득되지 않은 적어도 하나의 분석식(f_{bot} 또는 f_{top})을 선택할 수 있다(제6 줄). 미처리된 적어도 하나의 분석식(f_{bot} 또는 f_{top})은 상술한 바와 같이 false 값으로 초기화된 것일 수 있다. 이어서 선택부(122)는 이들에 대해서 상향식 처리를 수행할지 또는 하향식 처리를 수행할지 여부를 결정할 수 있다(제7 줄). 다시 말해서, 선택부(122)는 학습 대상 분석식(f_{bot} 또는 f_{top})를 제1 학습부(124-1)에서 처리할지 또는 제2 학습부(124-2)에서 처리할지 여부를 결정할 수 있다.

[44] 일 실시예에 의하면, 선택부(122)는 먼저 주어진 논리식 파라미터(Π)에 대해서 현제 가능한 가장 정교한 추상화를 갖는 분석 방법을 실행시켰을 때 주어진 하향식 처리(ψ_{top})가 가능한 문제 해결 수단인지 여부를 추정할 수 있다. 여기서 함수의 논리식 파라미터(Π)는 하기의 수학식 3으로 주어질 수 있다.

[45] [수식3]

$$\Pi = \langle f_0, f_1, \cdots, f_{top-1}, f_{rem}, f_{top+1}, \cdots, f_k \rangle$$

[46] 하향식 처리(ψ_j)는 미 학습 논리식 $\langle f_i, f_{i+1}, \ldots, f_j \rangle$ 중에서 분석 깊이가 가장 깊은 논리식 f_j를 획득하기 위해 수행되는 처리로, 분석식 f_j에 대한 하향식 처리(ψ_j)는, 예를 들어, 하기의 수학식 4와 같이 주어질 수 있다.

[47] [수식4]

$$\Psi_j \equiv Find\ f_j\ minimizing \sum_{p \in P} cost(F_p(H_{\Pi_j}(p)))\ subject\ to\ Precise(\Pi_j, F, P)$$

[48] 여기서

$Precise(\Pi_j, F, P)$

함수는 획득된 파라미터

Π_j

가 학습 데이터에 대해 충분히 정교한지 여부를 판단하는 함수이다. 이 경우, 분석 깊이가 정해지지 않은 함수들은 j-1의 깊이를 갖는 것으로 처리되어 학습한다.

$Precise(\Pi_j, F, P)$

는 예를 들어 하기의 수학식 5와 같이 정의될 수 있다.

[49] [수식5]

$$Precise(\Pi_j, F, P) = \frac{\sum_{p \in P} |proved(F_p(H_{\Pi_j}(p)))|}{\sum_{p \in P} |proved(F_p(k))|} \geq \gamma$$

[50] 즉,

$Precise(\Pi_j,F,P)$

는

Π_j

와 관련하여 증명된 쿼리와 모든 프로그램 구성요소(예: 함수)에 깊이 k를 적용하여 가능한 가장 큰 정확도를 가지는 파라미터(k)와 관련하여 증명된 쿼리의 비율이 소정의 임계 값(γ: γ∈[0,1])보다 큰 경우에 만족되는 함수이다. 한편, 휴리스틱 함수

H_{Π_j}

는 임의의 프로그램 P에 존재하는 각각의 프로그램 구성 요소(예: 함수)마다 분석 깊이를 정해주는 함수로, 예를 들어 하기의 수학식 6으로 정의될 수 있다.

[51] [수식6]

$$H_{\Pi}(P) \equiv \lambda m \in M_P \begin{cases} k & \text{if } m \in [[f_k]]_P \\ k\text{-}1 & \text{if } m \in [[f_{k-1}]]_P \\ \cdots & \\ k\text{-}i & \text{if } m \in [[f_{k-i}]]_P \\ \cdots & \\ 1 & \text{if } m \in [[f_1]]_P \\ 0 & \text{if } m \in [[f_0]]_P \end{cases}$$

[52] 다시 말해서, 하향식 처리(ψ_j)는 상대적으로 큰 j에 대해서(즉, 분석 깊이가 상대적으로 깊은 분석식 f_j에 대해서) 휴리스틱 함수

H_{Π_j}

의 성능이 최대화되는 f_j를 찾는 것이다. 휴리스틱 함수

H_{Π_j}

의 성능 최대화를 위해서는 f_j가 의미하는 함수의 집합이 정확도를 유지하는 한도 내에서 최소가 되는 분석식을 찾아내는 것으로 귀결된다.

[53] 이를 위해서 선택부(122)는, 만약 분석식에 대한 처리가 주어진 비용 임계 값(C) 범위 내에서 종료된다면, 소정의 값(일례로 true)을 리턴 시키고, 반대로 분석 및 처리가 주어진 비용 임계 값(C) 범위 내에서 종료될 수 없다면, 선택부(122)는 소정의 값과 상이한 다른 값(일례로 false)을 리턴시킨다. 이러한 동작은 적어도 하나의 함수(일례로 ChooseProblem(top, Π, F, P, f_{rem}))을 이용하여 수행될 수 있다. 여기서, ChooseProblem(top, Π, F, P, f_{rem}) 함수는 하기의 수학식 7과 같이 표현될 수 있다.

[54] [수식7]

$$ChooseProblem\ (top,\Pi,F,P,f_{rem}) = \begin{cases} true & \sum cost(F_P(H_{\Pi_j}(P)) < C \\ false & otherwise \end{cases}$$

[55] 만약 소정의 값, 즉 true가 출력되면, 하향식 처리(ψ_j)가 처리 가능한 것으로 판단되고(제8 줄), 주어진 함수 f_{top}에 대해 하향식 처리(ψ_{top})가 수행된다(제9 줄). 일 실시예에 따르면, 하향식 처리(ψ_{top})는 제1 처리부(124-1)에 의해 수행될 수 있으며, 제1 처리부(124-1)는 제1 학습 알고리즘(30-1)을 이용하여 하향식 처리(ψ_{top})를 수행할 수도 있다. 이에 따라 하향식 처리(ψ_{top})에 대응하는 학습된 분석식(이하 제1 학습된 분석식)이 획득될 수 있게 된다. 여기서 위 문제에서 깊이(depth)가 정해지지 않는 함수($m \not\in f_0 \vee f_1 \vee \cdots \vee f_k$)는, 일 실시예에 있어서, j-1의 깊이를 가지도록 설정되어 처리될 수 있다. 한편, 하향식 처리(ψ_{top})가 수행되면, 변수 top에는 기존의 변수 top의 값에 1을 차감한 값이 입력된다. 이에 따라서 상대적으로 낮은 분석 깊이(j-1)의 분석식에 대한 학습이 수행될 수 있게 된다. 즉, 분석식 f_j에 대한 학습이 수행되고 이후의 선택 과정에 있어서 동일하게 하향식 처리(ψ_{j-1})가 선택되면, 이에 응하여 분석식 f_{j-1}에 대한 학습이 수행될 수 있다. 결과적으로 하향식으로(즉, 분석 깊이가 얕아지는 방향으로) 적어도 하나의 분석식(f_m, , m= j, j-1, j-2, …)이 처리(ψ_m)되고, 이에 대응하는 적어도 하나의 제1 학습된 분석식이 획득된다.

[56] 반대로 만약 소정의 값과 상이한 다른 값, 즉 false가 출력되면, 하향식 처리(ψ_{top})는 처리 불가능한 것으로 판단되고, 다른 처리(즉, 상향식 처리(ψ_{bot}))가 선택 수행된다(제11 내지 14줄). 상향식 처리(ψ_{bot})가 수행되면 그에 따라 분석식 f'_{top}이 리턴될 수 있다(제12 줄). f'_{top}의 부정($\neg f'_{top}$) 및 f_{rem}(f_{rem}은 f_0 이내 f_k와 서로소인 논리식(제 6줄)) 값의 교집합($f_{rem} \wedge \neg f'_{top}$)이 변수 f_{bot}에 입력된다(제13 줄). 따라서, f'_{top}의 부정 값은 f_{bot}으로 이용되는 다른 분리식들과 분리된다. 상향식 처리(ψ_{bot})가 수행되면, 이에 응하여 변수 bot에는 기존의 변수 bot의 값에 1을 부가한 값이 입력된다(제14 줄). 이에 따라서 상대적으로 분석 깊이가 더 깊은 분석식에 대한 학습이 수행될 수 있게 된다. 즉, 상대적으로 분석 깊이(i)가 얕은 분석식 f_j에 대한 학습이 수행되어 학습된 분석식(이하 제2 학습된 분석식)이 획득된 후, 이후의 선택 과정에 있어서 상향식 처리(ψ_{j-1})가 선택되면, 분석식 f_{i+1}에 대한 학습이 수행되고 이에 응하여 다른 제2 학습된 분석식이 획득될 수 있다. 결과적으로 상방향으로(즉, 분석 깊이가 증가하는 방향으로) 적어도 하나의 분석식(f_n, , m= i, i+ 1, i+2, …)이 처리(ψ_m)되고, 이에 대응하는 하나 또는 둘 이상의 제2 학습된 분석식이 획득될 수 있게 된다. 상향식 처리(ψ_{bot})는, 제2 처리부(124-2)에 의해 처리될 제2 학습 알고리즘(30-2)를 기반으로 수행 가능하며, 제2 학습 알고리즘(30-2)는, 예를 들어, 후술하는 SPEC2GEN 함수를 포함할 수 있다.

[57] 일 실시예에 따르면, 상술한 상향식 처리(ψ_i)는 하기의 수학식 8을 연산함으로써 획득되도록 설계될 수도 있다.

[58] [수식8]

$$\psi_i \equiv Find\ f_i\ minimizing \sum_{p \in P} cost(F_p(H_{\Pi_i}(p)))\ \ subject\ to\ Precise(\Pi_i, F, P)$$

[59] 상향식 처리(ψ_i)는 하향식 처리(ψ_j)와는 반대로 상대적으로 작은 i에 대해서(즉, 분석 깊이가 상대적으로 얕은 분석식 f_i에 대해서) f_i가 의미하는 함수의 집합이 정확도를 유지하는 한도 내에서 최대가 되는 분석식이 휴리스틱 함수

H_{Π_j}

의 성능을 극대화한다. 이 경우, 분석 깊이가 정해지지 않은 함수들은 j의 깊이를 갖는 것으로 처리되어 분석된다.

[60] 상술한 과정은 변수 bot 및 top이 서로 동일하게 될 때까지(즉, top=bot) 반복될 수 있다(제5 줄 및 제16줄 참조). 반복의 과정에서 선택부(122)는 상황에 따라 하향식 처리(ψ_{top})를 반복 선택할 수도 있고, 상향식 처리(ψ_{bot})를 반복 선택할 수도 있으며, 하향식 처리(ψ_{top})와 상향식 처리(ψ_{bot})를 교차 선택할 수도 있다. 따라서, 전 과정에서 하향식 처리(ψ_{top})가 선택되었다고 하더라도, 후 과정에서 반드시 하향식 처리(ψ_j) 또는 상향식 처리(ψ_{bot})가 선택되는 것은 아니다.

[61] 도 5는 제1 학습부의 동작에 대응하는 프로그램 코드의 일례를 도시한 도면이다.

[62] 제1 학습부(124-1)는 상술한 바와 같이 하향식 처리(ψ_{top})를 수행할 수 있다. 이 경우, 하향식 처리(ψ_{top})는 일반적인 분석식에서 점차로 특정 논리식으로 학습을 수행하는 알고리즘을 기반으로 구현된 것일 수 있다.

[63] 하향식 처리(ψ_{top})는 기 정의된 제1 학습 알고리즘(30-1)을 통하여 수행되되, 제1 학습 알고리즘(30-1)는, 도 5에 도시된 일 실시예와 같이, GEN2SPEC(bot, top, <f_0, f_1, …, f_k>, F, P, A, f_{rem}) 함수를 포함할 수 있다. 구체적으로 GEN2SPEC 함수는 먼저 개별 특성(a_j)의 집합(A) 및 이들의 부정 값($\neg a_j$)을 포함하는 대상 집합을 설정하고(제2 줄), 분석식 f가 대상 집합의 개별 특성(a_i) 또는 적어도 하나의 개별 특성(들)(a_j)의 논리합을 포함하도록 하여 가장 일반적인 초기 분석식으로 생성하도록 마련된다(제3 줄). 이어서 분석식 f는 작업 리스트(W)에 기록된다(제4 줄). 만약 작업 리스트(W)가 공집합이 아니라면, 작업 리스트(W)로부터 정제될 구절(clause) c가 선택되고, 작업 리스트(W)로부터 제거된다(제6 줄 및 제7 줄). 이 경우, c의 선택은 하기와 같이 가장 많은 비용을 소모하는 구절 c를 리턴 하도록 설계된 수학식 9를 기반으로 수행될 수 있다.

[64] [수식9]

$$MostExpensiveClause\ (W,F,P) = \underset{c\in W}{argmin} \sum_{p\in P} cost(F_p(H_{\Pi_c}(p)))$$

[65] 순차적으로 특성 a가 획득될 수 있다(제8 줄). 특성 a는 구절 c와 가장 동일 또는 유사한 특성을 의미하며, 예를 들어, 하기의 수학식 10를 이용하여 획득될 수 있다.

[66] [수식10]

$$ChooseAtom(A,c) = \begin{cases} argmax_a\in A\{c\} \sum_{p\in P} |[c\wedge a]_p| & \text{if } A\{c\}\neq\phi \\ false & otherwise \end{cases}$$

[67] 순차적으로 구절 c 및 특성 {a}을 조합하여 정제된 구절 c′를 획득하고(제9 줄), 정제된 구절 c′를 기반으로 기존의 구절 c를 대체하여 새로운 분석식 f_{top}를 획득한다(제10 줄). 이어서 각 함수 f_0 내지 f_k 간의 연결을 분리하여 새로운 파라미터(Π)을 결정하고(제11 줄), 상술한 precise(Π_j, F, P) 함수를 이용하여 함수의 집합(Π)이 정교한지 여부를 결정한다(제12 줄). 만약 파라미터(Π)가 정교하다면, 분석식 f는 분석식 f_{top}에 의해 갱신되고(제13 줄), 작업 리스트(W)에는 정제된 구절 c′이 추가된다(제14 줄). 그렇지 않다면, 분석식 f 및 작업 리스트(W)의 갱신은 수행되지 않는다. 이와 같은 과정은 작업 리스트(W)가 공집합이 될 때까지 반복된다.

[68] 도 6은 제2 학습부의 동작에 대응하는 프로그램 코드의 일례를 도시한 도면이다.

[69] 제2 학습부(124-2)는 상술한 바와 같이 상향식 처리(ψ_{bot})를 수행할 수 있다. 여기서, 상향식 처리(ψ_{bot})는 상대적으로 적은 개수의 함수 집합을 포함하는 분석식에서부터 점점 더 많은 개수의 프로그램 구성요소의 집합을 포함하는 분석식으로 일반화하도록 구현된 알고리즘을 포함할 수 있다.

[70] 일 실시예에 의하면, 상향식 처리(ψ_{bot})는 제2 학습 알고리즘(30-2)을 기반으로 수행되되, 예를 들어, 도 6에 도시된 바와 같이 SPEC2GEN(bot, top, <f_0, f_1, …, f_k>, F, P, A, f_{rem}) 함수를 이용하여 수행될 수 있다. 일 실시예에 따른 SPEC2GEN 함수에서는, 먼저 개별 특성(a_j)의 집합(A) 및 이들의 부정 값($\neg a_j$)을 포함하는 대상 집합이 설정되고(제2 줄), 분석식 f가 초기 분석식 함수(일례로 InitialFormula (A, P))에 의해 정의된다(제3 줄).

[71] 여기서, InitialFormula (A, P))는 하기의 수학식 11과 같이 주어질 수 있다.

[72] [수식11]

$$\{c \in C \mid C = ClausesThatProveQueries(P,A), \forall c' \in C.\ c' \subseteq c \Rightarrow c' = c\}$$

[73] 수학식 11의 ClausesThatProveQueries(P,A)는 수학식 12로 정의될 수 있다.

[74] [수식12]

$$ClausesThatProveQueries(P, A) =$$

[75] $$\{AtomsThatProveQuery(P, q, A) \mid$$

[76] $$p \in P, q \in Q_p, q \notin proved(F_p(H_{\Pi_{false}}(p)))\}$$

[77] 수학식 12의 함수 AtomsThatProveQuery(P,q,A)는 수학식 13과 같이 규정 가능하다.

[78] [수식13]

$$AtomsThatProveQuery(P,q,A) = \{a \in A, q \in proved(F_p(H_{\Pi_a}(p)))\}$$

[79] 상술한 수학식 11 내지 13에 따라 생성된 초기 분석식 f는 가장 작은 가능한 일군의 구절에 의한 쿼리를 나타낸다.

[80] 순차적으로 작업 리스트(W)에 초기 분석식 f가 입력되고(제4 줄), 작업 리스트가 공집합이 아니라면(제5 줄), 적어도 하나의 구절 c가 선택되고(제6 줄), 작업 리스트(W)에서 구절 c가 제거될 수 있다(제7 줄). 적어도 하나의 구절 c는 수학식 14와 같은 ChooseClause(W) 함수를 이용하여 수행될 수 있다.

[81] [수식14]

$$ChooseClause(W)= \underset{c \in W}{argmin} \; | \sum_{p \in P} [[c]]_p |$$

[82] 구절 c가 획득되면, 제2 학습부(144-2)는 반복적으로 구절에 대한 일반화를 수행한다(제8 내지 20줄). 구체적으로 특질 a가 수학식 10의 ChooseAtom 함수 등을 이용하여 획득되고(제9 줄), 만약 획득된 특질 a가 false 값이라면, 반복은 종료되고(제10 및 11 줄), 반대로 특징 a가 false 값이 아니라면 일반화된 분석식 f_{top}와 일반화된 집합(Π)가 각각 획득된다(제13 및 14줄). 분석식 f_{top}는 분석식 f에서 구절 c를 뺀 차집합과 구절 c에서 특질의 집합 {a}를 차감한 차집합의 합집합으로 정의되고, 일반화된 집합(Π)는 각 함수 f_0 내지 f_k 간의 연결을 분리하여 결정될 수 있다.

[83] 이어서 일반화된 분석식 f_{top}가 정교한지 여부가 검토되되, 이를 위해 상술한

$$precise(\Pi_j, F, P)$$

함수가 이용될 수 있다. 만약

$$precise(\Pi_j, F, P)$$

함수의 연산 결과 일반화된 분석식 f_{top}가 정교한 것으로 판단되면, 분석식 f_{top}는 리턴 된다(제15 및 16줄). 반대로 분석식 f_{top}가 정교한 것으로 판단되지 않으면(제17 줄), f에 분석식 f_{top}가 입력되고(제18 줄), 상술한 과정이 반복된다(제20 줄). 즉, 구절 c를 더 일반화 한다. 만약 구절 c에 더이상 차감할 특질이 없는 경우(제 10줄), 다른 구절을 일반화 한다(제 6줄).

[84] 한편 상술한 과정은 작업 리스트(W)가 공집합이 될 때까지 반복될 수 있다(제5 및 21줄).

[85] 도 7은 분석부의 동작의 일례를 설명하기 위한 도면이다.

[86] 상술한 바와 같이 학습된 분석식(80), 일례로 분석식 f_0, f_1 및 f_2가 도 7에 도시된 바와 같이 획득될 수 있다. 여기서 학습된 분석식(80)은 제1 학습부(144-1)가 획득한 적어도 하나의 제1 학습된 분석식 및 제2 학습부(144-2)가 획득한 적어도 하나의 제2 학습된 분석식 중 적어도 하나를 포함할 수 있다. 이 경우, 분석부(140)는 학습된 분석식(80)을 기반으로 분석 대상 프로그램(20)의 각각의 함수 또는 둘 이상의 함수의 조합 등(20:20-1 내지 20-4, 일례로 메소드 1 내지 4)이 어느 정도의 분석 깊이로 분석되어야 하는지를 결정하고 분석 결과(90)를 출력할 수 있다. 구체적으로 예를 들어, 학습 결과에 따라서 분석식 f_0(즉, 가장 얕은 깊이로 분석될 함수에 대한 집합)은 $(\neg a_1 \wedge a_2)$로 주어지고, 분석식 f_1(즉, 중간 깊이로 분석될 함수에 대한 집합)은 $(a_1 \wedge \neg a_2)$로 주어지며, 분석식 f_2(즉,

가장 깊은 깊이로 분석될 함수의 집합)은 $(a_1 \wedge a_2) \vee (\neg a_1 \wedge \neg a_2)$으로 주어진 상황에서 분석 대상 프로그램(20)의 메소드 1(20-1)은 특질 a_1 및 a_2를 포함하고($\{a_1, a_2\}$), 메소드 2(20-2)는 특질 a_1을 포함하며($\{a_1\}$), 메소드 3(20-3)은 특질 a_2를 포함하고($\{a_2\}$), 메소드 4(20-4)는 어떠한 특질도 포함하고 있지 않다면, 분석부(140)를 주어진 분석식 f_0 내지 f_2를 이용하여 메소드 1 및 4(20-1, 20-4)는 매우 정확하게 분석하고, 메소드 2(20-2)는 정확하되 메소드 1 및 4(20-1, 20-4)에 비해선 상대적으로 덜 정확하게 분석하며, 메소드 3(20-3)은 대략적으로 분석하도록 결정 및 분류할 수 있다. 이에 따라 특정 함수가 호출될 때, 해당 함수에 부합되는 적절한 깊이로 문맥을 구별 및 선별하여 분석 전략을 구축할 수 있게 되며, 포인터 분석 등에 채용된 경우, 포인터 분석 등을 보다 신속하고 정확하게 수행할 수 있게 된다.

[87] 이하 도 8을 참조하여 프로그램 분석 방법의 일 실시예를 설명한다.

[88] 도 8은 프로그램 분석 방법의 일 실시예에 대한 흐름도이다.

[89] 도 8에 도시된 프로그램 분석 방법의 일 실시예에 따르면, 학습 대상 프로그램과 관련하여 먼저 학습될 모든 또는 일부의 분석식이나 적어도 하나의 변수(일례로 상술한 top 및 bot)에 대한 초기화가 수행될 수 있다(400). 초기화는 분석식을 false 값으로 처리하거나 또는 변수에 0이나 k 값을 할당함으로써 수행될 수 있다.

[90] 초기화와 동시에, 이전에 또는 이후에 학습 처리가 수행되지 않은 적어도 하나의 분석식(일례로 f_{top} 또는 f_{bot})이 선택될 수 있다. 학습 처리가 수행되지 않은 적어도 하나의 분석식(f_{top} 또는 f_{bot})은, 예를 들어 false 값으로 초기화된 것일 수 있다.

[91] 학습 처리가 수행되지 않은 적어도 하나의 분석식에 대해 상향식 처리가 수행될지 또는 하향식 처리가 수행될지 여부가 판단 및 결정될 수 있다(404 및 408). 하향식 처리 및 상향식 처리의 선택은, 예를 들어, 먼저 주어진 함수 집합에 대해 하향식 처리가 가능한지 여부가 판단된다. 하향식 처리의 가능 여부는 상술한 수학식 7의 ChooseProblem(top, Π, F, P, f_{rem}) 함수를 이용하여 판단될 수 있다. 만약 수학식 7의 ChooseProblem(top, Π, F, P, f_{rem}) 함수가 true 값을 출력하면 하향식 처리가 가능한 것으로 판단되고(404의 예), 반대로 ChooseProblem(top, Π, F, P, f_{rem})가 false 값을 출력하면 상향식 처리가 가능한 것으로 판단될 수 있다(404의 아니오).

[92] 만약 하향식으로 적어도 하나의 분석식에 대한 수행을 하는 경우라면(404의 예), 제1 학습 알고리즘, 일례로 GEN2SPEC 함수를 기반으로 적어도 하나의 분석식에 대한 학습이 수행된다(406). 즉, 상향식 처리는 제1 학습부를 통해 설명한 바와 같이 수행될 수 있다. 이에 따라 제1 학습된 분석식이 획득될 수 있다.

[93] 반대로 하향식 처리를 수행하지 않는 경우라면(404의 아니오), 이에 응하여 상향식 처리가 수행되되(408), 제2 학습 알고리즘, 일례로 SPEC2GEN 함수를

기반으로 적어도 하나의 분석식에 대한 처리가 수행될 수 있다. 보다 상세하게 예를 들어, 상향식 처리 과정에 있어서 제2 학습 알고리즘에 의해 분석식 f'_{top}이 획득되고, f'_{top}의 부정($\neg f'_{top}$) 및 f_{rem} 값의 교집합($f_{rem} \wedge \neg f'_{top}$)이 변수 f_{bot}에 입력된다. 제2 학습 알고리즘에 의한 처리는 제2 학습부를 통해 설명한 바와 동일하게 수행 가능하다. 이에 따라 제2 학습된 분석식이 획득될 수 있다.

[94] 선택적으로 상향식 처리 또는 하향식 처리를 수행하는 과정(404 내지 408)은 모든 또는 기 설정된 일부의 분석식(들)에 대한 처리 방법(즉, 하향식 처리 또는 상향식 처리)의 선택과 이에 따른 처리(즉, 하향식 처리 또는 상향식 처리)가 종료될 때까지 반복 수행될 수 있다(412). 다시 말해서, 모든 또는 기 설정된 일부의 분석식(들)에 대해서 하향식 처리 또는 상향식 처리가 결정되면, 분석식에 대한 상향식 또는 하향식 처리 방법의 선택 과정과 이에 따른 상향식 또는 하향식 처리 과정은 종료될 수 있다.

[95] 상향식 처리 또는 하향식 처리의 결정(404, 408) 및 이에 따른 제1 학습 알고리즘의 적용(406) 또는 제2 학습 알고리즘의 적용 과정(408)의 반복 결과에 따라서, 적어도 하나의 학습된 분석식이 결정된다(412, 414).

[96] 분석식 결정 도중, 분석식 결정 전 및/또는 분석식 결정 후에 분석 대상 프로그램이 입력될 수 있다(416). 분석 대상 프로그램은 학습 대상 프로그램일 수도 있다. 분석 대상 프로그램에 대해 상술한 분석식을 기반으로 분석이 수행되며, 이에 따라 분석 대상 프로그램 내의 각각의 함수에 대해 분석 깊이가 결정될 수 있다(418). 분석 결과는 필요에 따라 외부로 출력될 수도 있다.

[97] 상술한 실시예에 따른 프로그램 분석 방법은, 컴퓨터 장치에 의해 구동될 수 있는 프로그램의 형태로 구현될 수 있다. 여기서 프로그램은, 프로그램 명령, 데이터 파일 및 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 프로그램은 기계어 코드나 고급 언어 코드를 이용하여 설계 및 제작된 것일 수 있다. 프로그램은 상술한 방법을 구현하기 위하여 특별히 설계된 것일 수도 있고, 컴퓨터 소프트웨어 분야에서 통상의 기술자에게 기 공지되어 사용 가능한 각종 함수나 정의를 이용하여 구현된 것일 수도 있다. 또한, 여기서, 컴퓨터 장치는, 프로그램의 기능을 실현 가능하게 하는 프로세서나 메모리 등을 포함하여 구현된 것일 수 있으며, 필요에 따라 통신 장치를 더 포함할 수도 있다. 또한, 상술한 프로그램 분석 방법을 구현하기 위한 프로그램은, 컴퓨터에 의해 판독 가능한 기록 매체에 기록될 수 있다. 컴퓨터에 의해 판독 가능한 기록 매체는, 예를 들어, 솔리드 스테이트 드라이브(SSD), 롬, 램 또는 플래시 메모리 등과 같은 반도체 저장 장치, 하드 디스크나 플로피 디스크 등과 같은 자기 디스크 저장 매체, 콤팩트 디스크나 디브이디 등과 같은 광 기록 매체, 플롭티컬 디스크 등과 같은 자기-광 기록 매체 및 자기 테이프 등 컴퓨터 등의 호출에 따라 실행되는 특정 프로그램을 저장 가능한 적어도 한 종류의 물리적 장치를 포함할 수 있다.

[98] 이상 프로그램 분석 장치 및 방법을 적어도 하나의 실시예를 들어

설명하였으나, 프로그램 분석 장치 및 방법은 오직 상술한 실시예에 한정되는 것은 아니다. 해당 기술 분야에서 통상의 지식을 가진 자가 상술한 실시예를 기초로 수정 및 변형하여 구현 가능한 다양한 장치나 방법 역시 상술한 프로그램 분석 장치 및 방법의 일례가 될 수 있다. 예를 들어, 설명된 기술들이 설명된 방법과 상이한 순서로 수행되거나 및/또는 설명된 구조, 기능, 회로, 장치 등의 구성 요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나 다른 구성 요소 또는 균등물에 의하여 대치되거나 또는 치환되더라도 상술한 프로그램 분석 장치 및 방법의 일 실시예가 될 수 있다.

청구범위

[청구항 1] 복수의 분석식 중 적어도 하나의 분석식에 대해서 하향식 처리 및 상향식 처리 중 어느 하나를 선택하는 선택부;
상기 선택부가 상기 하향식 처리를 선택하는 경우, 이에 응하여 제1 학습 알고리즘을 기반으로 상기 적어도 하나의 분석식에 대한 하향식 처리를 수행하여 적어도 하나의 제1 학습된 분석식을 획득하는 제1 학습부; 및
상기 선택부가 상기 상향식 처리를 선택하는 경우, 이에 응하여 제2 학습 알고리즘을 기반으로 상기 적어도 하나의 분석식에 대한 하향식 처리를 수행하여 적어도 하나의 제2 학습된 분석식을 획득하는 제2 학습부;를 포함하는 프로그램 분석 장치.

[청구항 2] 제1항에 있어서,
상기 제2 학습부는, 적어도 하나의 특성의 집합 및 이들의 부정 값을 포함하는 대상 집합을 설정하고, 초기 분석식 함수를 이용하고 초기 분석식을 정의하고, 적어도 하나의 구절을 선택하여 획득하고 상기 적어도 하나의 구절을 일반화하고, 일반화된 구절 및 상기 초기 분석식을 이용하여 적어도 하나의 일반화된 분석식을 획득하고 획득된 적어도 하나의 일반화된 분석식에 대한 정교함 여부를 검토하여 상기 상향식 처리를 수행하는 프로그램 분석 장치.

[청구항 3] 제2항에 있어서,
상기 적어도 하나의 일반화된 분석식이 정교한 것으로 판단되지 않으면, 상기 적어도 하나의 일반화된 분석식에 대하여 보다 일반화된 분석식을 획득하는 프로그램 분석 장치.

[청구항 4] 제1항에 있어서,
상기 선택부는, 분석식에 대한 처리가 주어진 비용 임계 값 범위 내에서 종료 가능하면 하향식 처리를 선택하고, 분석식에 대한 처리가 주어진 비용 임계 값 범위 내에서 종료 불가능하면 상향식 처리를 선택하는 프로그램 분석 장치.

[청구항 5] 제1항에 있어서,
상기 선택부는, 상기 복수의 분석식 모두에 대해 하향식 처리 또는 상향식 처리가 선택될 때까지 상기 적어도 하나의 분석식에 대해서 하향식 처리 및 상향식 처리 중 어느 하나를 선택하는 프로그램 분석 장치.

[청구항 6] 제1항에 있어서,
상기 제1 학습부는, 적어도 하나의 특성의 집합 및 이들의 부정 값을 포함하는 대상 집합을 설정하고, 상기 대상 집합의 개별 특성의 논리합을 포함하도록 하여 초기 분석식을 생성하고, 구절 및 특성을 결정한 후 결정된 구절 및 특성을 조합하여 정제된 구절을 획득하고, 정제된 구절 및 초기 분석식을 기반으로 새로운 분석식을 획득하고, 상기 새로운

분석식에 대한 정교함 여부를 검토하여 상기 하향식 처리를 수행하는 프로그램 분석 장치.

[청구항 7] 제1항에 있어서,
상기 제1 학습된 분석식 및 제2 학습된 분석식 중 적어도 하나를 이용하여 분석 대상 프로그램의 적어도 하나의 함수에 대한 분석 깊이를 결정하는 분석부;를 더 포함하는 프로그램 분석 장치.

[청구항 8] 복수의 분석식 중 적어도 하나의 분석식에 대해서 하향식 처리 및 상향식 처리 중 어느 하나를 선택하는 단계; 및
상기 선택부가 상기 상향식 처리를 선택하는 경우, 이에 응하여 제2 학습 알고리즘을 기반으로 상기 적어도 하나의 분석식에 대한 하향식 처리를 수행하여 적어도 하나의 제2 학습된 분석식을 획득하는 단계;를 포함하는 프로그램 분석 방법.

[청구항 9] 제8항에 있어서,
상기 제2 학습 알고리즘을 기반으로 상기 적어도 하나의 분석식에 대한 하향식 처리를 수행하여 적어도 하나의 제2 학습된 분석식을 획득하는 단계는,
적어도 하나의 특성의 집합 및 이들의 부정 값을 포함하는 대상 집합을 설정하는 단계;
초기 분석식 함수를 이용하고 초기 분석식을 정의하는 단계;
적어도 하나의 구절을 선택하여 획득하고 상기 적어도 하나의 구절을 일반화하는 단계;
일반화된 구절 및 상기 초기 분석식을 이용하여 적어도 하나의 일반화된 분석식을 획득하는 단계; 및
획득된 적어도 하나의 일반화된 분석식에 대한 정교함 여부를 검토하여 상기 상향식 처리를 수행하여 상기 제2 학습된 분석식을 획득하는 단계;를 포함하는 프로그램 분석 방법.

[청구항 10] 제9항에 있어서,
상기 제2 학습 알고리즘을 기반으로 상기 적어도 하나의 분석식에 대한 하향식 처리를 수행하여 적어도 하나의 제2 학습된 분석식을 획득하는 단계는,
상기 적어도 하나의 일반화된 분석식이 정교한 것으로 판단되지 않으면, 상기 적어도 하나의 일반화된 분석식에 대하여 보다 일반화된 분석식을 획득하는 단계;를 더 포함하는 프로그램 분석 방법.

[청구항 11] 제8항에 있어서,
상기 복수의 분석식 중 적어도 하나의 분석식에 대해서 하향식 처리 및 상향식 처리 중 어느 하나를 선택하는 단계는,
분석식에 대한 처리가 주어진 비용 임계 값 범위 내에서 종료 가능하면 하향식 처리를 선택하는 단계; 및

분석식에 대한 처리가 주어진 비용 임계 값 범위 내에서 종료 불가능하면 상향식 처리를 선택하는 단계;를 포함하는 프로그램 분석 방법.

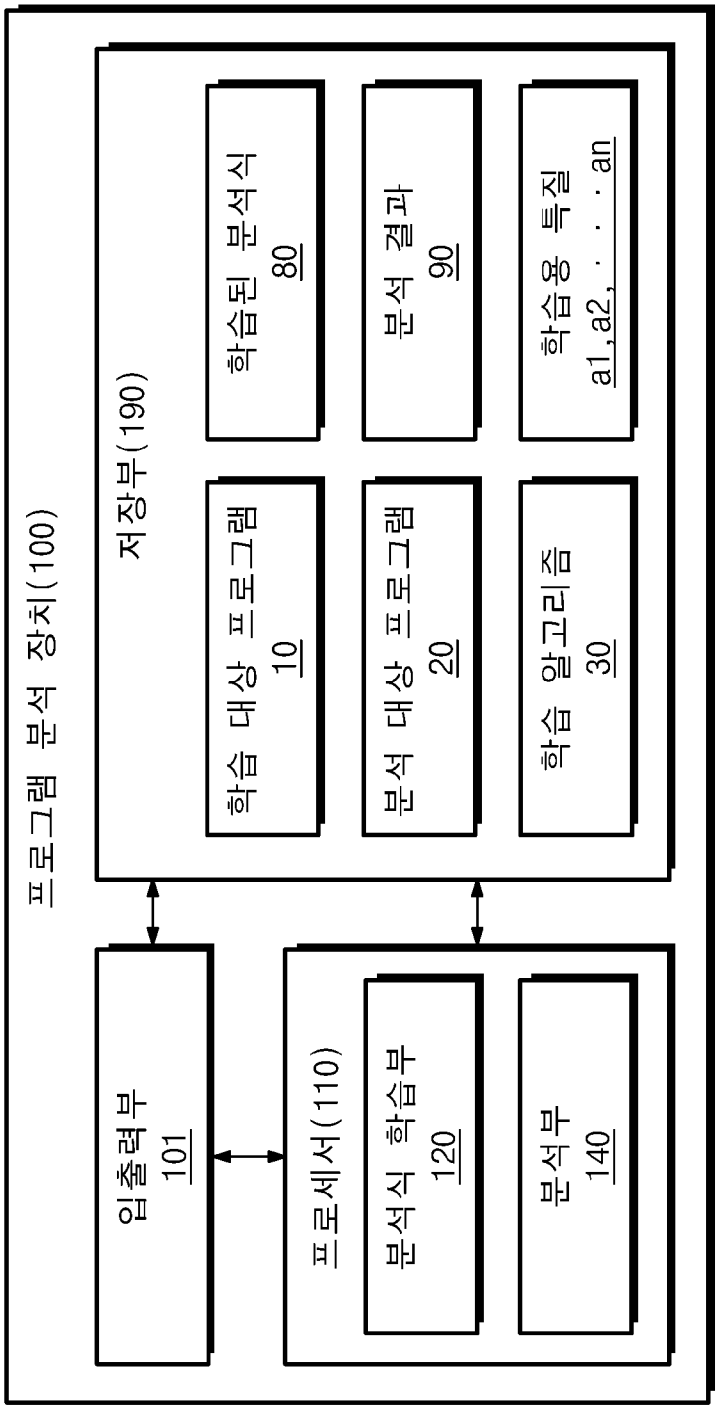
[청구항 12] 제8항에 있어서,
상기 복수의 분석식 모두에 대해 하향식 처리 또는 상향식 처리가 선택되면, 상기 적어도 하나의 분석식에 대해서 하향식 처리 및 상향식 처리 중 어느 하나에 대한 선택을 종료하는 단계;를 더 포함하는 프로그램 분석 방법.

[청구항 13] 제8항에 있어서,
상기 선택부가 하향식 처리를 선택하는 경우, 이에 응하여 제1 학습 알고리즘을 기반으로 상기 적어도 하나의 분석식에 대한 하향식 처리를 수행하여 적어도 하나의 제1 학습된 분석식을 획득하는 단계;를 더 포함하는 프로그램 분석 방법.

[청구항 14] 제13항에 있어서,
상기 제1 학습 알고리즘을 기반으로 상기 적어도 하나의 분석식에 대한 하향식 처리를 수행하여 적어도 하나의 제1 학습된 분석식을 획득하는 단계는,
적어도 하나의 특성의 집합 및 이들의 부정 값을 포함하는 대상 집합을 설정하는 단계;
상기 대상 집합의 개별 특성의 논리합을 포함하도록 하여 초기 분석식을 생성하는 단계;
구절 및 특성을 결정한 후 결정된 구절 및 특성을 조합하여 정제된 구절을 획득하는 단계;
상기 정제된 구절 및 상기 초기 분석식을 기반으로 새로운 분석식을 획득하는 단계;
상기 새로운 분석식에 대한 정교함 여부를 검토하는 단계;를 포함하는 프로그램 분석 방법.

[청구항 15] 제1항에 있어서,
상기 제2 학습된 분석식을 이용하여 분석 대상 프로그램의 적어도 하나의 함수에 대한 분석 깊이를 결정하는 단계;를 더 포함하는 프로그램 분석 방법.

[도1]



[도2]

$$\langle f_0 = false, f_1 = false, f_2 = false \rangle$$

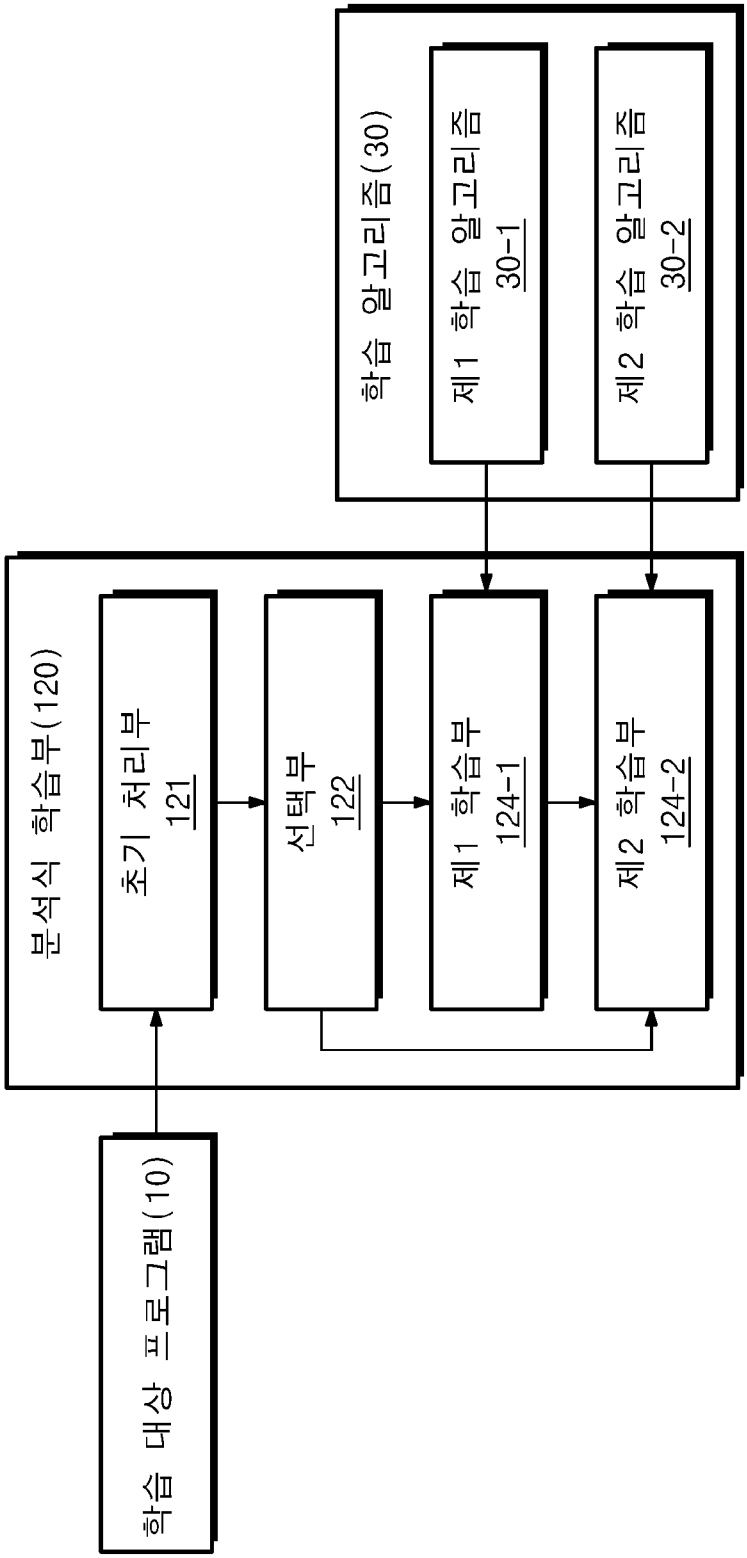
$$\downarrow f_0 \text{ 학습}$$

$$\langle f_0 = (a_1 \wedge a_2 \wedge \ldots) \vee \ldots, f_1 = false, f_2 = false \rangle$$

$$\downarrow f_2 \text{ 학습}$$

$$\langle f_0 = (a_1 \wedge a_2 \wedge \ldots) \vee \ldots, f_1 = \neg f_2 \wedge \neg f_0, f_2 = (\neg a_1 \wedge a_3 \wedge \ldots) \rangle$$

[도3]


학습 대상 프로그램(10)
분석식 학습부(120)
초기 처리부 121
선택부 122
제1 학습부 124-1
제2 학습부 124-2
학습 알고리즘(30)
제1 학습 알고리즘 30-1
제2 학습 알고리즘 30-2

[도4]

```
1: procedure LEARN( F, P, k, 𝔸 )
2:     ⟨f_0, f_1, ..., f_k⟩ ← ⟨false, false, ..., false⟩
3:     bot ← 0
4:     top ← k
5:     while top ≠ bot do
6:         f_rem ← ¬(⋁_{n=0}^{k} f_n)
7:         b ← ChooseProblem(top, ⟨f_0, f_1, ..., f_k⟩, F, P, f_rem
8:         if(b=true) then
9:             f_top ← GEN2SPEC(top−1, top, ⟨f_0, f_1, ... f_k⟩, F, P, 𝔸, f_rem)
10:            top ← top−1,
11:        else
12:            f'_top ← SPEC2GEN(bot, top, ⟨f_0, f_1, ... f_k⟩, F, P, 𝔸, f_rem)
13:            f_bot ← f_rem ∧ ¬f'_top
14:            bot ← bot + 1
15:        end if
16:    end while
17:    f_top ← ¬(⋁_{n=0}^{k} f_n)
18:    return ⟨f_0, f_1, ... f_k⟩
19: end procedure
```

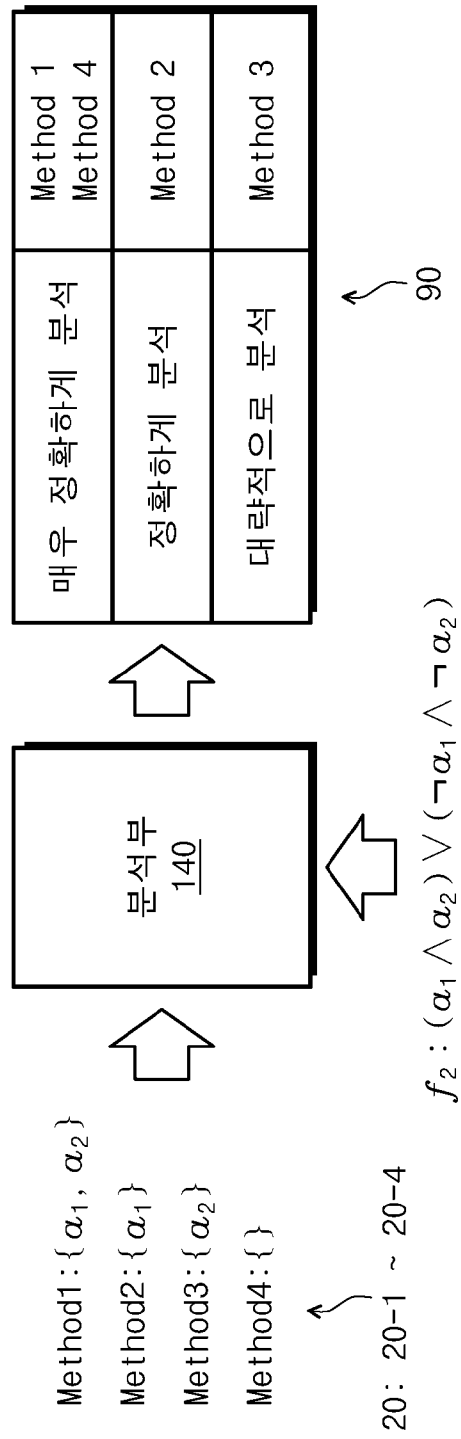
[도5]

```
1: procedure GEN2SPEC(bot, top, ⟨f_0, f_1, ... f_k⟩, F, P, 𝔸, f_rem)
2:     𝔸 ← 𝔸 ∪ {¬a_j | a_j ∈ 𝔸}
3:     f ← {{a_i} | a_i ∈ 𝔸}
4:     W ← f
5:     while W ≠ ∅ do
6:         c ← MostExpensiveClause (W, F, P,)
7:         W ← W \ {c}
8:         a ← ChooseAtom(𝔸, c)
9:         c' ← c ∪ {a}
10:        f_top ← (f \ {c}) ∪ {c'}
11:        Π ← MakeDisjoint( bot, top, f_rem, ⟨f_0, f_1, ... f_k⟩)
12:        if Precise( Π, F , P) then
13:        f ← f_top
14:        W ← W ∪ {c'}
15:        end if
16:    end while
17:    return f ∧ f_rem
18: end procedure
```

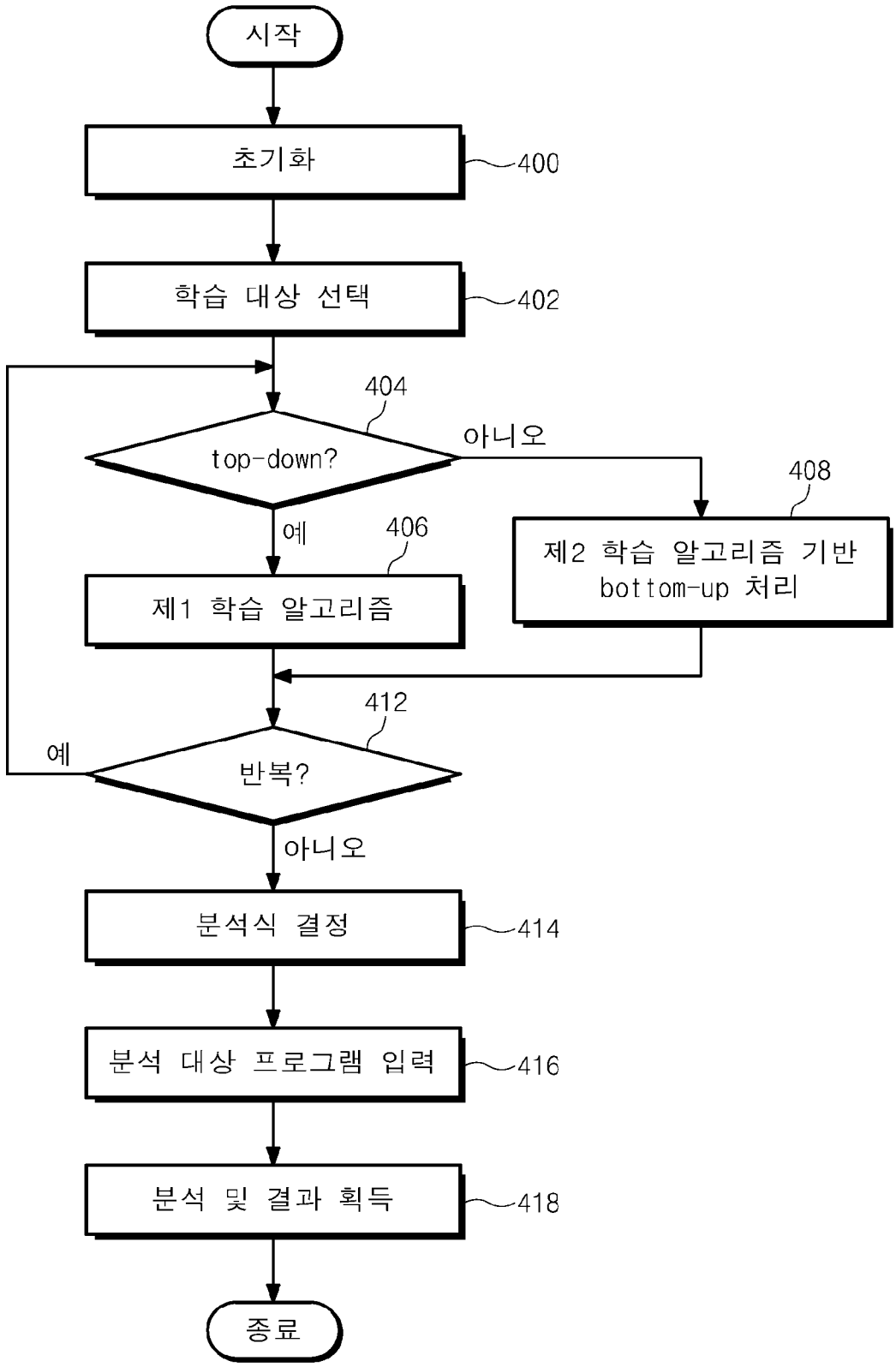
[도6]

```
1: procedure SPEC2GEN(bot, top, ⟨f_0, f_1, ... f_k⟩, F, P, 𝔸, f_rem)
2:     𝔸 ← 𝔸 ∪ {¬a_j | a_j ∈ 𝔸}
3:     f ← InitialFormula(𝔸, P)
4:     W ← f
5:     while W ≠ ∅ do
6:         c ← ChooseClause(W)
7:         W ← W \ {c}
8:         while true do
9:             a ← ChooseAtom(c)
10:            if a = false then
11:                break
12:            end if
13:            f_top ← (f \ {c}) ∪ {c \ {a}}
14:            Π ← MakeDisjoint(bot, top, ⟨f_0, f_1, ... f_k⟩)
15:            if Precise (Π, F, P) then
16:                return f_top
17:            else
18:                f ← f_top
19:            end if
20:            end while
21:        end while
22:        return true
23: end procedure
```

[도7]


Method1:{α_1, α_2}
Method2:{α_1}
Method3:{α_2}
Method4:{}
20: 20-1 ~ 20-4
분석부
140
$f_2 : (\alpha_1 \wedge \alpha_2) \vee (\neg\alpha_1 \wedge \neg\alpha_2)$
$f_1 : (\alpha_1 \wedge \neg\alpha_2)$
$f_0 : (\neg\alpha_1 \wedge \alpha_2)$
80
매우 정확하게 분석
정확하게 분석
대략적으로 분석
Method 1
Method 4
Method 2
Method 3
90

[도8]


시작
초기화
400
학습 대상 선택
402
404
top-down?
아니오
408
예
406
제1 학습 알고리즘
제2 학습 알고리즘 기반
bottom-up 처리
412
예
반복?
아니오
분석식 결정
414
분석 대상 프로그램 입력
416
분석 및 결과 획득
418
종료

INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR2022/002759

A. CLASSIFICATION OF SUBJECT MATTER

G06F 11/36(2006.01)i; **G06N 20/20**(2019.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 11/36(2006.01); G06F 1/00(2006.01); G06F 11/28(2006.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models: IPC as above
Japanese utility models and applications for utility models: IPC as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS (KIPO internal) & keywords: 프로그램 테스트(program test), 상향식(bottom-up), 하향식(top-down), 선택(select), 함수(function)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2015-001825 A (FUJI ELECTRIC CO., LTD.) 05 January 2015 (2015-01-05) See claims 1-10.	1-15
A	KR 10-2114547 B1 (KOREA ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY) 25 May 2020 (2020-05-25) See paragraphs [0042]-[0052]; and figure 1.	1-15
A	KR 10-2018-0119260 A (CODEMIND CORPORATION et al.) 02 November 2018 (2018-11-02) See paragraphs [0054]-[0063]; and figure 3.	1-15
A	WO 98-06019 A2 (INTRINSA CORPORATION) 12 February 1998 (1998-02-12) See claims 1-22.	1-15
A	US 2020-0285563 A1 (FUJITUS LIMITED) 10 September 2020 (2020-09-10) See paragraphs [0065]-[0081]; and figures 3A-3B.	1-15

☑ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 May 2022	**30 May 2022**

Name and mailing address of the ISA/KR	Authorized officer
Korean Intellectual Property Office **Government Complex-Daejeon Building 4, 189 Cheongsa-ro, Seo-gu, Daejeon 35208**	
Facsimile No. **+82-42-481-8578**	Telephone No.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 1605968 A (NANJING UNIVERSITY) 13 April 2005 (2005-04-13) See claims 1-4.	1-15

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/KR2022/002759

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
JP 2015-001825 A	05 January 2015	JP 6107455 B2	05 April 2017
KR 10-2114547 B1	25 May 2020	KR 10-2019-0128457 A	18 November 2019
KR 10-2018-0119260 A	02 November 2018	None	
WO 98-06019 A2	12 February 1998	JP 2002-515996 A	28 May 2002
		JP 4633203 B2	16 February 2011
		WO 98-06019 A3	26 March 1998
US 2020-0285563 A1	10 September 2020	JP 2020-144842 A	10 September 2020
		US 11237943 B2	01 February 2022
CN 1605968 A	13 April 2005	CN 100465853 C	04 March 2009

국 제 조 사 보 고 서

국제출원번호
PCT/KR2022/002759

A. 발명이 속하는 기술분류(국제특허분류(IPC))

G06F 11/36(2006.01)i; **G06N 20/20**(2019.01)i

B. 조사된 분야

조사된 최소문헌(국제특허분류를 기재)

G06F 11/36(2006.01); G06F 1/00(2006.01); G06F 11/28(2006.01)

조사된 기술분야에 속하는 최소문헌 이외의 문헌

한국등록실용신안공보 및 한국공개실용신안공보: 조사된 최소문헌란에 기재된 IPC
일본등록실용신안공보 및 일본공개실용신안공보: 조사된 최소문헌란에 기재된 IPC

국제조사에 이용된 전산 데이터베이스(데이터베이스의 명칭 및 검색어(해당하는 경우))

eKOMPASS(특허청 내부 검색시스템) & 키워드: 프로그램 테스트(program test), 상향식(bottom-up), 하향식(top-down), 선택(select), 함수(function)

C. 관련 문헌

카테고리*	인용문헌명 및 관련 구절(해당하는 경우)의 기재	관련 청구항
A	JP 2015-001825 A (FUJI ELECTRIC CO., LTD.) 2015.01.05 청구항 1-10	1-15
A	KR 10-2114547 B1 (한국과학기술원) 2020.05.25 단락 [0042]-[0052]; 및 도면 1	1-15
A	KR 10-2018-0119260 A (코드마인드(주) 등) 2018.11.02 단락 [0054]-[0063]; 및 도면 3	1-15
A	WO 98-06019 A2 (INTRINSA CORPORATION) 1998.02.12 청구항 1-22	1-15
A	US 2020-0285563 A1 (FUJITUS LIMITED) 2020.09.10 단락 [0065]-[0081]; 및 도면 3A-3B	1-15
A	CN 1605968 A (NANJING UNIVERSITY) 2005.04.13 청구항 1-4	1-15

☐ 추가 문헌이 C(계속)에 기재되어 있습니다.

☑ 대응특허에 관한 별지를 참조하십시오.

* 인용된 문헌의 특별 카테고리:

"A" 특별히 관련이 없는 것으로 보이는 일반적인 기술수준을 정의한 문헌

"D" 본 국제출원에서 출원인이 인용한 문헌

"E" 국제출원일보다 빠른 출원일 또는 우선일을 가지나 국제출원일 이후에 공개된 선출원 또는 특허 문헌

"L" 우선권 주장에 의문을 제기하는 문헌 또는 다른 인용문헌의 공개일 또는 다른 특별한 이유(이유를 명시)를 밝히기 위하여 인용된 문헌

"O" 구두 개시, 사용, 전시 또는 기타 수단을 언급하고 있는 문헌

"P" 우선일 이후에 공개되었으나 국제출원일 이전에 공개된 문헌

"T" 국제출원일 또는 우선일 후에 공개된 문헌으로, 출원과 상충하지 않으며 발명의 기초가 되는 원리나 이론을 이해하기 위해 인용된 문헌

"X" 특별한 관련이 있는 문헌. 해당 문헌 하나만으로 청구된 발명의 신규성 또는 진보성이 없는 것으로 본다.

"Y" 특별한 관련이 있는 문헌. 해당 문헌이 하나 이상의 다른 문헌과 조합하는 경우로 그 조합이 당업자에게 자명한 경우 청구된 발명은 진보성이 없는 것으로 본다.

"&" 동일한 대응특허문헌에 속하는 문헌

국제조사의 실제 완료일	국제조사보고서 발송일
2022년05월30일(30.05.2022)	**2022년05월30일(30.05.2022)**

ISA/KR의 명칭 및 우편주소	심사관
대한민국 특허청 **(35208) 대전광역시 서구 청사로 189, 4동 (둔산동, 정부대전청사)**	**양정록**
팩스 번호 **+82-42-481-8578**	전화번호 **+82-42-481-5709**

서식 PCT/ISA/210 (두 번째 용지) (2019년 7월)

국 제 조 사 보 고 서
대응특허에 관한 정보

국제출원번호
PCT/KR2022/002759

국제조사보고서에서 인용된 특허문헌	공개일	대응특허문헌	공개일
JP 2015-001825 A	2015/01/05	JP 6107455 B2	2017/04/05
KR 10-2114547 B1	2020/05/25	KR 10-2019-0128457 A	2019/11/18
KR 10-2018-0119260 A	2018/11/02	없음	
WO 98-06019 A2	1998/02/12	JP 2002-515996 A	2002/05/28
		JP 4633203 B2	2011/02/16
		WO 98-06019 A3	1998/03/26
US 2020-0285563 A1	2020/09/10	JP 2020-144842 A	2020/09/10
		US 11237943 B2	2022/02/01
CN 1605968 A	2005/04/13	CN 100465853 C	2009/03/04