논문 2025-2-5 http://dx.doi.org/10.29056/jsav.2025.06.05

소프트웨어 유사도 평가를 위한 분석 도구

이창훈*†

Analysis Tools for Software Similarity Evaluation

Chang-Hoon Lee*†

요 약

현대 소프트웨어 산업은 빠른 개발 주기와 치열한 경쟁 속에서 방대한 양의 소스코드가 재사용되고 공유되는 구조로 변화하고 있다. 이러한 환경에서 일부 기업은 경쟁사의 소스코드를 무단으로 복제하거나 오픈소스 라이선스를 위반하는 방식으로 제품 개발에 활용함으로써, 지적 재산권 침해 및 법적 분쟁의 원인을 초래하고 있다. 특히 최근에는 코드 난독화, 구조 변경, 함수 분할 등 다양한 표절 회피 기법을 이용하여 기존의 단순한 문자열 비교 방식으로는 표절 여부를 정확히 판별하기 어려운 상황이다. 이에 따라, 프로그램의 문법적 구조와 의미적 유사성까지 분석할 수 있는 고도화된 탐지 기법의 필요성이 대두되고 있다.

소프트웨어 표절 탐지에서 규모가 작은 소프트웨어는 수작업으로 표절을 판단할 수 있지만, 규모가 큰 소프트웨어의 경우 수작업으로 표절 여부를 판단하는 것은 불가능하다 볼 수 있다. 따라서 표절 판단을 위한 표절 유사도 분석 도구가 필요하다. 본 논문에서는 소프트웨어 특성에 따라 문서(텍스트), 데이터베이스, 기계어로 분류하고 각 특성에 맞는 유사도를 분석하고자 한다.

Abstract

In today's software industry, rapid development cycles and intense competition have led to the widespread reuse and sharing of source code. Within this context, some companies have engaged in unauthorized replication of competitors' code or violated open-source licenses to accelerate product development, leading to serious intellectual property infringements and legal disputes.

As plagiarism techniques become increasingly sophisticated through methods such as code obfuscation, structural reordering, and function fragmentation traditional string-based detection methods are no longer sufficient to accurately identify illicit reuse. In the field of software plagiarism detection, manual judgment may be feasible for small-scale software; however, for large-scale software systems, it is practically impossible to determine plagiarism manually. Therefore, automated similarity analysis tools are essential for accurate and efficient plagiarism detection. This paper classifies software into three categories—document-based (text), database-driven, and machine code—based on their characteristics, and investigates appropriate similarity analysis tools tailored to each type.

한글키워드 : 소프트웨어 분석, 소프트웨어 표절, 소프트웨어 감정, 유사도, 감정 도구

keywords : Software Analysis, Software Plagiarism, Software Appraisal, Similarity, Similarity Tool

* 한경국립대학교 컴퓨터응용수학부
† 교신저자: 이창훈(email: be4u@hknu.ac.kr)
접수일자: 2025.05.19. 심사완료: 2025.06.01.
게재확정: 2025.06.20.

1. 서 론

최근 사회가 복잡하고 빠르게 정보화됨에 따

라, 소프트웨어 불법 복제의 지적 재산권 침해 분쟁의 빈도가 증가하고 있다[1].

소프트웨어 저작권 분쟁은 저작권을 침해당했을 때 발생하는 분쟁을 말하며, 주로 소프트웨어의 코드, 디자인, 기능 등을 불법 복제했을 때 발생한다. 한국소프트웨어저작권협회[2]의 2024년 불법복제 SW 사용 제보 통계에 따르면, 제보·접수된 불법복제 프로그램은 전년(956건) 대비 29% 증가한 1,237건이다. 침해 사례를 SW 용도에 따라 분류했을 때 '설계 분야 SW' 128건(74%), '일반사무용 SW' 28건(16%), 이 외 '유틸리티 및 그래픽, 백신/보완 관련 SW'17건(10%) 순으로, 전년 대비'설계 분야 SW'의 비중이 늘었다. 상호 비교할 표절 소프트웨어의 분량이 방대하고 이를 육안 비교는 불가능하다 볼 수 있다. 따라서 비교를 자동화 할 수 있는 감정 도구가 필요하다 볼 수 있다[3-5].

본 논문에서는 유사도 평가를 위한 도구를 분석해 보고자 한다. 2장에서는 유사도 산출 방법에 대하여 살펴보고, 3장에서는 소프트웨어 유사도 산출 도구들에 대하여 살펴본 후 4장에서 결론을 맺는다.

2. 소프트웨어 유사도 산출 절차

프로그램 유사도 분석은 그림 1과 같은 절차를 따른다.

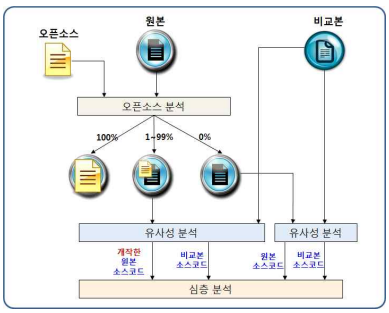


그림 1. 프로그램 유사도 분석 절차
Fig. 1. Program Similarity Analysis Procedure

첫째, 원본 프로그램에서 오픈소스 참조 여부 분석한다. 오픈소스 참조 여부는 3가지 분류로 구분할 수 있다. 오픈소스와 100% 똑같은 프로그램, 오픈소스 일부를 사용한 프로그램, 오픈소스를 사용하지 않은 프로그램으로 분류할 수 있다.

둘째, 오픈소스를 사용하지 않은 원본과 비교본간의 유사성 분석을 수행한다

셋째, 오픈소스 일부를 사용한 원본과 비교본간의 유사성 분석을 수행한다.

넷째, 그림 2와 같이 유사 라인수가 높은 원본 및 비교본에 대해 분석을 수행한다.

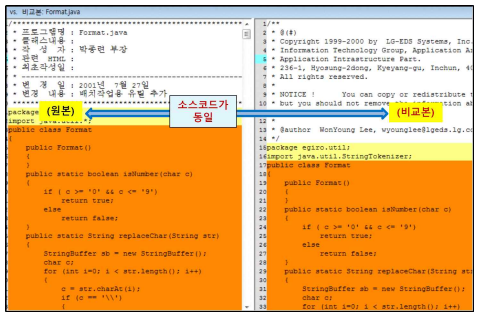


그림 2. 원본과 비교본의 유사 라인 분석
Fig. 2. Analysis of Similar Lines Between the Original and the Comparison Version

3. 소프트웨어 유사도 산출을 위한 도구

3.1 소프트웨어 표절(유사도) 산출 도구

(1) SIM

SIM은 변수 이름 변경 등에 유연하여 패턴이 유사한 부위를 잘 검출하는 장점이 있다. 문서의 어휘 유사성을 측정(lexical similarity) 함으로써 표절 여부를 검출하는 도구이다. 비교 결과는 문서 형태(토큰의 수, 이름, 사용된 전체 파일 수)로 제공된다[6].

(2) Plague

Plague 서비스는 최신 표절 탐지 알고리즘과 보고서에 대한 편리한 기능을 제공하고 있다. PDF, DOC, DOCX, RTF, TXT 등 다양한 파일 형식을 허용한다. 문서를 분석하는 동안 자동으로 선택된 텍스트 용어에 대해 여러 온라인 데이터베이스에서 동시에 검색이 가능하다[7].

(3) YAP3 (Yet Another Plague)

YAP은 Plague를 확장한 개념으로 개발되어졌다. YAP3는 구조적 매트릭스 방법을 이용한 유사도 평가 시스템이다. YAP3에서는 유사도를 평가하기 위하여 프로그램을 재설정한다. 즉, 설명문과 스트링 상수 제거하기, 대문자를 소문자로 바꾸기, 소스 프로그램을 똑같은 혹은 유사한 동의어로 변환하기, 함수 호출 순서를 재 정렬하기 등을 수행한다[8].

(4) MOSS(Measure of Software Simility)

MOSS는 코드 비교 소프트웨어이며, 소프트웨어 개발 프로젝트에 있어서 일반적으로 코드 유사성을 탐지하는 데 사용된다. MOSS는 Winnowing 알고리즘을 사용하며, 이를 지문법(fingerprinting) 기술이라 한다. MOSS는 Java, C, C++, Python, JavaScript 등 약 25개의 코딩 언어를 지원한다. 의심스러운 코드 조각, 유사성 비율, 토큰, 라인을 강조 표시해준다[9].

(5) JPlag

JPlag은 웹 서비스로서 온라인 시스템으로 서비스를 제공하고 있다. JPlag는 Greedy String Tiling 비교 알고리즘을 사용한다. MOSS 같은 비교 알고리즘을 사용하고 있으며, 실행 시간을 줄이기위해 최적화 방법을 추가였다. Java, C/C++, C#, Go, Kotlin, Python, Scala, Scheme, Swift 등 작성된 소스코드 사이의 유사성을 측정해 주는 도구이다[10].

(6) exEyes 5.1

한국저작권위원회가 자체 개발한 GUI 기반 소프트웨어 표절 검사 도구로, 소프트웨어 감정평가 프로그램으로 지원하는 언어는 Java, C, C++, VB, Text(Web) 등이다. 소프트웨어 소스코드 유사도 감정에 사용되고 있는 감정 도구이기도 하다. 코드 클론 탐지 또는 소프트웨어 유사도 측정 시에 선택할 수 있는 기능은, 동일/유사한 블록의 크기, 유사라인의 동일 토큰수 비율, 유사라인의 최소 토큰 수, 유사라인의 최대 토큰 비, 확장자 구분 등이다[11].

(7) Source Insight

Source Insight는 그림 3과 같이 코드 분석을 위해 참조 트리, 클래스 상속 다이어그램 및 함수 호출 트리도 표시할 수 있다. 함수 호출과 호출자를 빠르게 탐색. 함수, 변수 등에 대한 참조를 거의 즉시 실행하고 호출 그래프 및 클래스 트리 다이어그램을 표시한다. Source Insight는 호출 그래프를 제공하여 클래스 계층 구조, 호출 트리, 참조 트리 등을 표시할 수 있다. Source Insight는 클래스 상속 표시도 제공한다.

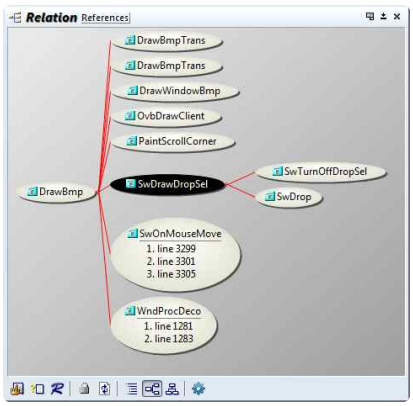


그림 3. Source Insight
Fig. 3. Source Insight

3.2 자연어(문서) 표절(유사도) 산출 도구

(1) Nifty Docs

Nifty 는 내장된 문서 관리 기능을 제공하는 올인원 프로젝트 관리 소프트웨어이다. Nifty Docs는 다른 도구를 사용하지 않고도 프로젝트 문서와 파일을 공동 작업할 수 있는 문서 관리 솔루션이다. Nifty Docs는 Google Docs 오픈 소스 코드를 기반으로 구축되었으므로 모든 Google Docs 기능과 프로젝트 관리 소프트웨어의 강력한 기능을 결합할 수 있다.

(2) DiffChecker

Diffchecker는 텍스트를 비교하여 두 텍스트 파일 간의 차이점을 찾는 도구이다. 추가 기능은 데스크톱용 diff checker를 다운로드하여 설치하거나 Bibcitation 도구를 이용한다. 무료 온라인 도구는 인용, 참조 목록 및 참고문헌을 생성한다.

(3) Draftable

Workshare Compare는 놀라운 안정성과 보안 기능을 위해 수많은 법률 팀이 사용하는 도구이다. 법률 분야에서는 의도하지 않은 실수 방지를 위해 작업 공유 비교 방법을 사용하면 원본 문서에 대한 변경 사항이나 설명이 필요하지 않는다. 이 도구는 Office Suite에서 가장 잘 작동하도록 설계되었으며 Word, Paint 및 Excel 파일에 사용할 때 좋은 결과를 제공한다.

(4) Beyond Compare

비욘드 컴페어(Beyond Compare)는 빠르고 쉽게 파일과 폴더를 비교 및 병합하며, 동기화 시키고, 이에 대한 결과 보고서를 생성할 수 있는 비교/병합 도구이다. 텍스트 비교와 폴더 비교가 대표하는 기능으로써, 보기 쉬운 UI와 함께 빠른 속도로 차이점을 분석해낸다. Beyond Compare는 문서 비교와 폴더 비교 외에도 이미지, 데이터 테이블(엑셀 포함), MP3, 레지스트리, 응용프로그램(바이너리 비교) 등 비교/병합 툴 중에서 가장 많은 파일 포맷을 지원한다.

(5) Workshare Compare

Workshare Compare는 작업 공유 비교 기능을 사용하면 문서를 비교할 수 있다. 또한 문서 편집 및 수정 과정을 최대한 빠르고 효율적으로 만들어 주도록 설계된 문서 비교 도구이다. Workshare Compare를 사용하면 원본 문서와 수정된 문서를 비교하고, 비교(Redline) 문서를 생성하여 두 문서의 차이점을 즉시 확인할 수 있다.

(6) ExamDiff

원클릭 재 비교가 가능하며 드래그 앤 드롭을 지원한다. 파일 변경 사항을 자동으로 감지하는 기능을 사용하면 정확한 결과를 제공한다. 플랫폼은 사용하기 매우 쉽기때문에 인터페이스 탐색 방법을 사용하는 데 어려움이 없다. 이 인터페이스는 완전히 사용자 정의가 가능하므로 필요에 맞게 플랫폼을 구성할 수 있다.

(7) AraxisMerge

AraxisMerge는 3방향 파일 비교, 병합 및 폴더 동기화를 제공할 수 있다. 3방향 파일 비교의 필요성이 강조되고 있다. 고급 기능과 관련하여 이미지 및 바이너리 비교(픽셀 및 바이트 수준)는 타의 추종을 불허한다.

AraxisMerge의 가장 강력한 기능 중 하나는 HTML, XML 또는 UNIX로 생성할 수 있는 이식 가능한 보고서이다. 이를 무기한 저장하고 동료 등과 공유하는 것은 매우 간단하다.

(8) Windiff

두 폴더의 내용을 비교할 때 차이점을 알 수

있는 도구이다. 폴더를 비교하는 것도 더 이상 복잡하지 않다. MS Windows NT 4.0 Resource Kit 지원 도구 패키지에 Windiff가 이미 포함되어 있으므로 Windiff를 개별적으로 다운로드할 필요가 없다.

(9) UltraCompare

UltraCompare는 파일과 폴더를 비교하고, 파일을 병합하고, 원격 파일을 비교하고, 파일을 동기화할 수도 있다. 개인용 라이센스를 사용하면 세 대의 컴퓨터를 동시에 조합하여 사용할 수 있고 휴대폰, 노트북 및 데스크톱이 동일한 OS에서 실행되지 않을 수 있으므로 이는 특히 중요하다.

(10) Kompare

Linux용 비교 플랫폼에서 Kompare이 가장 강력하다 할 수 있다. 이 도구는 사용 방법 간단하고, 사용자 정의를 위한 디스플레이를 갖추고 있으며, 지속적으로 새로운 업데이트가 가능하다. 패치 파일을 비교할 때 컨텍스트, 통합 및 diff 형식의 그래픽 보기가 필요하다.

3.3 데이이터베이스 표절(유사도) 산출 도구

(1) Unix/Linux utility - comm

DB 스키마를 비교하기 위하여 그림 4와 같이 Linux/Unix에서 제공하는 comm 명령어를 이용하여 동일 라인을 검사할 수 있다. DB 스키마의 경우 일부 속성 값이 변하면 스키마에 영향을 주기 때문에 comm명령은 정렬된 두 파일을 한 줄씩 비교하는데 사용한다. 또한 대소문자를 구분하지 않고 행을 비교하는 기능도 지원한다.

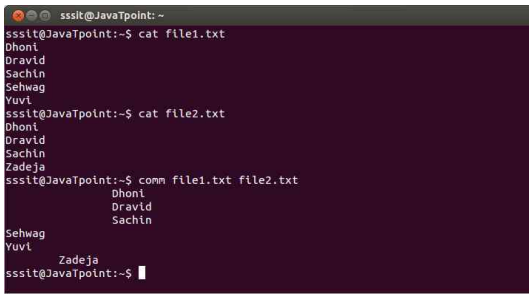


그림 4. 유닉스/리눅스 - comm
Fig. 4. Unix/Linux utility - comm

(2) Cross-Database Studio 9.0

Cross-Database Studio는 그림 5와 같이 Business의 데이터베이스 및 도구 소프트웨어이다. 개발한 회사는 DBBalance Ltd.이다. 개발사가 출시한 최신 버전은 9이다. 데이터베이스를 비교하기 위한 도구로, 테이블 명, 칼럼명 비교를 위해 사용된다[12].

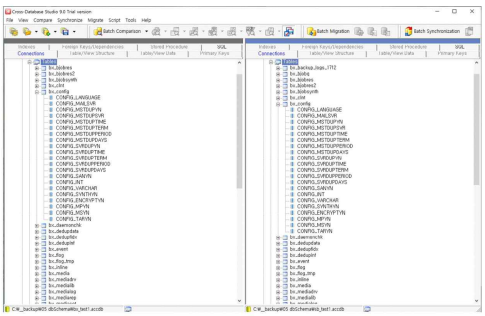


그림 5. Cross-Database Studio
Fig. 5. Cross-Database Studio

(3) SQL Server Management Studio

SQL Server Management Studio(SSMS)는 그림 6과 같이 SQL Server 및 데이터베이스 인스턴트 구성, 모니터링 및 관리하는 도구이다. 데이터베이스의 테이블 명과 컬럼 구조, 테이블 구조를 분석하기 위해 사용한다.

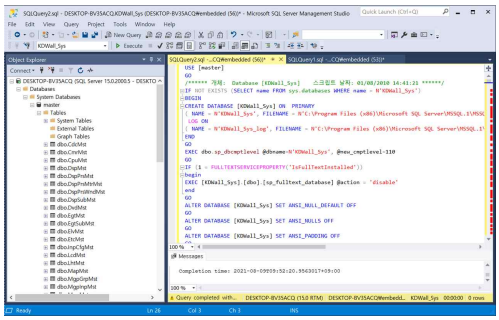


그림 6. SQL Server Management Studio(SSMS)
Fig. 6. SQL Server Management Studio(SSMS)

3.4 기계어 표절(유사도) 산출 기술

(1) JD(Java Decompiler)-GUI : 역 컴파일 (java 바이트 코드 -> 소스 코드)

Java Decompiler는 그림 7과 같이 자바를 디컴파일하고 분석하기 위한 도구이다. JD-Core, JD-GUI 및 JD-Eclipse는 GPLv3 라이센스에 따라 릴리스된 오픈 소스이다. JD-GUI 도구는 바이트 코드로 컴파일되어 있는 자바 클래스 파일의 소스코드로 변환하기 위한 역컴파일 도구이다. class 확장자를 가진 바이너리 파일을 java 형태의 소스코드로 변환하여 분석할 수 있는 기능을 제공한다[13].

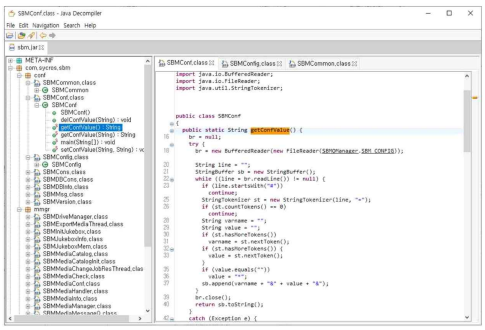
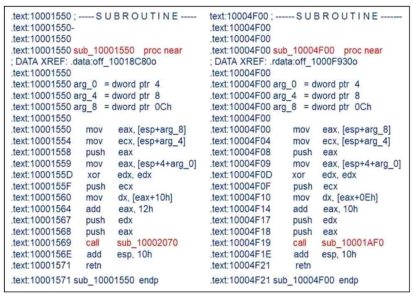


그림 7. JD(Java Decompiler)
Fig. 7. JD(Java Decompiler)

(2) IDA((Interactive Disassembler) - 역컴파일

IDA는 그림 8과 같이 역 어셈블러로서 IDA Pro는 프로세서에 의해 실제로 실행되는 바이너리 명령을 기호 표현(어셈블리 언어)으로 표시하기 위해 실행 맵을 생성할 수 있다. IDA Pro에는 기계 실행 가능 코드에서 어셈블리 언어 소스 코드를 생성하고 이 복잡한 코드를 사람이 더 쉽게 읽을 수 있도록 고급 기술이 구현되었다. 그림 8은 실행 파일에 대해 역 어셈블리 도구인 IDA를 이용하여 생성한 어셈블리어 코드의 예이다[14].



```
.text:10001550 ; ------ S U B R O U T I N E ------
.text:10001550-
.text:10001550
.text:10001550 sub_10001550   proc near
; DATA XREF:  data:off_10018C80o
.text:10001550
.text:10001550 arg_0  = dword ptr  4
.text:10001550 arg_4  = dword ptr  8
.text:10001550 arg_8  = dword ptr  0Ch
.text:10001550
.text:10001550      mov     eax, [esp+arg_8]
.text:10001554      mov     ecx, [esp+arg_4]
.text:10001558      push    eax
.text:10001559      mov     eax, [esp+4+arg_0]
.text:1000155D      xor     edx, edx
.text:1000155F      push    ecx
.text:10001560      mov     dx, [eax+10h]
.text:10001564      add     eax, 12h
.text:10001567      push    edx
.text:10001568      push    eax
.text:10001569      call    sub_10002070
.text:1000156E      add     esp, 10h
.text:10001571      retn
.text:10001571 sub_10001550 endp
```

```
.text:10004F00 ; ------ S U B R O U T I N E ------
.text:10004F00
.text:10004F00
.text:10004F00 sub_10004F00   proc near
; DATA XREF:  .rdata:off_1000F930o
.text:10004F00
.text:10004F00 arg_0  = dword ptr  4
.text:10004F00 arg_4  = dword ptr  8
.text:10004F00 arg_8  = dword ptr  0Ch
.text:10004F00
.text:10004F00      mov     eax, [esp+arg_8]
.text:10004F04      mov     ecx, [esp+arg_4]
.text:10004F08      push    eax
.text:10004F09      mov     eax, [esp+4+arg_0]
.text:10004F0D      xor     edx, edx
.text:10004F0F      push    ecx
.text:10004F10      mov     dx, [eax+0Eh]
.text:10004F14      add     eax, 10h
.text:10004F17      push    edx
.text:10004F18      push    eax
.text:10004F19      call    sub_10001AF0
.text:10004F1E      add     esp, 10h
.text:10004F21      retn
.text:10004F21 sub_10004F00 endp
```

그림 8. IDA((Interactive Disassembler)
Fig. 8. IDA((Interactive Disassembler)

4. 분야별 감정 도구의 분석 결과

본 논문에서의 소프트웨어 분야별(자연어, 데이터베이스, 기계어) 감정 도구가 어떤 것들이 있는지, 어떻게 사용되고 있는지 그 특성 분석에 그 의의 두고있다.

소프트웨어 표절(유사도) 산출 도구에서 주로 사용되는 감정 도구는 구조적 매트릭스 방법을 이용한 YAP3와 한국저작권위원회에서 개발한 exEyes 5.1가 유용한 도구로 보여 진다. 특히 Source Insight는 소프트웨어의 구조적 콜 그래프 형태의 분석에 아주 유용한 도구로 볼 수 있다.

자연어(문서) 표절(유사도) 산출 도구에서는 빠르고 쉽게 파일과 폴더를 비교 및 병합하는 Beyond Compare와 UltraCompare 도구가 가장 많이 사용되고 있다.

데이이터베이스 표절(유사도) 산출 도구는 그 분야가 특수하여 분석 도구가 많지 않으며 Cross-Database Studio와 SQL Server Management Studio가 가장 많이 사용되고 있는 분석 도구이다.

기계어 표절(유사도)은 산출 도구에서는 역컴파일 도구 감정 도구가 필요하며 JD-GUI와 IDA Pro가 많이 사용되고 있는 감정 도구이다.

5. 결 론

본 논문에서는 유사도 평가를 위한 도구를 조사 연구해 보았다. 자연어, 데이터베이스, 기계어 소프트웨어에 특성별 적합한 유사도 분석 도구가 어떤 것이 있는지 분석하였다.

본 논문의 의의는 감정을 처음 할때 소프트웨어 분야별 감정 도구를 일일이 찾아볼 필요가 없이 분야별 감정 도구를 쉽게 파악할 수 있는데 그 의의가 있다.

향후 인공지능에 의해 제작된 소프트웨에 대한 유사도 분석에 대한 연구가 진행되어야 된다.

본 연구는 한경국립대학교 2024년도 학술연구조성비의 지원에 의한 것임(영문: This work was supported by a research grant from Hankyong National University in the year of 2024.)

참 고 문 헌

[1] Yunseok Pak, “A Study on unfair competition of intellectual property infringement”, Korea Institute of Intellectual Property, vol.13, no.3, pp. 67-96 (30 pages) 2018, DOI : 10.34122/jip.2018.09.13.3.67

[2] Korea Software Property Right Council URL https://www.spc.or.kr/ko/index

[3] Stefan Bellon, Rainer Koschke, Giuliano Antoniol, Jens Krinke, and Ettore Merlo, “Comparison and evaluation of clone detection tools”, IEEE Transactions on Software Engineering, Vol.33,No.9, pp.577-591, 2007.

[4] Chanchal K. Roy, JamesR. Cordy, and Rainer Koschke, “Comparison and evaluation of code clone detection techniques and tools: A qualitative approach”, Science of Computer Programming, Vol.74,No.7, pp.470-495, 2009.

[5] Chanchal K. Roy et al., “Comparison and evaluation of techniques code clone detection and tools: A qualitative approach”, Proc. of the Science of Computer Programming Volume 74, Issue7, pp.470-495, 1 May 2009, https://doi.org/10.1016/j.scico.2009.02.007

[6] D. Gitchell, N. Tran, “SIM: A Utility for Detecting Similarity in Computer Programs”, in Proc. of 30th SIGCSE Technical Symposium on Computer Science Education, New Orleans, USA, pp.266~270, May 1999. https://doi.org/10.1145/299649.299783

[7] G. Whale, Plague: Plagiarism Detection using Program Structure, TR Vol.8805, Department of Computer Science, University of NSW, Kensington, Australia, 1988.

[8] M. Wise, “YAP3: Improved Detection of Similarities in Computer Program and OtherTexts”, in Proc. Of 27th SIGCSE Technical Symposium, Philadelphia, USA, pp.130~134, 1996. https://doi.org/10.1145/236462.236525
[9] A. Aiken, MOSS: A System for Detecting Software Similarity, Stanford University, USA, 2020. http://theory.stanford.edu/~aiken/moss/
[10] Prechelt, L., Malpohl, G., Philippsen, M. “Finding plagiarisms among a set of programs with JPlag”, Journal of Universal Computer Science, 2002, vol. 8, no. 11, pp.1016-1038.
[11] Sungha Choi, Kyung-Goo Doh, “Assessment of exEyes′ Recall using Bellon Reference Corpus as a Benchmark”, Vol.1, 2015. Journal of Software Assessment and Valuation, http://www.ksavs.or.kr/html/paper/2015-1/(4)2015-1.pdf
[12] Cross-Database Studio 9.0 : http://www.dbalance.com/
[13] JD(Java Decompiler) : http://java-decompiler.github.io
[14] IDA((Interactive Disassembler) : https://hex-rays.com/ida-pro/

저 자 소 개

이창훈(Chang-Hoon Lee)

1998. 8. 중앙대학교 공학박사
2002.3-현재 한경국립대학교 컴퓨터응용수학부 교수
<주관심분야> 소프트웨어 공학, 객체지향 패턴설계