

## 한글 프로그래밍 언어 “새싹”을 위한 과제 평가 시스템 개발 Development on an Assignment Evaluation System for “Saesark,” a Korean Programming Language

김연어, 류샤오, 우균\*

부산대학교, 부산대학교/LG전자 스마트제어센터\*

Yeoneo Kim, Xiao Liu, Gyun Woo\*

Pusan National Univ., Pusan National Univ.,  
Smart Control Center of LG Electronics\*

### 요약

최근 국내외에서 프로그래밍 교육이 주목받고 있다. 그중 초보자를 위한 한글 프로그래밍 언어인 새싹은 소프트웨어 프로그래밍 교육에 적합한 언어이다. 하지만 새싹을 위한 과제 평가 시스템은 아직 개발되어 있지 않다. 이 논문에서는 새싹을 위한 과제 평가 시스템을 개발하고자 한다. 이를 위해 기존 과제 평가 시스템인 neoESPA를 수정하여 새싹을 지원하도록 한다. 그리고 새싹을 위한 과제 간의 표절 여부를 판단하는 도구를 개발하고 이를 neoESPA에 탑재하여 정당한 평가를 도와주는 과제 평가 시스템을 제안한다.

### I. 서론

최근 스마트폰의 보급으로 수많은 애플리케이션이 개발되고 있으며, 이로 인한 1인 개발자 시대가 열렸다고 봐도 과언이 아니다. 이에 발맞춰 최근에는 국내외의 초·중학교에서 프로그래밍 교육을 의무화하겠다는 발표가 이어지고 있다[1]. 하지만 대학교의 프로그래밍 교육에서도 일부 학생이 프로그래밍 수업을 따라가지 못하는 경우도 종종 발생하는 상황에서 초, 중학생을 대상으로 프로그래밍 수업은 어려운 문제이다.

프로그래밍 언어 측면에서도 쉬운 프로그래밍 수업을 위해 다양한 연구가 계속 진행되고 있으며 대표적인 연구로는 프로그래밍을 모국어로 하는 방법이 있다. 기존 프로그래밍 언어는 대부분이 영어로 작성되어 있어 초보자의 경우 영어를 해석하는데 많은 시간을 소모하기 때문에 모국어로 작성 가능한 프로그래밍 언어는 초보자에게 많은 도움이 된다. 기존에 다양한 한글 프로그래밍 언어가 개발되었고, 가장 한글에 가깝게 사용할 수 있는 언어로는 “새싹”이 있다[2]. 하지만 새싹은 Eclipse 기반의 개발 환경을 제공하고 있지만, 과제 평가를 위한 환경은 아직 제공되고 있지 않다.

이 논문에서는 한글 프로그래밍 언어인 “새싹”을 위한 과제 평가 시스템을 제안하고자 한다. 새싹을 위한 과제 평가 시스템은 과제를 출제 및 제출하는 기능뿐만 아니라 과제 간의 표절 검사 기능까지 제공하여야 한다. 이는 프로그래밍 수업에서 표절 검사가 이루어지지 않는다면 정당한 평가가 이루어지지 않기 때문이다, 이를 위해 이 논문에서는 기존 평가 시스템인 neoESPA[3]를 수정하여 새싹을 위한 과제 평가 시스템을 구현하고자 한다.

이 논문은 다음과 같이 구성되어 있다. 2장에서는 기존 과제 평가 시스템인 neoESPA에 새싹 컴파일러 기능을 추가한다. 3장에서는 새싹을 위한 표절 검사기를 개

발한다. 그리고 4장에서 결론을 맺는다.

### II. 새싹을 위한 과제 평가 시스템

이 장에서는 기존 과제 평가 시스템인 neoESPA를 수정하여 새싹을 위한 과제 평가 시스템을 개발하고자 한다. 과제 평가 시스템에서 필요한 기능으로는 회원 관리 시스템, 과제 출제, 과제 제출 기능이 필요하다. 이러한 기능은 프로그래밍 언어와 무관한 기능이기 때문에 neoESPA에서 제공하고 있는 기능을 그대로 사용할 수 있다. 즉 프로그래밍 언어와 관련이 있는 컴파일 수행 단계와 오류 및 실행 결과를 보여주는 기능을 추가해야 한다. neoESPA에서 컴파일러 과정 및 실행 결과를 추가하기 위해서는 neoESPA의 PHP 코드 수정이 필요하며 이를 수정한 코드는 그림 1과 같다.

```

...
case saesark:
    $this->saesarkCompiler();
    $this->send_info();
...
function saesarkCompiler() {
    system("spc " . $this->store_dir/
    $this->upload_file_name 2> " . "
    $this->store_dir/compile_error", $res1);
    if (filesize("$this->store_dir/compile_error") == 0) {
        $this->compile_msg = 1;
        for ($i = 0; $i < $this->hwNum; $i++) {
            $j = $i + 1;
            $res = substr(system("cd $this->store_dir;
            (time p java $this->shortfilename < .././
            $this->rating_data_dir/hw$this->hw/$j.inp >
            $j.out)>2&1 | grep real"), 5);
            echo "result time :$res<br>";
            if ((int) $res > (int) $this->sec) {
                file_put_contents("$this->store_dir/$j.out",
                "timeout-1");
            }
        }
        $this->rate_score();
    } else {
        $this->compile_msg = 2;
    }
}

```

▶▶ 그림 1. 새싹 컴파일러 추가 코드

그림 1의 첫 문단의 코드는 웹 화면상에서 새싹 컴파일러가 선택되었을 때 실행되는 코드로 saesarkCompiler 함수를 호출한다. 두 번째 문단의 saesarkCompiler 함수는 입력된 소스 코드를 이용하여 컴파일을 수행하고 오류 메시지나 실행 결과를 웹으로 표시해주기 위한 과정을 표현한 함수이다. 그리고 그림 2는 웹상에서 새싹 컴파일러 옵션을 추가한 웹 화면이다.



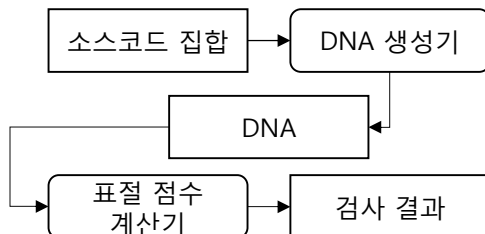
▶▶ 그림 2. 새싹 컴파일러를 추가한 neoESPA

그림 2는 neoESPA 화면은 과제를 제출하기 위한 화면이다. neoESPA는 기본적으로 C/C++, Java, Python과 같은 컴파일러를 지원하고 있다. neoESPA에 새싹을 지원하기 위해 새싹 컴파일러를 선택할 수 있는 옵션을 추가하였으며 새싹을 선택 후 소스 코드를 올리면 그림 1의 코드가 실행되어 과제 평가가 이루어진다.

### III. 새싹을 위한 표절 검사 도구 개발

이 장에서는 제출된 새싹 소스 코드를 대상으로 표절 여부를 판단하는 도구를 개발한다. 프로그래밍 교육에서 표절을 검사하는 일은 학생 간에 정당한 평가를 위해서도 꼭 필요한 일이다. 하지만 사람이 직접 표절 여부를 판단하기에는 많은 시간이 소모되기에 자동화된 도구가 필요하다.

이 논문에서는 여러 표절 검사 기법 중 토큰 정보를 이용하여 표절을 검사하는 기법인 SoVAC (Software Verification and Analysis Center)을 이용하고자 한다[4]. 이 방법은 파싱을 통해 생성된 토큰 정보를 DNA로 변환하고, DNA 간의 유사도를 지역 정렬(local alignment) 알고리즘을 이용하여 계산하여 표절 여부를 판단한다. 이 기법의 전체적인 구조는 그림 3과 같다.



▶▶ 그림 3. 표절 검사 도구 구조

새싹을 위한 표절 검사 도구를 개발하기 위해 그림 3과 같이 DNA 생성기와 표절 점수 계산기를 개발하고자 한다. DNA 생성기는 새싹 코드를 파싱하여 DNA를 생성

해야 한다. 이를 위해 이 논문에서는 새싹 컴파일러를 수정하여 DNA를 생성할 수 있도록 수정하였다. 그리고 표절 점수 계산기를 구현하기 위해서는 DNA 간의 유사도를 지역 정렬을 위해 계산하고 계산된 점수를 백분율로 표시하여야 한다. 이러한 과정은 식 1과 같이 표현할 수 있다.

$$SIM_{AB} = \frac{LA(A, B)}{\min(LA(A, A), LA(B, B))} \quad (1)$$

식 1에서 A, B는 표절 검사 대상의 DNA를 의미하며 LA 함수는 지역 정렬 함수를 의미한다. 그리고 A, B 사이의 유사도 점수를 정규화하기 위해 자기 자신과의 지역 정렬 값 중에서 작은 값을 사용한다. 이 논문에서는 표절 점수 계산기를 Java로 구현하였다. 그리고 개발된 새싹을 위한 표절 검사 도구는 단독으로도 실행할 수 있지만 neoESPA에 탑재하여 제출된 과제에 한하여 과제별 표절 검사 판단이 가능하도록 개발하였다.

### IV. 결론

이 논문에서는 새싹을 위한 과제 평가 시스템을 개발하였다. 이를 위해 기존 과제 평가 시스템인 neoESPA를 개선하여 새싹 컴파일 및 실행을 할 수 있도록 하였다. 그리고 새싹을 위한 표절 검사 도구를 개발하였으며 이를 neoESPA 상에서 사용할 수 있도록 하였다.

향후 연구로 현재 새싹용 표절 검사 도구는 단순히 토큰의 나열을 DNA 변경한 수준에 그치고 있다. 이를 프로그램 실행 흐름을 고려하여 DNA를 생성하도록 변경할 예정이다. 또한, 실제 교육 환경에 제안한 과제 평가 시스템을 사용하여 성능을 점차 개선해 나갈 예정이다.

### ACKNOWLEDGMENT

이 논문은 2013년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(NRF-2013R1A1A4A01006704)

\*교신 저자 : 우균(부산대학교, woogyun@pusan.ac.kr)

### ■ 참고 문헌 ■

- [1] 권상희, "세계는 프로그래밍 교육 바람," <http://www.etnews.com/20160419000130> (16.04.27 방문).
- [2] 친준석, 우균, "새싹: 초보자를 위한 한글 객체 지향 프로그래밍 언어," 한국콘텐츠학회논문지, 제 16권, 제 3호, pp. 288-295, 2016.
- [3] X. Liu, Y. Kim, H. Cho and G. Woo, "NeoESPA-A New Evaluation System for Programming Assignments," Proceedings of the 14th International Conference on Electronics, Information, and Communication (ICEIC 2015), 2015.
- [4] J. Ji, G. Woo and H. Cho, "A Source Code Linearization Technique for Detecting Plagiarized Programs," ACM SIGCSE Bulletin, Vol.39, No.3, pp.73-77, 2007.