

프로그래밍 학습 시스템에서 자동 평가 기능*

선동언† · 류기곤†† · 김현철††

† 고려대학교 컴퓨터교육학과 · †† 고려대학교 컴퓨터학과

Comparison of Auto Graders in Programming Tutoring Systems

DongEun Sun† · Kigon Lyu†† · Hyeon-Cheol Kim††

† Dept. of Computer Education, Korea University · †† Dept. of Computer Science,
Korea University

요 약

프로그래밍 학습 시스템에서 자동 평가 기능은, 코드에 대한 실행 결과 외 다른 피드백을 제공할 수 있기 때문에, 그 피드백을 통해 학습자 스스로가 자신의 코드를 개선할 수 있다. 본 논문에서는 다양한 프로그래밍 학습 시스템에서 사용중인 자동 평가 기능을 비교하고, 지능형 프로그래밍 튜터링 시스템인 Everycoding에서의 새로운 자동 평가 기능을 소개한다.

1. 서론

MOOC의 등장으로 또다시 자동 평가에 대한 관심이 증대되고 있다. 대규모 수업에서는 사람이 직접 평가를 하기에는 비용 문제, 평가의 일관성 등 다양한 문제가 발생하기 때문이다. 전통적으로 자동 평가는 OX 문제나 4지 선다 문제, 단답형 문제와 같은 정답의 일치 여부를 명확히 할 수 있는 문제에 국한이 되어 적용되었다. 이러한 한계로 자동 평가는 교육적 목적에서 평가 보다는 단순한 측정에 가까웠다. 그러나 인공 지능의 발전으로 교육적 효과를 극대화 시키기 위한 다양한 연구[1][2][3]가 시도되고 있다.

프로그래밍에서 자동 평가[4]는 1960년대부터 연구되어 왔다. MOOC의 등장과 더불어 프로그래밍의 중요성이 인지됨에 되고, 자동 평가를 교육적 목적에서 활용하려는 새로운 시도들[5][6]이 진행되고 있다. 특히, 프로그래밍은 일반적으로 결과에 대한 피드백이 제공되지만 자동 평가 기능은 결과 외 다른 피드백을 줄 수 있기 때문에 이를 통해 학습자 스스로가 자신이 작성한 코드에 대한 반성적 사고와 코드에 대한 새로운 인지를 할 수 있다. 본 논문에서는 특정 문제에 중

속적인 방법이 아니라 범용적으로 사용되고 있는 프로그래밍에 대한 자동 평가 기능이 적용된 시스템에 대한 비교 및 지능형 프로그래밍 학습 시스템 Everycoding[7]에 활용중인 자동 평가 기법을 소개한다.

2. 프로그래밍 자동 평가 기능

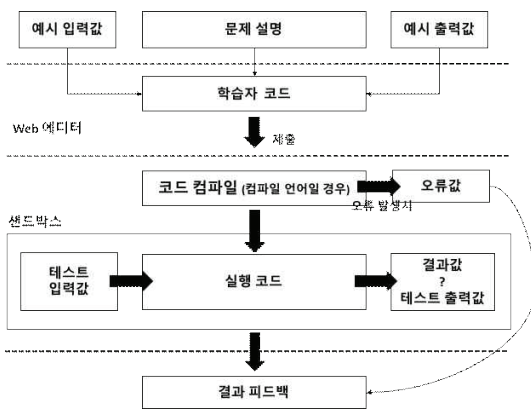
자동 평가 기능은 특정 문제만 적용 가능한 기법과 범용적 기법으로 구분할 수 있다. 범용적 기법으로 프로그래밍 자동 평가가 적용되고 있는 시스템은 다음과 같이 크게 2종류로 나눌 수 있다.

- 온라인 저지(Online Judge) 시스템
- 프로그래밍 학습 시스템

자동 평가 기능이 가장 많이 적용되고 있는 시스템은 온라인 저지 시스템이다. 온라인 저지 시스템은 원래 프로그래밍 대회에서 프로그램들을 시험할 목적으로 개발된 시스템이지만, 프로그래밍 교육용으로도 사용되고 있다. 교육용으로 사용할 때는 알고리즘이 아닌 문법에 대한 문제가 제시되는 것이 일반적이다. 온라인 저지 시스템의 기본 구조는 [그림 1]과 같다. 학습자에게 문제와 예제 입출력 값을 제시하면, 학습자는 문제와 입출력 값을 고려해서 코드를 작성하고 시

* 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음 (2015-0-00936)

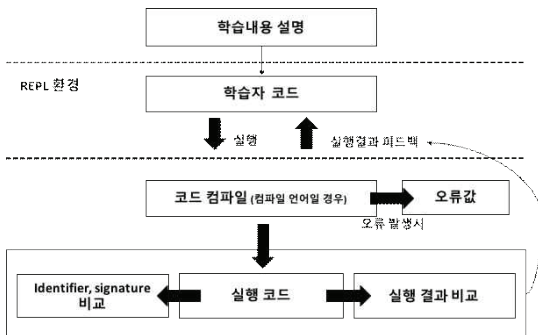
시스템에 제출한다. 시스템은 코드를 컴파일하고 실행하고, 미리 작성된 데이터로 테스트 하게 된다. 제출된 코드는 시간, 메모리, 보안 등에 관련된 제한을 하기 위해서 자동 평가 부분을 샌드박스화 하는 것이 일반적이다. 온라인 저지 방식의 자동 평가 기능의 장점은 작성한 코드의 정오답 뿐만 아니라 시간, 메모리 등의 결과값을 제공하기 때문에, 다른 학습자와의 코드 비교를 통해서 프로그래밍을 개선시킬 수 있다. 그러나 입력과 출력의 일치값으로만 코드를 평가하기 때문에, 올바른 평가를 하지 못하는 경우가 종종 발생한다.



[그림 1] 온라인 저지 시스템의 자동 평가 기능

일부 온라인 저지 시스템[10]의 경우, 프로그래밍 학습시스템형 자동 평가와 같이 functional signature 등의 일치여부를 평가하지만 마찬가지로 입력값과 출력값을 비교해서 최종 평가한다.

Code.org, CodeAcademy로 대표되는 프로그래밍 학습 시스템은 온라인 저지형과 달리 [그림 2]와 같이 학습 내용에 대한 REPL(Read - eval - print loop) 실행 환경을 구성해서 평가보다는 실제 코딩을 한다는 느낌으로 결과를 확인할 수 있다.



[그림 2] 프로그래밍 학습 시스템의 자동 평가 기능

입출력값만을 비교하지 않기 때문에, 제시된 조건 (identifier, signature)의 일치 여부도 함께 평가할 수 있다. 예를 들어[9], my_lint를 7에서 3으로 재할당하고 출력하라는 문제에서 my_lint 대신 my_lin2로 변수를 선언했을 시 identifier와 signature가 다르기 때문에 오답으로 평가한다. 이러한 프로그래밍 학습시스템 자동 평가 방식은 학습자가 세세한 문법 기능까지 학습하는데 온라인 저지 방식보다 뛰어나다. 그러나 Code Academy의 경우[8], "Welcome to Python!"을 출력하는 문제에 대해서,

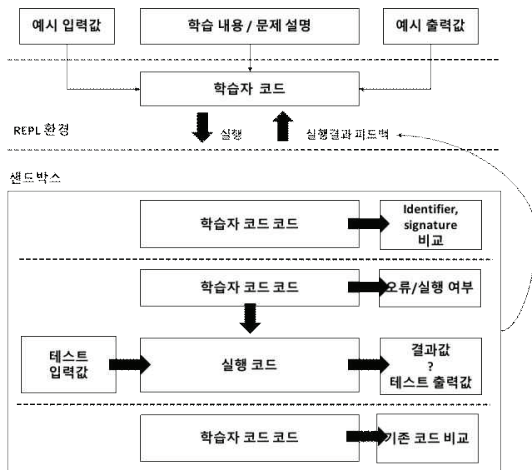
```

print 'a'
print 'Welcome to Python!'
    
```

으로 문제에 대한 코드를 실행할 경우, 온라인 저지의 경우 출력값이 다르기 때문에 오답으로 평가되지만, print 'Welcome to Python!'이 포함되기 때문에 정답으로 평가한다. 이는 입력값 대 출력값 일치 여부를 확인하지 않는 데서 오는 단점이다.

3. Everycoding의 자동 평가 기능

Everycoding은 지능형 프로그래밍 튜터링 시스템이다. 앞서 소개한, 온라인 저지 시스템과 프로그래밍 학습 시스템과 달리, 학습자 개인의 특성을 고려한 맞춤형 프로그래밍 교육을 지원한다. 기존 Everycoding[11]에서는 문제 해결형 콘텐츠에 대한 코딩 학습 기능을 제공하는 코딩 튜터에서는 프로그래밍 학습시스템형 자동 평가 기능을, 학습한 기능을 실습해 볼 수 있는 코딩 도전에서 온라인 저지형 자동 평가 기능을 제공했다. 새로운 버전의 코딩 도전에서는 [그림 3]과 같이 온라인 저지형 자동 평가 구조에 프로그래밍 학습형 자동 평가 기능 구조 및 학습자 모델과 연동해 학습자의 기존 코드와 비교하는 기능을 추가하였다. 특히 멀티 코어 프로그래밍 기법을 활용해서 자동 평가에 소비되는 시간을 최소화하였다.



[그림 3] Everycoding 시스템의 자동 평가 기능

4. 결론 및 논의

지금까지 대표적인 프로그래밍 학습 관련 시스템별 자동 평가 기능에 대하여 알아보았다. 자동 평가 기능은 단순한 코딩 결과가 아닌 다른 피드백을 제공할 수 있기 때문에, 교육적 목적으로 활용 가치가 높다. 그러나 아직까지 평가 항목 및 내용이 사람이 평가하는 것에 비해서 부족하다. 최근 들어, 딥러닝 기법을 활용하여 코드에 대한 분류를 사람처럼 하는 연구에 대한 성과가 있었다. 자동 평가 또한 차후 딥러닝 기법을 활용한다면 사람처럼 코드를 평가하는 기능이 개발될 것으로 기대한다.

참고 문헌

[1] Balfour, S. P. (2013). Assessing writing in MOOCs: Automated essay scoring and calibrated peer review(TM). *Research & Practice in Assessment*, 8

[2] Sandeen, C. (2013). Assessment's place in the new MOOC world. *Research & Practice in Assessment*, 8(1), 5-12.

[3] Joseph, A. I. M., & Nath, B. A. (2013). Integration of massive open online education (MOOC) system with in-classroom interaction and assessment and accreditation: *An extensive report from a pilot study. Paper presented at*

the 105-113.

[4] Ack Hollingsworth. 1960. Automatic graders for programming classes. *Commun. ACM* 3, 10 (October 1960), 528-529.

[5] Petri Ihantola, Tuukka Ahoniemi, Ville Karavirta, and Otto Seppälä. 2010. Review of recent systems for automatic assessment of programming assignments. *In Proceedings of the 10th Koli Calling International Conference on Computing Education Research (Koli Calling '10)*. ACM, New York, NY, USA, 86-93.

[6] Vreda Pieterse. 2013. Automated Assessment of Programming Assignments. *In Proceedings of the 3rd Computer Science Education Research Conference on Computer Science Education Research (CSERC '13)*, Article 4, 12

[7] <https://everycoding.net>

[8] https://www.codecademy.com/courses/introduction-to-python-6WeG3/0/1?curriculum_id=4f89dab3d788890003000096

[9] https://www.codecademy.com/courses/introduction-to-python-6WeG3/0/3?curriculum_id=4f89dab3d788890003000096

[10] <https://dmoj.ca/>

[11] 선동언, 2017. 코딩 교육에서의 자동 평가 시스템 활용. **정보처리 학회 2017 춘계 학술발표대회**.