УДК 37.014.1, 004.588

К РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ ОБУЧЕНИЯ ДИФФЕРЕНЦИРОВАНИЮ ФУНКЦИЙ

Заркуа Теодор

Европейский университет, проспект Давида Гурамишвили 76, Тбилиси, Грузия

Аннотация

Автор, опираясь на результаты, полученные им в направлении разработки инструментальных средств для автоматизации обработки функиональных выражений на уровне алгоритмических языков, предлагает подход для реализации сценария обучающей программы, покрывающей полный цикл обучения приемам аналитического дифференцирования функций.

Abstract

The author, relying on the results obtained by him in the direction of developing tools for automating the processing of functional expressions at the level of algorithmic languages, suggests an approach for implementing a script of a training program covering a full cycle of learning techniques for analytical differentiation of functions.

Введение

Благодаря программному обеспечению, предназначенному для автоматизации обработки функциональных выражений на уровне алгоритмическогго языка ([6]), стала возможна постановка задачи компьютерной реализации полнофункционального цикла автоматизации обучения дифференцированию функций.

Требования к обучающей программе

Для эффективной автоматизации процесса обучения по той или иной тематике, как правило, необходимо:

- уметь генерировать вопросы обучаемому в условиях управления сложностью вопроса;
- уметь определять правильность ответа, не ограничивая обучаемого жесткими требованиями формата этого ответа.

Применительно к тематике дифференцирования функций вышеизложенное означает, с одной стороны, умение генерировать разнообразные функциональные выражения для иллюстрации того или иного правила дифференцирования, а также для формирования задания обучаемому. А с другой стороны, умение установить тождественность ответа обучаемого с правильным ответом не ограничивая его необходимостью придать ответу вид, считающийся каноническим.

Серъезным подспоръем для решения обоих вопросов является возможность использования программного обеспечения, эквивалентного по своим возможностьям описанному в [6]. Ведь совершенно очевидно, что для генерирования выражения удобнее использовать одну из бесскобочных (т.н. польских) форм записей. Причем каждая из этих форм удобна и для управления сложностью выражения, так как последнее определяется длиной выражения, набором операций (включая элементарные функции), входящих в него и максимальной глубиной стека при его преобразовании к инфиксной форме, что соответствует глубине вложенности скобок. Благодаря результатам, изложенным в [1-5], для наших целей вполне можно ограничиться использованием только префиксной бесскобочной формы функциональных выражений.

Что касается идентификации ответа, то эта проблема сродни той, которая решалась в [6] для упрощения выражения. Упрощение осуществлялось как раз на уровне

префиксной записи на основании определенных правил, которым была дана форма набора шаблонов. И так как для идентификации тождественности выражений только упрощения недостаточно (упрощение не позволяет учитывать, например, эффект от коммутативности той или иной операции или идентичность, достигаемую разными операциями) здесь имеет смысл задаться дополнительными правилами, которые позволят программно реализовать получение некоторого, единственного для всех тождественных выражений, вида данного выражения, которое можно назвать каноническим. В частности, для коммутативных операций необходимо будет определить дополнительное правило, в соответствии с которым однозначно опрелелится последовательность операндов в канонической форме. Например, в соответствии с некоторым упорядочением. Совершенно ясно, что реализация данного подхода позволит исчерпывающе решить задачу идентификации правильного ответа.

Заключение

Вышеизложенный подход позволяет реализовать гибкий сценарий обучения, позволяющий оперативно реагировать на выполнение контрольных заданий обучаемым не только определением следующего шага обучения, но и регулированием степени сложности иллюстрационных примеров и контрольных ответов.

Очевидно, предлагаемый подход можно обобщить на любую обучающую программу, при условии наличия возможности регулирования сложности генерируемых примеров, а также наличия механизма автоматизации идентификации правильного ответа.

В дальнейшем представляется интересным разработка понятия удаленности от правильного ответа. Разумеется, подобная информация позволит еще более точно представить текущий уровень обучаемого и может быть использована для повышения эффективности процесса обучения.

Постольку поскольку, в основу предлагаемого подхода лежат понятия и алгоритмы, не привязанные к конкретному алгоритмическому языку, то соответствующие реализации легко переносимы с одного языка программирования на другой.

Список использованных источников:

1. Заркуа Т. К свойству перевернутой польской записи. Internet-Edication-Sciense-2008. New Informational and Computer Tecnologies in Education and Sciense. H. Intelligence Information Systems. 2008 year, p.543-544

2. Заркуа Т. Некоторые способы обработки функциональных выражений. Winter Programming School. Материалы зимней школы по программированию, Харьков, ХНУРЭ, 2009, p. 205-211

3. Zarkua T. An Approach to Functional Expressions Processing Automation. Billetin №2 of Saint Andrew the First-Called Georgian University of the Patriarchy of Georgia, Tbilisi, 2009, p. 50-61

4. Т.Заркуа.Автоматизация обработки функциональных выражений на уровне алгоритмических языков. Internet-Edication-Sciense-2010. New Informational and Computer Tecnologies in Education and Sciense.p.201-206.

5. Т.Заркуа.К вопросу об использовании понятия сопряженного по отношению к бесскобочным выражениям. Internet-Edication-Sciense-2012. New Informational and Computer Tecnologies in Education and Sciense.p.156-157.

6. Заркуа Теодор. Программное обеспечение на C++ для автоматизации обработки функциональных выражений. Internet-Edication-Sciense-2016. New Informational and Computer Tecnologies in Education and Sciense.p.159-163.