

УДК 37.04:372.8
DOI 10.17513/snt.40375

ОБУЧЕНИЕ АНАЛИЗУ ЛОГИЧЕСКИХ ВЫРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON

¹Козлов С.В., ²Быков А.А.

¹ФГБОУ ВО «Смоленский государственный университет»,
Смоленск, e-mail: svkozlov1981@yandex.ru;

²Филиал ФГБОУ ВО «Национальный исследовательский университет «МЭИ»,
Смоленск, e-mail: alex1by@mail.ru

Цель исследования заключается в оценке эффективности объяснения обучающимся подходов к анализу логических выражений с использованием языка программирования Python. Изучение математической логики в школе является основной дидактической линией и носит междисциплинарный характер. В связи с этим пониманию обучающимися сущности логических операций, последовательности их выполнения и выполнимости логических формул уделяется особое внимание. Наглядное представление табличных данных программными средствами, связь с пропедевтикой и дидактикой программирования обуславливают востребованность применения алгоритмических конструкций при решении заданий анализа истинности логических формул. Восстановление исходных данных логических переменных с помощью инструментов систем программирования позволяет не только раскрыть функциональные особенности логических зависимостей, но и показать применение компьютерных сред. Использование их возможностей открывает перед обучающимися перспективы совершенствования своих навыков при обработке логических типов данных, что служит основой грамотного составления условий проверки в конструкциях ветвлений и циклов. В статье представлены результаты апробации методологических приемов обучения анализу логических выражений в среде Python. Основное внимание уделено правилам записи логических выражений, порядку следования логических переменных, а также возможному расположению наборов логических кортежей в соответствии с уникальностью данных в таблицах истинности. Авторами приводятся примеры основных типов заданий, используемых в экспериментальной работе. Исследование демонстрирует связь теории построения логических выражений и их анализа с использованием современных средств компьютерной практики.

Ключевые слова: информатика, логика, логические выражения, анализ, программирование, язык Python, таблица истинности

LEARNING TO ANALYZE LOGICAL EXPRESSIONS USING THE PYTHON PROGRAMMING LANGUAGE

¹Kozlov S.V., ²Bykov A.A.

¹Smolensk State University, Smolensk, e-mail: svkozlov1981@yandex.ru;

²Branch of National Research University "MPEI", Smolensk, e-mail: alex1by@mail.ru

The purpose of the study is to evaluate the effectiveness of explaining to students approaches to the analysis of logical expressions using the Python programming language. The study of mathematical logic at school is the main didactic line and is interdisciplinary in nature. In this regard, students pay special attention to understanding the essence of logical operations, the sequence of their execution and the feasibility of logical formulas. Visual representation of tabular data by software, connection with propaedeutics and programming didactics determines the demand for the use of algorithmic designs when solving tasks for analyzing the truth of logical formulas. Restoring the original data of logical variables using the tools of programming systems allows not only to reveal the functional features of logical dependencies, but also to show the use of computer environments. Using their capabilities opens up prospects for students to improve their skills in processing logical data types, which forms the basis for competent compilation of verification conditions in branch and loop constructs. The article presents the results of testing methodological techniques for learning to analyze logical expressions in a Python environment. The main attention is paid to the rules for writing logical expressions, the order of logical variables, as well as the possible arrangement of sets of logical tuples in accordance with the uniqueness of the data in the truth tables. The authors provide examples of the main types of tasks used in experimental work. The study demonstrates the connection between the theory of constructing logical expressions and their analysis using modern means of computer practice.

Keywords: computer science, logic, logical expressions, analysis, truth table, programming, Python language

Введение

Математическую логику как элемент пропедевтики начинают освещать еще в начальной школе. Обучающиеся учатся строить и определять истинность простых логических высказываний, выяснять, к какому виду они относятся: единичному, частному или общему. Затем переходят к изучению

составных логических высказываний в составных повествовательных предложениях с использованием союзов «И» и «Или», а также отрицательной частицы «Не». В начале обучения в среднем звене школы они знакомятся с логическими функциями, соответствующими этим частям речи, дают им строгое определение [1, 2]. Обучающиеся

ся на практике пробуют строить противоположные утверждения к составным высказываниям, тренируются применять закон контрапозиции. На диаграммах Эйлера–Венна учителя проводят перед ними параллели между логическими операциями и операциями на множествах, раскрывают сущность установленных взаимосвязей [3, 4]. В продолжение обучающиеся постигают правила составления таблиц истинности сложносоставных высказываний, отрабатывают навыки упрощения логических выражений [5]. На занятиях им демонстрируется приложение элементов математической логики к построению контактно-релейных схем. Обучающиеся при этом в разных темах школьного курса информатики решают логические задания с использованием таблиц, схем, графов [6]. Это помогает им получить полное представление об областях применения инструментов математической логики. В завершение обучения информатике в средней школе при изучении программирования они изучают логический тип данных и правила записи логических выражений в конструкциях ветвлений и циклов в инструментальной системе. Обучающиеся записывают в среде программирования решения практических заданий с помощью композиций основных алгоритмических конструкций, применяя свои навыки логики определения правильной последовательности выполнения инструкций. Таким способом они получают целостное представление о возможностях применения математической логики от теории к практике в разных областях знаний, тем самым осваивая одну из основных дидактических линий школьного курса информатики.

В то же время отметим, что прикладной характер данной дидактической линии раскрывается на занятиях не в полной мере. Так, в программировании при решении заданий невозможно обойтись без обсуждения правил записи простых логических выражений и составления из них сложных. Однако как инструмент построения таблиц истинности логических выражений, определения порядка логических переменных в их записи или вычисления значений пользовательских функций алгоритмические конструкции на уроках практически не используются [7, 8]. Учителями не демонстрируется потенциал систем программирования в задачах анализа значений входных данных логических выражений [9, 10]. В связи с этим необходимо в общеобразовательной практике делать акцент на взаимосвязи дидактических линий школьного курса информатики, взаимопроникновении и переплетении их тем и разделов изучения.

Цель исследования состояла в оценке эффективности объяснения обучающимся подходов к анализу логических выражений с использованием языка программирования Python.

Материалы и методы исследования

Экспериментальная деятельность по обучению анализу логических выражений с использованием языка программирования Python проводилась в Смоленском физико-математическом лицее ЯВИР при МИФИ и МЭИ и средней школе № 6 г. Смоленска. Педагогическое исследование осуществлялось в 10-х классах физико-математического профиля с углубленным изучением информатики в течение 2 месяцев. В эксперименте принимали участие 32 обучающихся.

В педагогическом исследовании использовались такие методы, как:

- анализ педагогической и методической литературы;
- обобщение существующего педагогического опыта;
- формирующий эксперимент;
- констатирующий эксперимент.

Результаты исследования и их обсуждение

Задания на построение таблиц истинности логических выражений составляют основу введения в математическую логику. Умениям оперировать значениями простых выражений, определять последовательность действий в сложносоставных записях, анализировать результаты применения логических функций отводится фундаментальное значение как базису изучения. Формирование навыков построения таблиц истинности логических выражений требует владения этими умениями как инструментом на базовом уровне. Анализ исходного порядка логических переменных при заданных значениях в таблице истинности уже предполагает совсем иной, более высокий уровень навыков умозаключений. В связи с этим обучающиеся достаточно уверенно оперируют навыками логических рассуждений с прямым ходом действий и в недостаточной мере – с обратным порядком. В этой ситуации изменить подход к анализу исходных значений логических выражений можно с использованием современных языков программирования, таких как Python.

Для применения инструментальных возможностей языка программирования Python к решению заданий построения и анализа таблиц истинности логических выражений необходимы определенные знания. Во-первых, обучающиеся должны иметь представление об основных алгоритмиче-

ских конструкциях, таких как следование, ветвление и циклы. Эти знания составляют фундаментальное ядро дидактической линии алгоритмизации и программирования в школе как на профильном, так и на базовом уровне [11, 12]. Обучающиеся должны уметь организовать систему вложенных циклов по количеству переменных в условии заданного логического выражения. При этом используется более компактный по структуре цикл с параметром *for*. Он уже в своей записи содержит параметры начала цикла, конца цикла и шага изменения переменной цикла. Следует отметить, что логические переменные могут принимать два значения – ложь и истина – 0 и 1 соответственно. Для этого можно воспользоваться указанием диапазона, не забывая о том, что в языке программирования Python он представляет собой полуинтервал, левый конец которого включен в заданный промежуток, а правый – нет. Либо для этого можно воспользоваться списком, состоящим из двух элементов – 0 и 1, который бы просматривался в каждом вложенном цикле по числу переменных, содержащихся в заданном логическом выражении. При этом список можно задать в программе единожды.

Во-вторых, необходимо изучить правила записи сложных составных логических выражений. В их конструкциях могут присутствовать три основные логические операции – отрицание, конъюнкция и дизъюнкция. С помощью них можно построить любое логическое высказывание, они образуют базис операций. В то же время в условии задания логического выражения, помимо них, могут использоваться операции импликации и эквивалентности. В связи с этим одним из теоретических подходов, применяемых на практике, может служить методология преобразования логических выражений к базису. Так, импликацию в выражении можно равносильно заменить на дизъюнкцию отрицания посылки и заключения, а эквивалентность – на дизъюнкцию от конъюнкции двух исходных выражений и конъюнкции их отрицаний. При этом, если при «ручном» способе решения поступают именно так, то при использовании системы программирования Python прибегают к замене импликации на арифметическое сравнение выражений с помощью знака «меньше или равно», именно при его выполнении логическое выражение остается истинным, а в противном случае – ложным. Это соответствует определению логической функции импликации. При этом следует заметить, что в языке Python выполняется переопределение семантики функций с одинаковым именем. Если

сравнение применяется к значениям логических выражений, то результатом будет, так же как при сравнении числовых данных, логическая константа *False* или *True*, но при этом приоритет операций будет иным. Сравнение будет производиться в той последовательности, в которой указаны логические переменные в выражении. Другими словами, сравнение «меньше или равно» имеет более сильный приоритет и выполняется раньше логических операций отрицания, конъюнкции и дизъюнкции, если их не взять в скобки с набором их переменных. Такой же подход действует при замене эквивалентности на сравнение «равно», которое записывается в виде знака двойного равенства. Это позволяет существенно упростить запись логического выражения, сделав ее менее громоздкой и более читаемой. Кроме того, при решении задания «на бумаге» можно для упрощения записи логического выражения применить и другие законы равносильного преобразования логических формул, такие как инволютивность, коммутативность или законы де-Моргана. Использование же инструментов системы программирования Python этого не требует. Логическое выражение можно не упрощать: среда выполнит вычисления в указанном порядке, не затрачивая дополнительных вычислительных ресурсов и не прибегая к логическим преобразованиям [13]. Таким образом, обучающиеся должны уметь грамотно записывать заданные в условии логические выражения с учетом приоритета выполнения действий, определенных в записи, и упрощать их структуру.

В-третьих, обучающиеся внутри конструкции вложенных циклов должны составить с помощью конструкции оператора ветвления *if* запись простейшего сравнения значения логического выражения с логической константой *False* или *True* в зависимости от условия задания. При этом следует обратить их внимание, что для этого так же более удобно, как и при интерпретации логических функций импликации и эквивалентности, использовать числа 0 и 1 для обозначения значений логических констант. Значение 0 соответствует по условию заданий ситуации, что функция тождественно ложна, а 1 – тождественно истинна. После проверки условия необходимо вывести на экран консоли значения набора переменных, которые соответствуют указанному значению функции, и дожидаться появления результатов выполнения программы, которое происходит за долю миллисекунды. Заметим также, что для удобства перед выполнением циклов можно с помощью той же команды *print()* вывести на экран наимено-

вания переменных, которые участвуют в записи логического выражения, приведенного в задании. Это позволяет отобразить исходное расположение логических переменных по столбцам как отправную точку анализа полученных данных.

В дальнейшем решение задания строится на выборе такого положения строк и столбцов, которое бы соответствовало фрагменту данных значений логических переменных, представленных в условии задания. При этом отметим, что использование языка программирования Python исключает возможную ловушку, отмеченную в задании, состоящую в том, что строки таблицы истинности не должны совпадать. Фрагмент таблицы истинности, полученный после запуска программы, исключает такую ситуацию априори. Рассмотрим авторский пример такого задания, решение которого будет получено с помощью системы программирования Python.

Пример. Задано логическое выражение $F = (a \rightarrow b \wedge \neg c) \vee d$. Определите, какому столбцу приведенного фрагмента истинности логического выражения соответствует каждая из логических переменных a, b, c, d , учитывая, что строки в таблице не повторяются (табл. 1).

Таблица 1

Таблица истинности логического выражения F

?	?	?	?	F
1	0			0
	1		0	0
1			1	0

Примечание: составлено авторами.

Решение. Представим программный код построения фрагмента таблицы истинности, при котором значение логического выражения равно 0, в языке Python.

```
print('a b c d')
for a in range(2):
    for b in range(2):
        for c in range(2):
            for d in range(2):
                f = (a <= (b and not c)) or d
                if f == 0: print(a, b, c, d)
```

После запуска данной программы на экран консоли будет выведена следующая информация, соответствующая расположению переменных a, b, c, d по столбцам.

```
a b c d
1 0 0 0
1 0 1 0
1 1 1 0
```

Остается для получения ответа на вопрос о расположении логических переменных по столбцам проанализировать ее. Заметим, что в таблице есть два столбца, полностью состоящих из одинаковых чисел. Это столбцы переменных a и d . Они полностью состоят из 1 и 0 соответственно. Тогда, если в пустой столбец исходной таблицы истинности расположить переменную a , то для переменной d не останется в ней места, так как в оставшихся столбцах есть по крайней мере одна единица. Ввиду этого следует поместить переменную a в первый столбец, так как во втором и четвертом есть нули, а переменную d расположить в третьем столбце таблицы. Осталось правильно расположить переменные b и c по столбцам. Для этого обратим внимание, что во фрагменте во второй строке есть значения 1 и 0 во втором и четвертом столбце соответственно. А в выведенном программой таблице такой набор есть также во второй строке, но в обратном порядке. Следовательно, второму столбцу будет отвечать переменная c , а четвертому – b . Таким образом, искомая последовательность логических переменных – $acbd$.

Ответ: $acbd$.

Задания такого типа были предложены обучающимся в ходе экспериментальной работы по углубленному изучению математической логики в курсах профильной информатики Смоленского физико-математического лицея ЯВИР при МЭИ и МИФИ и средней школы № 6 г. Смоленска. Задания были разноуровневые. На базовом уровне обучающимся предлагались задания, в которых в выражениях использовались только логические операции отрицания, конъюнкции и дизъюнкции, а значение функции F было для всех строк одинаково – либо 0, либо 1. На повышенном уровне – в выражениях были добавлены операции импликации и эквивалентность. В заданиях высокого уровня сложности число выводимых программой строк фрагмента таблицы истинности было больше, чем количество строк в представленной в задании таблице. При этом, если на базовом и повышенном уровнях сложности порядок действий всегда определялся скобками, то в заданиях на высоком уровне сложности было необходимо следить за приоритетом выполнения логических операций и определяемой в соответствии с этим последовательностью действий. Кроме того, строки выводимой таблицы истинности приходилось при анализе переставлять местами, что также увеличивало сложность проводимого обучающимися анализа.

Такая система заданий определила методологическую основу экспериментальной работы. Результаты, полученные в ходе формирующего и констатирующего экспериментов, были обработаны и проанализированы с помощью математических методов. Гипотеза исследования состояла в том, что использование инструментальных средств языка программирования Python при обучении анализу значений переменных логических выражений в школе повышает эффективность образовательного процесса.

Педагогический эксперимент, в котором принимали участие 32 обучающихся, состоял из двух этапов. На констатирующем этапе эксперимента обучающиеся изучали основные понятия математической логики, принципы построения логических выражений, правила выполнения логических операций, приемы и особенности построения таблиц истинности логических выражений. Они выполняли задания по теме «Таблицы истинности логических выражений», решая их в тетради или используя блокноты для записи на электронных планшетах. Большая часть времени была отведена получению навыков составления таблиц истинности и «ручному» способу анализа данных, содержащихся в них. В конце этапа была проведена диагностическая работа, включающая 3 задания базового уровня сложности, 2 – повышенного и 1 – высокого уровня.

На данном этапе эксперимента из 18 обучающихся СФМЛ ЯВИР при МИФИ и МЭИ высокий уровень усвоения знаний и сформированности умений показали 4 человека, повышенный уровень – 7 человек и базовый уровень – 7 человек, среди обучающихся средней школы № 6 г. Смоленска – 3, 6 и 5 человек соответственно. Такие данные демонстрируют нормальный закон распределения с пологом спуском в зону базового уровня.

На формирующем этапе эксперимента обучающиеся пробовали свои силы при решении заданий анализа таблиц истинности логических выражений с использованием системы программирования Python. Отметим, что базовые навыки алгоритмизации и программирования ими были получены ранее. Они дополнительно изучили правила записи логических операций импликации и эквивалентности в среде Python. При этом особое внимание было уделено приоритету выполнения действий в записи сложных логических выражений. Анализ соответствия полученного программным способом с помощью языка программирования Python фрагмента таблицы истинности и таблицы из задания проходил в его инструментальной среде. После этого об-

учающиеся определяли и записывали полученную последовательность логических переменных. В конце этапа была проведена диагностическая работа, включающая 1 задание базового уровня сложности, 2 – повышенного и 3 – высокого уровня.

Участие в эксперименте на данном этапе принимали те же обучающиеся. При этом в СФМЛ ЯВИР при МИФИ и МЭИ высокий уровень продемонстрировали 9 человек, повышенный – 8 учеников и 1 обучающийся – базовый уровень усвоения знаний и сформированности умений, в средней школе № 6 г. Смоленска – 7, 6 и 1 человек соответственно. Такие данные также отвечают нормальному закону распределения, но при этом уже с пологом подъемом в области высоких результатов. Это свидетельствует о существенном приросте учебных достижений по теме «математическая логика».

В ходе эксперимента для генерации наборов заданий, предлагаемых на занятиях обучающимся, использовалась система «Advanced Tester». В среде этой системы осуществляли статистический анализ диагностических работ, а также выполняли построение индивидуальных и групповых траекторий обучения [14, 15]. При этом отметим, что возможности системы «Advanced Tester» таковы, что обучающиеся могли готовиться к занятиям в удаленном режиме, как и выполнять домашние задания.

Качественный анализ условий и результатов эксперимента. Анализ результатов, полученных в ходе экспериментальной работы, свидетельствует, что у обучающихся произошел качественный скачок в знаниях и умениях. Если на этапе формирующего этапа эксперимента они в большинстве своем уверенно справлялись с заданиями базового и повышенного уровня сложности, то на констатирующем этапе приоритетом стало выполнение заданий высокого уровня сложности. При использовании инструментов программирования на языке Python в совокупности с базовыми знаниями математической логики обучающиеся не теряются при установлении правильной последовательности действий, уверенно оперируют наборами значений переменных логического выражения и эффективно определяют их местоположение в представленной таблице истинности.

Отметим, что использование системы программирования Python при решении данных задач позволяет существенно расширить их круг. Помимо стандартных заданий, обучающимся можно предлагать и задания, содержащие переменные значения 0 и 1 в результирующем столбце значений

логического выражения F . Это не вызывает у них дополнительных затруднений ввиду того, что поиск необходимой комбинации строк и столбцов осуществляется по тем же правилам. При этом такое видоизменение задания предполагает увеличение времени его решения при поиске правильного ответа, так же как и анализ в одном задании двух логических выражений, зависящих от одного набора переменных при заданных фрагментах таблиц истинности. Это позволяет утверждать, что гипотеза исследования эффективности использования системы программирования Python при анализе таблиц истинности и восстановлении исходной последовательности значений переменных логического выражения находит свое доказательство.

Заключение

Таким образом, проведенная в ходе эксперимента работа отражает необходимость более широкого использования системы программирования Python при решении задач анализа значений таблиц истинности логических выражений. Обучающиеся с большим интересом начинают изучать как математическую логику, так и программирование, так как видят применение на практике изученных инструментов в методах исследования ситуаций из взаимопроникающих элементов знаний и умений. Это расширяет их профессиональный кругозор и способствует формированию у них прочной базы компонентов курса информатики, образующих единую систему предметной подготовки.

Список литературы

1. Колобов А. Н. Математическая логика на современном этапе образования // Мир науки, культуры, образования. 2024. №4 (107). С. 236-238. DOI: 10.24412/1991-5497-2024-4107-236-238.
2. Авдеева Е.В. Методические особенности изучения математической логики // Abyss (Вопросы философии, политологии и социальной антропологии). 2019. № 4 (10). С. 64-69.
3. Александров А.А., Гильфанов А.Ф., Беттигер Д.Л. Проектирование адаптивной обучающей среды для изучения операций логики высказываний // Образовательные технологии и общество. 2018. Т. 21, №3. С. 241-248.
4. Корчажкина О.М. Вербально-визуальный метод при обучении булевой алгебре в курсе информатики для старшей школы // Вестник МГПУ. Серия: Информатика и информатизация образования. 2020. № 4 (54). С. 16-26. DOI: 10.25688/2072-9014.2020.54.4.02.
5. Попов В.С., Алефиренко Е.А., Черницына Л.Ю. Способы проверки ответа при решении задания № 2 ЕГЭ по информатике на определение порядка переменных в частично заполненной таблице истинности логической функции // Педагогическая информатика. 2024. №3. С. 72-81.
6. Козлов С.В., Быков А.А. Особенности изучения междисциплинарных тем школьных курсов математики и информатики с помощью методов математического моделирования // Проблемы современного образования. 2021. № 5. С. 250-261. DOI: 10.31862/2218-8711-2021-5-250-261.
7. Козлов С.В., Быков А.А. Обучение анализу значений арифметических выражений с использованием языка программирования Python в школе // Современные наукоемкие технологии. 2024. № 5-1. С. 148-153. DOI: 10.17513/snt.40020.
8. Кондратьева В.А. Обучение основам программирования на языке Python в школьном курсе информатики // Вестник МГПУ. Серия: Информатика и информатизация образования. 2021. № 1 (55). С. 8-16. DOI: 10.25688/2072-9014.2021.55.1.01.
9. Панова И.В., Коливын А.А. Методические аспекты обучения программированию на языке Python в школьном курсе информатики // Информатика в школе. 2020. № 6. С. 47-50. DOI: 10.32517/2221-1993-2020-19-6-47-50.
10. Каракозов С.Д., Маняхина В.Г. Python как базовый язык обучения программированию в школе // Информатика в школе. 2020. № 1 (154). С. 26-30. DOI: 10.32517/2221-1993-2020-19-1-26-30.
11. Моторин В.В. Целесообразность использования языка Python в качестве базового при обучении школьников программированию // Конструктивные педагогические заметки. 2020. № 8-2 (14). С. 118-127.
12. Козлов С.В., Быков А.А. Особенности обучения школьников олимпиадному программированию с использованием методов математического моделирования // Современные наукоемкие технологии. 2022. № 6. С. 141-146. DOI: 10.17513/snt.39215.
13. Маркелов В.К., Завьялова О.А. Язык программирования Python как альтернативный инструмент для решения заданий ЕГЭ по информатике // Информатика в школе. 2023. № 2 (181). С. 63-72. DOI: 10.32517/2221-1993-2023-22-2-63-72.
14. Козлов С.В., Быков А.А. Применение методов математического моделирования для диагностики знаний школьников // Современные наукоемкие технологии. 2021. № 4. С. 157-162. DOI: 10.17513/snt.38632.
15. Киселева О.М. Использование математических методов для формализации элементов образовательного процесса // Концепт. 2013. № 2. С. 51-57.