CB-002

A Study of Value Trace Problems in JavaScript Programming Learning Assistant System

キンテッモン* 舩曳信生* トートーサンディチョー* メイパインパインゾー† イイミャ†
Khin Thet Mon Nobuo Funabiki Htoo Htoo Sandi Kyaw May Paing Paing Zaw Ei Ei Myat

1. Abstract

Nowadays, JavaScript programming becomes very important in developing web application systems. However, it is still not educated in universities, although the study may not be easy for students since the code is usually made by special composing styles. To assist self-study of JavaScript programming, we are developing JavaScript programming learning assistant system (JSPLAS) by extending our JPLAS works for Java programming. JPLAS offers the value trace problem (VTP) for code reading study of novice students. A VTP instance consists of a source code and a set of questions, where each question asks the value of an important variable or an output message in the code. The correctness of a answer is marked through string matching with the correct one. In this paper, we study VTP in JSPLAS for novice students. We generate 57 instances using source codes on basic grammar concepts and confirm the effectiveness through applications to 45 university students in Myanmar and Japan.

Keywords - JavaScript Programming, value trace problem, coding reading, grammar concepts.

2. Introduction

Currently, *JavaScript* is the most popular programming language in the world [1]. *JavaScript* has been used in a variety of mobile/desktop application developments including game applications. *JavaScript programming* has remarkable features compared with other major programming languages. 1) It is easy to get started with by abstracting away most of the complex details of programming and to focus on learning how to program. 2) It allows coding and testing on a Web browser without setting up any development environment. 3) It becomes the essential web technology along with HTML and CSS to build interactive websites. 4) It has many prewritten functionalities to help programmers develop complex systems easily.

In web application systems, *JavaScript* programs are used to detect the browser type used by a person and to customize the webpages depending on it. They are also used for security password creations, check forms, interactive games, and special effects. As a result, *JavaScript programming* has become common in building mobile applications and creating server-based applications.

Under the above-mentioned situations, a lot of people have now started to interest in learning *JavaScript programming*. Their learning styles will depend on their environments as well as their needs. For university students, the high-quality learning tools of *JavaScript programming* have been highly demanded, especially for self-study at home since many students have no opportunities of taking the courses at schools. Unfortunately, most universities are still now teaching *JavaScript* programming in a part of a web-programming course.

Heretofore, we have developed the *Java programming learning assistant system (JPLAS)* to assist self-study of *Java programming* [2]. *Java* is widely used in IT societies as the practical programming language. To help *Java programming* study at various levels, JPLAS provides several types of exercise problems, such as the *grammar-concept understanding problem (GUP)*, the *value trace problem (VTP)*, the *element fill-in-blank problem (EFP)* and the *code writing problem (CWP)*. For any type of the problems in JPLAS, the answer from a student is marked automatically, to support programming self-studies by the students. Therefore, we have started the project of developing *JavaScript programming learning assistant system (JSPLAS)* by extending the works for JPLAS.

In JPLAS, the value trace problem (VTP) is designed for novice students to study basic grammar concepts and programming skills through code reading [3]. A VTP instance consists of a source code, a set of questions, and their correct answer strings. Each question requests to answer the value of an important variable or an output message in the source code. The correctness of any student answer is marked through string matching with the stored correct string.

In this paper, we study the value trace problem (*VTP*) in JSPLAS, based on the works for JPLAS. To study basic grammar concepts and code implementations of JavaScript programming, we collect 57 source codes containing basic grammar concepts. Then, we analyze the important variables, add their standard outputs in the codes, and prepare the correct answers, manually. After that, we generate the corresponding VTP instances by applying the program for generating the answer interface.

For evaluations, we assign the generated 57 VTP instances to students in Myanmar and Japan who are studying JavaScript programming, and confirm the effectiveness of the proposal in studying basic JavaScript programming. Here, we divide the instances in two groups, namely, 32 in VTP-I and 25 in VTP- II, to help the students solve them step-by-step.

The rest of this paper is organized as follows: Section 3 reviews our preliminary works to this study. Section 4 presents value trace problems (VTPs) for JavaScript programming. Section 5 evaluates them through applications to students. Finally, Section 7 concludes this paper with future work.

3. Preliminary Works

In this section, we review our preliminary works to the value trace problem (*VTP*) [4].

3.1 Concept of the Value Trace Problems

The design goals of the VTP are as follows:

1) a variety of source codes effective in programming study are depicted with full forms to novice students,

*岡山大学 Okayama University, Japan
†タンリン工科大学 Thanlyin Technology University, Myanmar

2) students can correctly answer the questions only by carefully reading and understanding the codes, and

3) any student answer can be marked through string matching automatically.

Table I. VTP instances for VTP-I

ID	Basic Grammar Concept	LOC	# of forms	avg correct rate
1	Data Types	13	5	98%
2	Arithmetic Operators	32	11	48%
3	Assignment Operators	48	7	86%
4	Functions	27	6	95%
5	Objects	22	3	98%
6	Strings	15	5	98%
7	Backslash Characters	25	3	82%
8	String Specifications	28	4	89%
9	String Extraction	39	9	52%
10	String Replace and Trim	34	6	77%
11	String Conversion	41	9	98%
12	Number Methods	42	15	100%
13	Number Methods	29	8	100%
14	Number to String	40	10	57%
15	Number to String	28	8	98%
16	Arrays	55	11	84%
17	Arrays Association	29	4	91%
18	Basic Arrays Methods	36	8	89%
19	Basic Arrays Methods	29	6	89%
20	Arrays Changing	41	7	89%
21	Arrays Splicing	50	9	98%
22	Arrays Splicing	48	12	98%
23	Arrays Merging	39	8	91%
24	Arrays Sorting	46	8	93%
25	Conditions (if)	11	1	100%
26	Conditions (if, else)	13	2	100%
27	Conditions (if, elseif, else)	15	3	100%
28	Conditions (switch)	58	3	98%
29	Looping (for)	13	11	100%
30	Looping (do while)	9	5	100%
31	Looping (while)	9	5	100%
32	Looping (while)	9	3	100%
Average		30.4	6.7	90.27%
Total (SD)		973	215	13.87%

To compose a VTP instance, a source code, a set of questions with the answer forms (blanks), and the correct answers are necessary. A question asks the messages or the values from the code to the standard output. Thus, to make an efficient VTP, the corresponding standard output statements should be added into the original source code, in order to display the important messages or the values of the key variables in the source code. Unfortunately, the selection of important messages or key variables are manually selected in our works.

3.2 Concept of the Value Trace Problems

A VTP instance can be generated by the following procedure:

1) to select a source code that is suitable for studying basic grammar concepts or a fundamental data structure/algorithm,

2) to find important messages and key variables in the code,

3) to add the corresponding standard output statements of them,

4) to prepare the questions of asking the messages/values at the standard output and their correct answers,

5) to put together the source code, the questions, and the correct answers into one text file,

6) to run the program with the text file to generate the HTML/CSS/JavaScript files for the answer interface on a browser, and

7) to register the generated VTP instance in the assignment to students.

4. Value Trace Problems in JSPLAS

In this section, we present the 57 VTP instances for studying basic grammar concepts of JavaScript programming in JSPLAS.

4.1 VTP Instances for Basic Grammar Concepts

As we discussed before, we divide them into two groups, 32 for VTP-I and 25 for VTP-II. Tables I and II show the grammar concept, the number of lines (LOC) in the source code, the number of answer forms to be filled in, and the average correct answer rate of the students for VTP-I and VTP-II respectively.

Table II. VTP instances for VTP-II

ID	Basic Grammar Concept	LOC	# of forms	avg correct rate
1	Data Types (I)	33	4	100%
2	Data Types (II)	16	2	95%
3	Comparison Operators	51	12	95%
4	Logical Operators	38	3	95%
5	Conditional Operators	11	1	95%
6	Precedence Associativity	20	4	100%
7	Upper Case Check	24	2	70%
8	Vowel Checking	23	1	100%
9	Prime Number Check	34	3	98%
10	Data Types Check	30	3	91%
11	Perfect Number Check	28	2	91%
12	Global Functions (Eval())	15	2	100%
13	Global Functions (isFinite())	17	2	95%
14	Global Functions (isNaN())	37	12	93%
15	GlobalFunctions(parseFloat())	20	7	93%
16	Global Functions (parseInt())	31	12	95%
17	Global Functions (Number())	24	9	91%
18	Numeric Array Sorting	46	3	95%
19	Dates	33	5	98%
20	Var VS Let – Block Scope	18	6	98%
21	Var VS Let – Redeclaration	17	2	98%
22	Var VS Let – Looping	16	2	98%
23	Variable Declaration (const)	25	5	93%
24	Var VS Const – Block Scope	11	3	98%
25	Arrow Functions	18	3	95%
Average		25.4	4.36	94.91%
Total (SD)		636	110	5.73%

4.2 Example of VTP Instances

Here, we show an example of the generated VTPs in this study. Source code shows the adopted **source code** for studying JavaScript Data Type Usage in instance ID=1 of the basic grammar of VTP Part I. **Question** shows the corresponding question with five answer forms. The correct answers to them are

undefined, null ,100, true and Hello. A student needs to read the source code carefully to understand it and fill in the forms correctly.

Source Code

```
main( ) {
    var undeclaredVar;
    var obj = null;
    var num = 100;
    var inProgress = true;
    var Greeting = "Hello";

    console.log(undeclaredVar); // undefined
    console.log(obj); // null
    console.log(num); // number
    console.log(inProgress); // boolean
    console.log(Greeting);   // string
}
```

Questions

What is the value of undeclaredVar? _1_
What is the value of obj? _2_
What is the value of num (approximately two decimal place)? _3_
What is the value of inProgress? _4_
What is the value of Greeting? _5_

Figure 1 illustrates the user interface for this VTP instance. After filling all answer forms, the student has to click the blue "Answer" button. If the answer is not correct, the background color of the form will change into red. Otherwise, the background color will be white. The student can submit the answers repeatedly until all the answers to be correct.

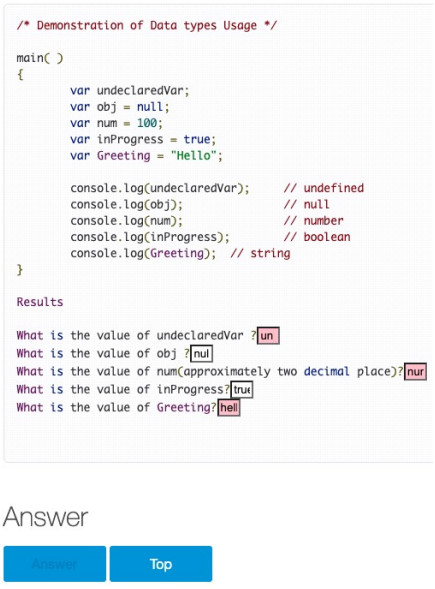

Fig. 1: VTP interface in JPLAS.

5. Evaluation

In this section, we evaluate the generated 57 VTP instances through applications to 45 university students in Myanmar and Japan. Among them, 11 third-year undergraduate students in Myanmar have studied JavaScript programming as one subject in the web programming course. On the other hand, 34 first- year master students in Japan first studied JavaScript programming by reading the provided references for the evaluations before solving them. It is noted that one student did not reply the answers for VTP-I.

5.1 Individual Student Result for VTPs

Figures 2 and 3 show the number of students for each correct answer rate range for VTP- I and VTP-II, respectively. For VTP-I, 6 students among 44 solved with 100% correct rate. There are 25 students who achieved 95 to 99% correct rate. Among the remaining 13 students, 8 did 90 to 94%, 5 did from 80 to 89%. For VTP-II, 21 students among 45 achieved 100% correct rate. 20 students achieved 95 to 99% correct rate, 2 students did 90 to 94%, 2 students did 80 to 89%. The average correct rate for VTP-I is 96.55% and for VTP-II is 98.30%. The smallest rate for VTP-I is 81% and that for VTP-II is 83%. Thus, the generated VTP instances are at proper levels for students to start studying JavaScript programming.

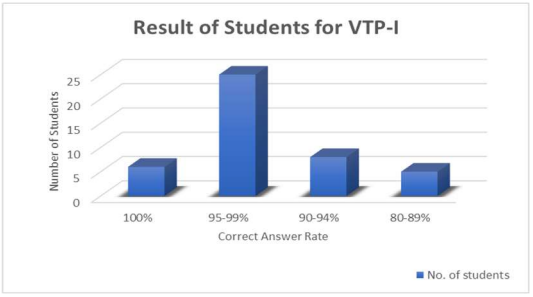

Fig 2. Student results for VTP-I.

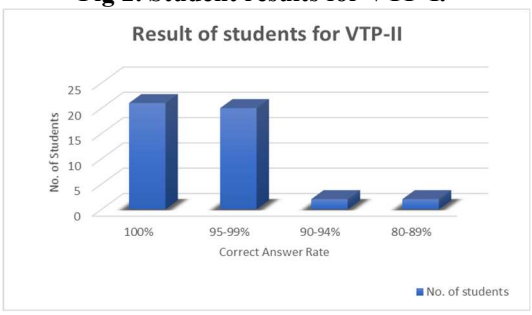

Fig 3. Student results for VTP-II.

5.2 Individual Instance Results

Tables I and II include the average correct answer rate by the students for each VTP instance in VTP-I and VTP-II, respectively. Most of the instances are solved well by the students. However, in some instances, the correct answer rate does not reach 90%. We will discuss these hard VTP instances.

5.2.1 Hard Instances in VTP-I

In VTP-I, the four instances with ID = 2, 9, 10 and 14 show the very low rates that are smaller than 80%, and the six instances with ID = 3, 7, 8, 18, 19, and 20 also give the low rates that are smaller than 90%.

The instance with ID = 2 is on arithmetic operator. The questions ask the values of 11 variables after applying operators. It seems that many students cannot calculate the values correctly,

although they understand the behaviors of the operators. The instance with ID = 3 is on assignment operator. The questions ask the values of seven variables after applying operators. Again, it seems that many students cannot calculate the values correctly. The instance with ID = 7 is on back slash character. The special characters "\", "`", and "}" have specific roles as the commands in a web page. To avoid the role, "}" must be inserted before the character. The four instances with ID = 8, 9, 10, and 14 are on String. The source codes use several library functions in this class such as slice, substr, replace, trim, toExponential, and toFixed. The three instances with ID = 18, 19, and 20 are on Array. The source codes use several library functions in this class such as length, push, pop, and shift.

5.2.2 Hard Instances in VTP-II

In VTP-II, the instance with ID=7 is on uppercase check and has the low rate of 70%. The source code determines where the first character of each string is uppercase or not using a regular expression. It seems that many students may not understand the regular expression.

5.3 Discussion

The average correct answer rate among the students is 96.55% for VTP-I and 98.30% for VTP-II. Thus, it is confirmed that the VTP instances in this paper are proper for JavaScript programming novice students to study. However, there exist several hard instances that resulted in low correct answer rates. To increase the correct rates to them, we will improve the references on these topics in JavaScript programming learning assistant system (JSPLAS), which will be in future works.

6. Related Works in Literature

In this section, we discuss related works to this study in literature.

In [5], G`omez-Albarr`an reviewed several tools to support teaching and learning of programming, and categorized them into four groups, 1) tools including simple reduced development environment, 2) example-based environments,3) tools based on visualization and animation, and 4) simulation environments.

In [6], Brusilovsky et al. presented QuizPACK that asks the value of a particular variable or a string with some fragment of a program. They demonstrate that it significantly improved the knowledge of semantics. Their approach is similar to our works in this paper. We will compare this study.

In [7], Kordaki10 presented a computer-based problem-solving environment named LECGO (Learning Environment for programming using C using Geometrical Objects) for learning C programming by beginners. It emphasized: (a) multiple external representations in student learning, (b) motivation through performing problem-solving activities from the familiar and meaningful context, (c) the active participation of students by using hands-on experience, (d) appropriate feedback to aid self-corrections, and (e) holistic, activity-based, multi-media, multi-representational and multi-layered content for learning basic concepts of C programming.

In [8], Lee et al. presented Gidget, a game where the eponymous robot protagonist is cast as a fallible character that blames itself for not being able to correctly write code to complete its missions. Players learn programming by debugging its problematic code.

In [9], Li et al. presented a game-based learning environment to support novice students learning programming. It exploits game construction tasks to make the elementary programming more intuitive to learn, and comprises concept visualization techniques, to allow students to learn key concepts in programming through game object manipulation.

In [10], Hwang et al. proposed the Web-based programming assisted system for cooperation called WPASC, designing learning activity for facilitating cooperative programming learning, and investigated cooperative programming learning behavior of students and its relationship with learning performance.

7. Conclusion

This paper studied the value trace problem (VTP) for JavaScript programming study by novice students in JSPLAS. 57 VTP instances were generated using various source codes for basic grammar concepts of JavaScript. The application results to 45 university students in Myanmar and Japan confirmed the effectiveness of them, where every student achieved the 80% or higher correct rate on average. In future works, we will improve the references on the hard topics, generate new VTP instances for studying advanced grammar concepts, common libraries, or data structure and algorithms, and apply them to students in JavaScript programming courses.

References

[1] JavaScript is the most popular programming language: Stack Overflow survey, https://fossbytes.com/javascript-most-popular-programming-language
[2] S. L. Ao, et al. ed., IAENG Transactions on Engineering Sciences - Special Issue for the International Association of Engineers Conferences 2016 (Volume II)," pp. 517-530, World Sci. Pub., 2018.
[3] K. K. Zaw, N. Funabiki, and W.-C. Kao, "A proposal of value trace problem for algorithm code reading in Java programming learning assistant system," Inform. Eng. Exp., vol. 1, no. 3, pp. 9-18, Sep. 2015.
[4] X. Lu, N. Funabiki, H. H. S. Kyaw, S. L. Aung, and N. K. Dim, "A study of value trace problems for code reading study of C programming," in Proc. WANC, pp. 445-459, Nov. 2020.
[5] M. G`omez-Albarr`an, "The teaching and learning of programming: a survey of supporting software tools," Comput. J., vol. 48, no. 25, pp. 130-144, 2005.
[6] P. Brusilovsky and S. Sosnovsky, "Individualized exercises for self-assessment of programming knowledge: an evaluation of QuizPACK," ACM J. Edu. Res. Comput., vol. 5, no. 3, pp. 1-22, Sep. 2005.
[7] M. Kordaki, "A drawing and multi-representational computer environment for beginners' learning of programming using C: Design and pilot formative evaluation," Comput. Edu. vol. 54, pp. 69{87, 2010.
[8] M. J. Lee and A. J. Ko, "Personifying programming tool feedback improves novice programmers learning," in Proc. ICER, pp. 109-116, Aug. 2011.
[9] F. W. B. Li and C. Watson, "Game-based concept visualization for learning programming," in Proc. MTDL, pp. 37-42, Dec. 2011.
[10] W.-Y. Hwang, R. Shadiev, C.-Y. Wang, and Z.-H. Huang, "A pilot study of cooperative programming learning behavior and its relationship with students' learning performance," Comput. Edu. vol. 58, pp. 1267{1281, 2012.