Error Detection Of HTML Code Using Pushdown Automata (PDA)

Thida Aung, Thi Thi Soe Nyunt
University of Computer Studies, Yangon
aung.demon85@gmail.com

Abstract

Pushdown Automata (PDA) is one of the hierarchies of automata theory models appropriate for the design of compiler. This system implemented to detect the compile errors of the HTML program when basic beginners who write the HTML program in Notepad run this program in web browser such as Microsoft Internet Explorer, Netscape, Mozilla Firefox and so on. Now, basic beginners don't know their errors after compiling HTML program. This thesis detects the compile errors by using the Pushdown Automata (PDA). The Pushdown Automata (PDA) uses the stack which can pop and push off the information. The advent of such system will provide the beginners with a guideline through which the beginners can see their errors of the HTML program. This system intends the basic beginners for teaching aids.

Keywords: HTML (HyperText Makeup Language)

1. Introduction

Nowadays, the knowledge of web plays an important role among many technologies. Since there is creating a web site in web design process we need to understand the building of HTML page. An HTML file is a text file containing small markup tags. The markup tags tell the web browser how to display the page. An HTML file can be created using a simple text editor. When an HTML file was saved, the .htm or .html extension can be used. HTML files can easily be edit using WYSIWYG (What You See Is What You Get), Macromedia Dreamweaver, Adobe Golive and Microsoft FrontPage, instead of writing the markup tags in plain text file. However to be a skillful web developer, a plain text editor was used to learn our primer HTML. This system intended the beginner who is not use the HTML editor. The user has no experience with HTML programming knowledge. When they run the HTML program, they can face many errors such as missing tags, lasting one of the container tags as well as syntax errors. The beginners don't know such errors. If the beginners are told that what errors are occurred and where the errors are existed in HTML program, the beginners will detect the errors and edit the program quickly. So the beginners will save time and they have no more works to find errors themselves. The system serves to know the position of the errors as soon as they compile the HTML program.

2. Definition of Pushdown Automata

Pushdown automata (PDAs) are acceptors for context-free languages. These automata differ from finite automata in that they use memory in the form of a stack. An example of a PDA for the Language $L=\{a^n b^n : n>=0\}$ is illustrated in Figure 1.

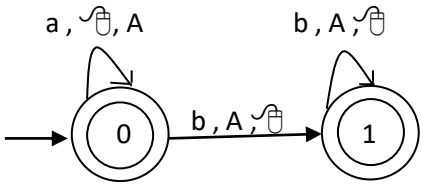

Figure 1 : An Example of a PDA

The start state is indicated by an arrow and a circle and an accept state is represented by a double circle. Note that both the states in Figure 1 are accepting states. Each transition is a 3-tuple specifying and element of the input alphabet, the element to pop off the stack and an element to be pushed onto the stack. The input string alphabet is specified in lowercase and the stack alphabet in uppercase. Note that a transition cannot be performed if the element to be popped off the stack is not currently at the top of the stack. If nothing is pushed onto the stack or popped off the stack this is indicated

by λ . A lambda is also used to represent a λ -transition, i.e. a transition which does not read in a character but moves to another state. A λ -transition can result in an element being pushed onto and/or popped from the stack. In such a transition the input symbol is λ . A PDA can be formally defined as follows:

$N= \{q_0, \Sigma, \Gamma, Q, F, \delta\}$

where

q is the start state

Σ is the input string alphabet

Γ is the stack alphabet

Q is the set of states

F is the set of accept or final states

δ is the set of defining the transitions

Pushdown automata (PDAs) can be deterministic (DPDA) or nondeterministic (NPDA). In an NPDA there can be more than one processing option when read in a particular input symbol, given the current status of the stack, i.e. there exists more than one path through an NPDA for an input string. However, in a DPDA there exists only one unique path for processing an input string. An input string is accepted by a PDA If upon reading in the entire string the PDA halts in an accept state the stack is empty. In the case of an NPDA there must be at least one path in the NPDA for which this holds.

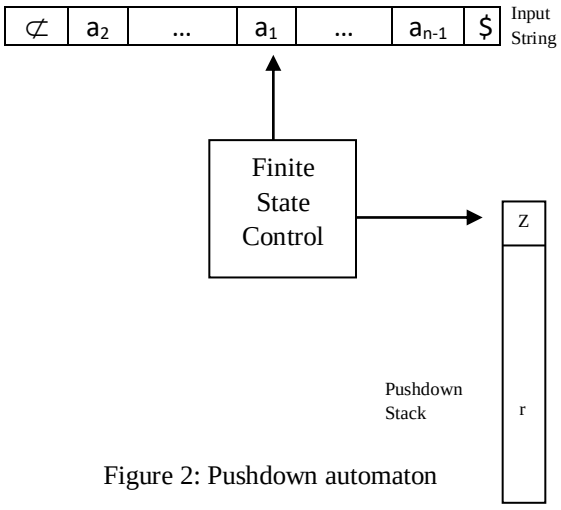

Figure 2: Pushdown automaton

In the Figure 2, $\subset$ and $ are the left endmarker and right end marker respectively. a_1, a_2, …, a_{n-1}, a_n is the input string. Z is the top stack symbol and r is the stack symbols in the stack. Figure 2 shows the operation of the pushdown automata with stack and control state.

A pushdown automaton detects that its input sequence is a nonsentence the first time it encounters a REJECT entry in its control table. At this point the compiler is accepted to produce an error message describing what mistake the programmer might have made. Furthermore, the compiler is excepted to somehow modify the configuration of the pushdown machine so that the processing of the input sequence can be continued and if appropriate, additional message can be produced. This modification method is error recovery.

To produce an appropriate error message, the compiler can analyze the input string and stack contents at the time rejection. Frequently an extensive analysis is not made and the error message is based only on the top stack symbol and the current input. The top stack symbol says what the compiler was looking for, and the input says what it found.

The transitions of the pushdown automata are denoted using the words PUSH, POP, ADVANCE, RETAIN, STATE in the following manners: POP means pop the stack symbol. PUSH (A) where A is a stack symbol means push the symbol A on the Stack. STATE(s) where s is a state s is to be used as the next state. ADVANCE means that the next input is to be used as the current input. In some implementations, this might “advancing” the input pointer. RETAIN means that the current input is to be retained.

3. Proposed System Overview

This system can provide to detect the errors of the HTML program for the beginners. In this system, the beginners write the HTML program in Notepad. Then the beginner run HTML program in the browser. If there are errors in HTML program, the browser cannot show the syntax error. So the beginners does not see their syntax errors in the

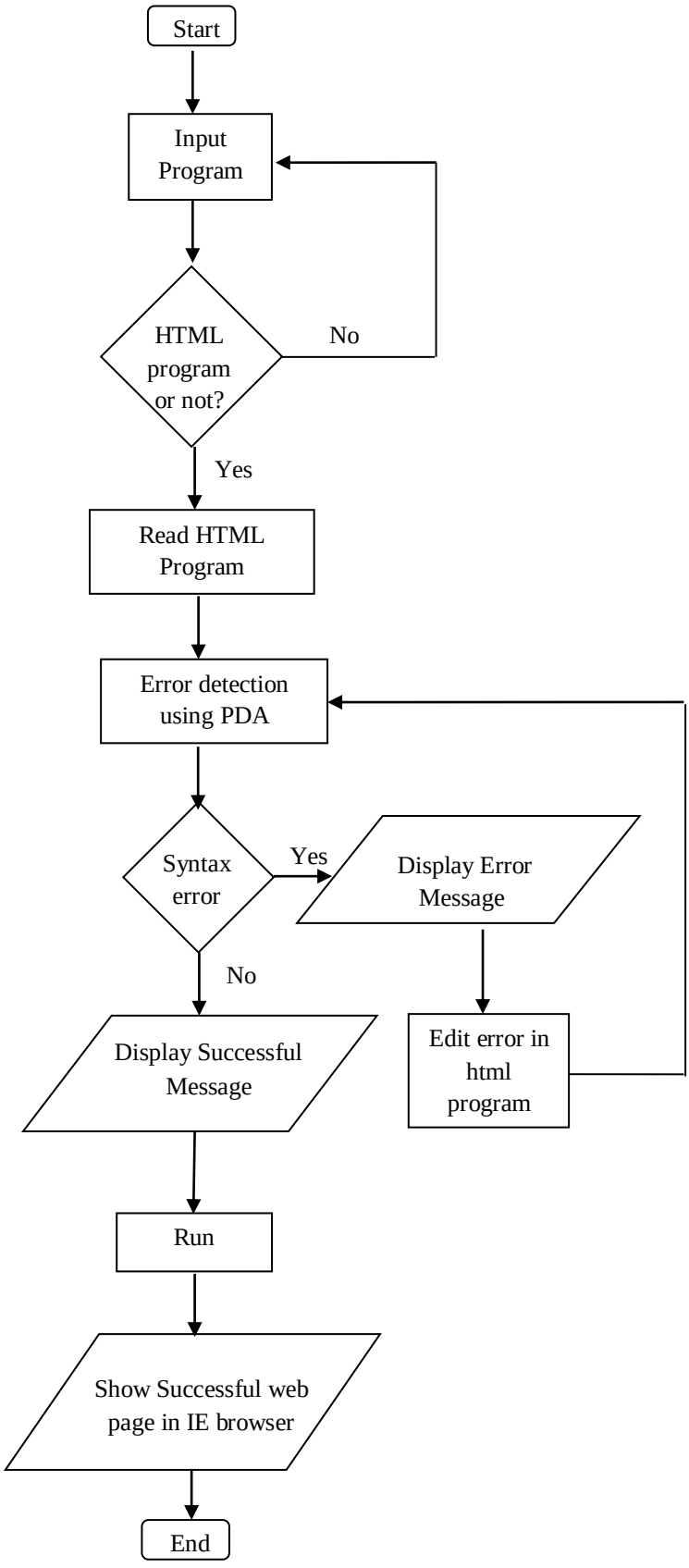

Figure 3: Proposed System Overview

HTML program. The system intended to detect their syntax errors in HTML program. If the errors are found, the system gives the appropriate error message to the users. The user easily sees and edits their syntax error. The system is used to help the basic beginners in teaching HTML program.

When the beginner input the HTML code to the system, the system reads the HTML program line by line. The system extracts all HTML tags and internally operates. In the error detection process, the Pushdown Automata (PDA) technique was used to detect the errors of the HTML program. When the error was found, the system shows the error message in the compile message area. The beginners easily can see the error message with line number. When the user inputs the HTML program, the system provides to detect the position of errors. The system produces an appropriate error message describing what mistake the user might have made. The system helps the beginners without delay in finding errors and they quickly perform the error recovery process. By using this system, the beginners can see where the errors occurred and edit this error immediately. The system operates the compiling process, the error detection process, the edit process and the run process.

3.1 Compiling Process

When the beginner input the HTML program, the system extracts all tags that exist in HTML program with predefined regular expression**.** After passing through regular expression, the system splits the tag without "<" and ">". If the tag has attributes, the system again split the tag string without attributes. The system finds the line number of the HTML program to push the stack. After this stage, the system classifies the standalone tag, the open tag and the close tag. The system has the stand lone tag array. This array can check what the stand lone tag is. If the "/" exists at the first position of the tag string, the system accepts the incoming tag is the close tag. Unless the "/" exists, the tag is open tag.

The system performs three operations.

(1) For the standalone tag, the system pushes the standalone tag with line number and pops the standalone tag from the stack immediately.

(2) For the open tag, the system pushes the open tag with line number to the stack.
(3) For the close tag, the system pops the top stack symbol from the stack. Error detection process operates at this stage.

3.2 Error Detection Process

Error detection process operates by the use of Pushdown Automata (PDA). A pushdown automaton has three information. They are the control state, the current input symbol and top stack symbol. If the incoming tag is the close tag, the system pops the top stack tag. The top stack tag has line number. After popping, the system splits the tag and line number. Then the pop tag compares with the incoming close tag. If they are match, there is no error. Unless they are match, the system displays the error messages with line number in the compile message area. When the beginner input the tag that is not HTML tag, the system shows the error messages. The beginner can easily see the syntax errors of the HTML code and they can easily edit the program.

3.3 Edit process

The beginner can see the error messages in the compile message area. They can know where the error occurs and what the error is. When the beginner clicks the line number double in compile message area, the line in HTML program will be selected. So they can easily edit the syntax errors error of the HTML program. After edit process, they can compile again. When there is no error, the system gives the message "The process is successfully completed". If the error exists, the system shows the error messages again.

3.4 Run process

When there is no error in HTML program, the beginner can run the program. If the errors exist, the system does not allow running. When the beginner run the program without errors, the system shows the successful web page in the browser.

There is a sample code that is as follow:

```
1  <html>
2  <body>
3  <form name="input" action="html1.html"
       method="get">
4  Type your first name:
5  <input type = "text" name = "FirstName" value =
   "Mickey" size = "20">
6  <br>Type your last name:
7  <input type = "text" name = "LastName" value =
   "Mouse" size = "20">
8  <br>
9  <input type = "submit" value = "Submit">
10 <p>
11 </form>
12 If you click the "Submit" button, you will send
13 your input to a new page called html1.html.
14 </p>
```

Compile Message:

Line Number: 11

Error Message : </form> is not match with

Line Number: 2

Error Message : is not match with <body>

Line Number 1

Error Message: The tag <html> must have its close tag.

4. Conclusion

In the system, the Pushdown Automata (PDA) technique was detected the errors of the HTML program. When the user inputs the HTML program, the system provides to detect the position of errors. The system produces an appropriate error message describing what mistake the user might have made. The system helps the basic beginners without delay

in finding errors and they quickly perform the error recovery process. By using this system, the beginners can see where the errors occurred and edit this error immediately. The system provides the basic beginners in teaching aids.

5. References

[1] C. Britton, J. Doake, "Object Oriented Systems Development: A gentle introduction", McGraw-Hill, 2000, 0-07-118869-X.

[2] E. L. Castro, "HTML for the World Wide Web (5^{th} Edition) with XHTML and CSS", Peachpit Press, 2003, 0-321-13007-3.

[3] P.M. Lewis, D.J. Rosenkrantz, R.E. Steams, "Compiler Design Theory", Addison-Wesley, 1976, 0-201-14455-7.

[4] H. Robin, "The Essence of Compiler", Prentice Hall Europe, 1999, 0-13-727835-7.

[5] A.B. Tucker, R.E Noonan, "Programming Languages: Principles and Paradigms", McGraw-Hill, 2002, 0-07-112280.

[6] Taung Win, "Learning Web Design: A Beginner's Guide to HTML Graphics, and Beyond", Myanmar Heritage, 2007.

[7] A.V. Aho, J. E. Hopcroft, J. D. Ullman, "A Recognition Algorithm for Pushdown Store Systems", Proceeding_1968 ACM National Conference, pp. 597-604.

[8] N. Amashini, P. Nelishia, "Evolving Pushdown Automata", ACM National Conference, 2007, pp. 83-90.

[9] "Learning HTML", Online Document,

http://www.w3schools.com/html /html_intro.asp.

[10] "Compiler Construction", Online Document,

http://www.wikipedia.org/Books/Compiler_construction.htm.