CASE STUDY

THEORY OF COMPUTATION

Problem statement:

Our sole objective is to develop an HTML parser that can accurately and efficiently extract relevant information from HTML documents. The purpose of this parser is to enable web developers and data analysts to easily access and analyse the content of web pages. The parser should be able to handle various types of HTML documents, including those with complex structures and nested elements. It should also be able to handle errors and exceptions gracefully, providing informative error messages to users.

Justification:

While a pushdown automaton (PDA) and a Turing machine (TM) are typically associated with theoretical computer science, they can also be applied practically in the development of an HTML parser. Here are some reasons why:

- ***Flexibility in Data Processing:*** A PDA, with its ability to store and process data using a stack, provides flexibility in handling complex HTML parsing operations. The stack allows for efficient storage and retrieval of HTML tags and attributes, facilitating the parsing of HTML documents.

- Computational Power: By emulating a Turing machine, the system benefits from the computational power and versatility of Turing machine operations. This enables advanced data processing, complex parsing algorithms, and the ability to handle diverse HTML scenarios.

- ***Modularity and Scalability:*** The design of the system based on a PDA and emulating a Turing machine allows for modularity and scalability. Additional functionalities and complex parsing rules can be easily added by extending the PDA's state transitions and stack operations, accommodating future requirements and changes in HTML standards.

- ***Standardization and Compatibility:*** Turing machines have been widely studied and used as a theoretical model for computation. By emulating a Turing machine, the HTML parsing system adheres to a well-established computational framework, ensuring compatibility and standardization across different web browsers and systems.

- ***Theoretical Foundation:*** The utilization of a PDA that emulates a Turing machine aligns with the theoretical foundations of computer science and automata theory. It provides a rigorous and formal approach to solving complex problems, ensuring accuracy and reliability in the HTML parsing process.

Overall, the adoption of a pushdown automaton emulating a Turing machine in the development of an HTML parser offers flexibility, computational power, modularity, compatibility, and a solid theoretical foundation, making it a suitable choice for achieving efficient HTML parsing and web development.

GRAMMAR:

S → <tagname>CDE

C →SC| λ |<img>|<br>|<meta>|

D →{Style val unit}| λ

E → </tagname>

Tagname →html|h1|a|p|div|span|body|head|title|script|style

Style→ width:|height:|padding:|margin:

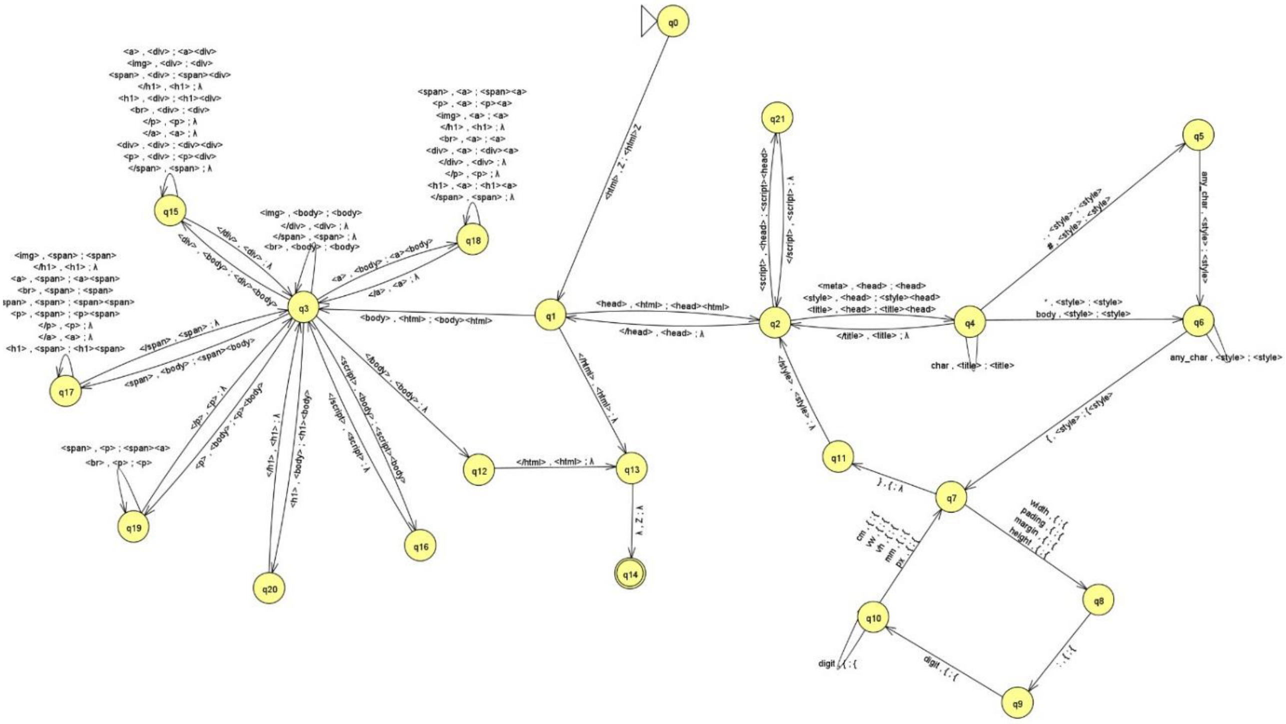
val→(integers)

unit→px|cm|vw|vh|mm

Pushdown Automata:

A pushdown automaton (PDA) is a computational model that incorporates a stack, a finite set of states, input alphabet, stack alphabet, transition rules, and acceptance conditions. It can read input symbols, modify the stack, and transition between states based on the current symbol and the stack. The stack allows for last-in-first-out (LIFO) information management, recognizing and processing context-free languages.

IMPLEMENTATION IN J FLAP:



TRANSITION FUNCTIONS:

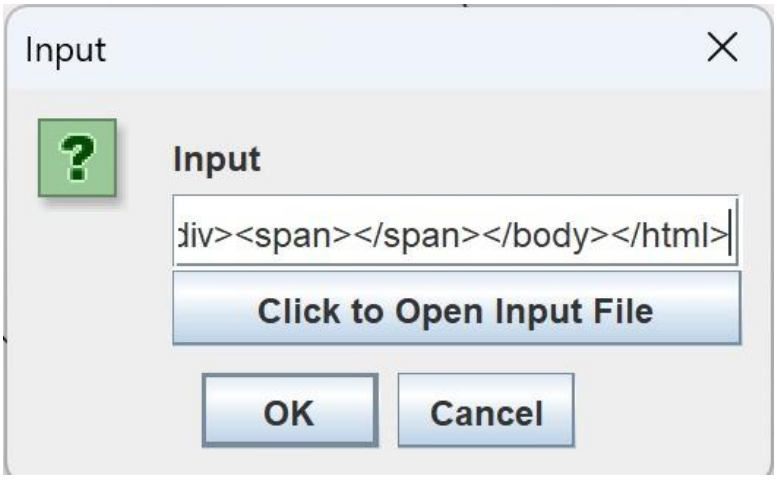
STATE	INPUT	STACK TOP	NEXT STATE	NEW STACK TOP
q 0	<html>	Z	q 1	<html>Z
q 1	<head>	<html>	q 2	<head><html>
q 2	</head>	<head>	q 1	λ
q 1	<html>	<html>	q 13	λ
q 13	λ	Z	q 14	λ
q 12	</html>	<html>	q 13	λ
q 2	<script>	<head>	q 21	<script><head>
q 21	</script>	<script>	q 2	λ
q 2	<style>	<head>	q 4	<style><head>
q 2	<title>	<head>	q 4	<title><head>
q 2	<meta>	<head>	q 4	<head>
q 4	</title>	<title>	q 2	λ
q 4	char	<title>	q 4	<title>
q 4	#	<style>	q 5	<style>
q 4	.	<style>	q 5	<style>
q 4	*	<style>	q 6	<style>
q 4	body	<style>	q 6	<style>
q 5	any_char	<style>	q 6	<style>
q 6	any_char	<style>	q 6	<style>
q 6	{	<style>	q 7	{<style>
q 7	}	{	q 11	λ
q 11	</style>	<style>	q 2	λ
q 7	width	{	q 8	{
q 9	height	{	q 8	{
q 9	padding	{	q 8	{
q 9	margin	{	q 8	{
q 8	}	}	q 9	;
q 9	digit	{	q 10	{
q 10	digit	{	q 10	{
q 10	px	{	q 7	{
q 10	vh	{	q 7	{
q 10	mm	{	q 7	{
q 10	vw	{	q 7	{
q 10	cm	{	q 8	{
q 7	}	{	q 11	λ
q 11	</style>	<style>	q 2	λ
q 1	<body>	<html>	q 3	<body><html>
q 3	</body>	<body>	q 3	λ
q 3	<script>	<body>	q 16	<script><body>
q 3	<h1>	<body>	q 20	<h1><body>
q 16	</script>	<script>	q 3	λ
q 20	</h1>	<h1>	q 3	λ
q 3	<p>	<body>	q 19	<p><body>
q 19	</p>	<p>	q 3	λ
q 3	<span>	<body>	q 17	<span><body>

q 17	</span>	<span>	q 3	λ
q 19	<span>	<p>	q 19	<span><a>
q 19	<br>	<p>	q 19	<p>
q 15	</span>	<span>	q 15	λ
q 15	</a>	<a>	q 15	λ
q 15	</h1>	<h1>	q 15	λ
q 15	</p>	<p>	q 15	λ
q 15	<p>	<div>	q 15	<p><div>
q 15	<a>	<div>	q 15	<a><div>
q 15	<img>	<div>	q 15	<div>
q 15	<br>	<div>	q 15	<div>
q 15	<span>	<div>	q 15	<span><div>
q 15	<h1>	<div>	q 15	<h1><div>
q 15	<div>	<div>	q 15	<div><div>
q 3	<img>	<body>	q3	<body>
q 3	</div>	<div>	q3	λ
q 3	</span>	<span>	q3	λ
q 3	<br>	<body>	q 3	<body>
q 3	<div>	<body>	q 15	<div><body>
q 15	<div>	<div>	q 3	λ
q 3	<a>	<body>	q 18	<a><body>
q 18	</a>	<a>	q 3	λ
q 3	</p>	<p>	q 3	λ
q 3	<p>	<a>	q 3	<p><a>
q 3	</h1>	<h1>	q 3	λ
q 3	</span>	<span>	q 3	λ
q 3	</div>	<div>	q 3	λ
q 3	<span>	<a>	q 3	<span><a>
q 3	<h1>	<a>	q 3	<h1><a>
q 3	<img>	<a>	q 3	<a>
q 3	<br>	<a>	q 3	<a>
q 17	</h1>	<h1>	q 17	λ
q 17	</p>	<p>	q 17	λ
q 17	</a>	<a>	q 17	λ
q 17	<a>	<span>	q 17	<a><span>
q 17	<br>	<span>	q 17	<span>
q 17	<img>	<span>	q 17	<span>
q 17	<span>	<span>	Q 17	<span><span>

q 17	<p>	<span>	q 17	<p><span>
q 17	<h1>	<span>	q 17	<h1><span>

PDA INPUT:

<html><head><script></script><title></title><style>.any_charany_char{width:digitdigitcm}</style></head><body><a></a><div></div><span></span><p></p><h1></h1><script></script><div></div><span></span></body></html>

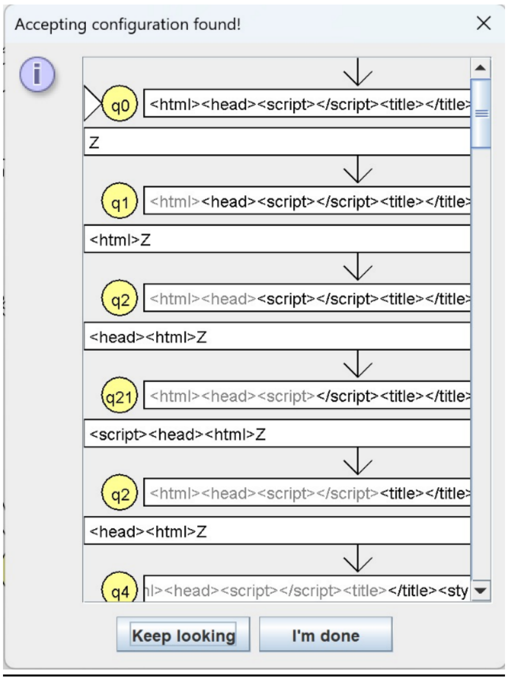


Input

Input

div><span></span></body></html>

PDA OUTPUT:



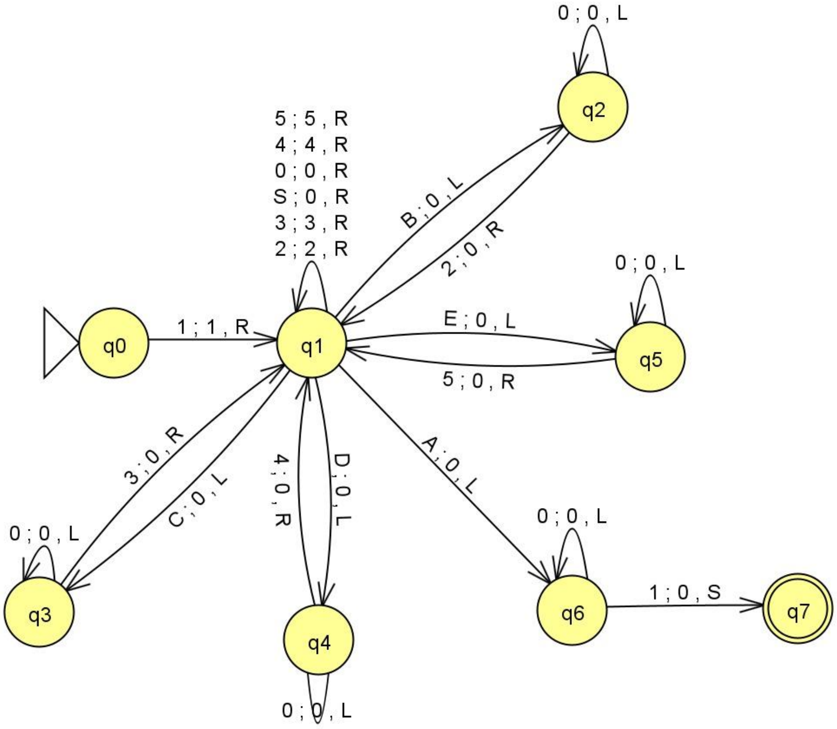
Accepting configuration found!

```
q0   <html><head><script></script><title></title>
Z
q1   <html><head><script></script><title></title>
<html>Z
q2   <html><head><script></script><title></title>
<head><html>Z
q21  <html><head><script></script><title></title>
<script><head><html>Z
q2   <html><head><script></script><title></title>
<head><html>Z
q4   l><head><script></script><title></title><sty
```

Turing Machine:

A Turing machine is a theoretical computational model introduced by Alan Turing. It consists of a tape divided into cells that can store symbols, a head that can read and write symbols on the tape, and a control unit that determines the machine's behaviour based on the current state and the symbol being read. It can perform an infinite number of operations, making it a powerful model for solving algorithmic problems and representing general-purpose computation.

IMPLEMENTATION IN J FLAP:



We have gone for a hierarchical indexing system where tags of lower hierarchy can be nested into parent tags. However the vice versa is not possible

1→ <html>

2 → <head> | <body>

3 → <style> | <script> | <title>

4 → <div> | <p> | <h1> - <h6> | <a>

5 → <span> | {

A → </html>

B → </head> | </body>

C → </style> | </script> | </title>

D → </div> | </p> | </h1> - </h6> | </a>

E → </span> | }

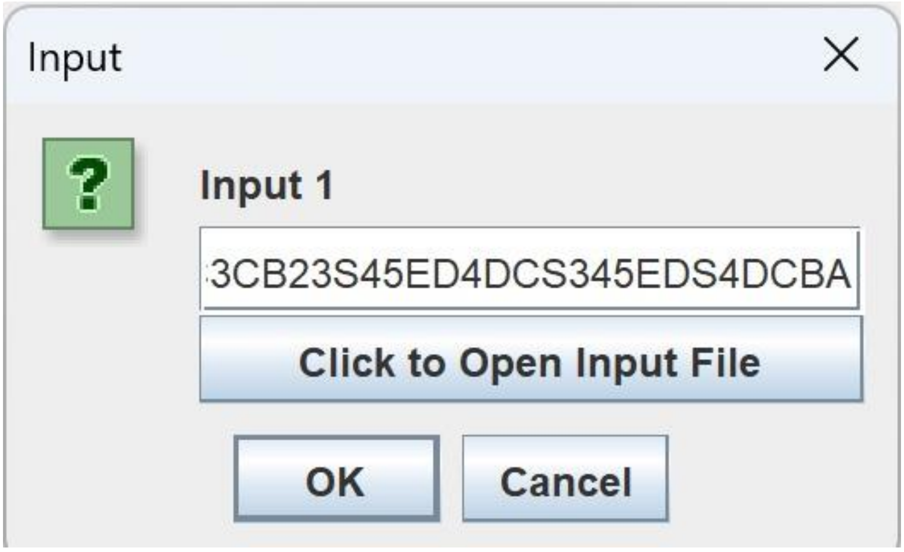
S → <meta> | <img> | <br> | width: | height: | padding: | margin: | num(0 - 9) | # | .

TRANSITIONS:

CURRENT STATE	NEXT STATE	CURRENT TAPE POINTER	REPLACE	DIRECTION
q 0	q 1	1	1	R
q 1	q 1	5	5	R
q 1	q 1	4	4	R
q 1	q 1	0	0	R
q 1	q 1	3	3	R
q 1	q 1	2	2	R
q 1	q 2	B	0	L
q 2	q 1	2	0	R
q 2	q 2	0	0	L
q 1	q 5	E	0	L
q 5	q 1	5	0	R
q 5	q 5	0	0	L
q 1	q 6	A	0	L
q 6	q 6	0	0	L
q 6	q 7	1	0	S
q 1	q 4	D	O	L
q 4	q 1	4	0	R
q 4	q 4	0	0	L
q 1	q 3	C	0	L
q 3	q 1	3	0	R
q 3	q 3	0	0	L

TURING MACHINE INPUT:

123C3CB23S45ED4DCS345EDS4DCBA

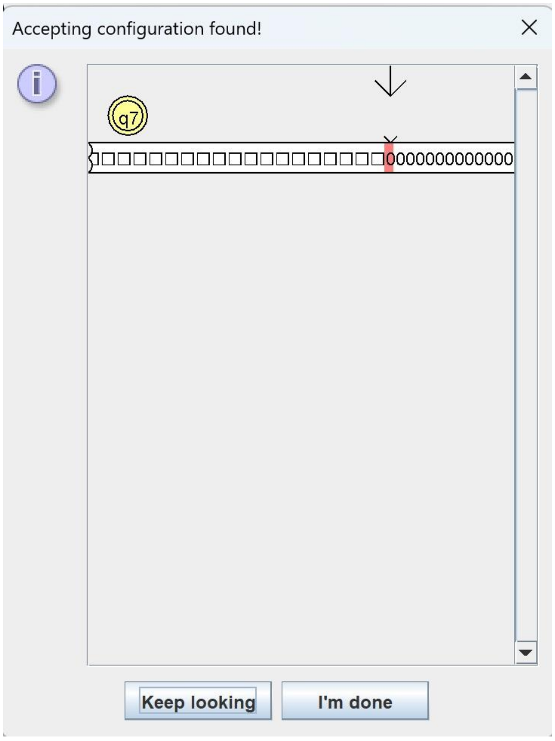


Input

Input 1

3CB23S45ED4DCS345EDS4DCBA

TURING MACHINE OUTPUT:



Accepting configuration found!

CONCLUSION:

The concept of using pushdown automata and Turing machines for HTML parsing provides interesting theoretical perspectives, real-world HTML parsers often employ more specialized techniques and tools to ensure efficient and accurate parsing of HTML documents.

In conclusion we learnt that HTML parsers using PDA and Turing machine are a good choice for developers who need to parse complex HTML documents. They are powerful, easy to implement, and relatively efficient.