STATEMENT OF NEEDS

Overview:

To develop a parser that will parse a perl script and generate Web GUI trees.

Functionalities:

The input to this parser will be a perl file that has html tags embedded within the script.
The goal is to capture dependence relationships between perl code and the html code. There are 2 types of nodes here, A document node and a statement node.

A **Document node** represents the structure of an html statement. Each document node denotes one web object or one attribute of the web object. A composite document node has one or more child nodes, while a primitive node is a leaf node in the WGUI tree indicating a primitive attribute of its parent node. Document nodes are represented as an ellipse in the Figure's below.

A **Statement node** represents the embedded logic in the server page. Each statement node denotes one statement of the embedded code. These statements do not represent any web objects or any of their attributes in the web pages, but they may carry the dependence relationships with web objects or their attributes, which then can be used to derive dependence relationships between web objects or attributes. Statement nodes are represented as circles in the Figure's below.

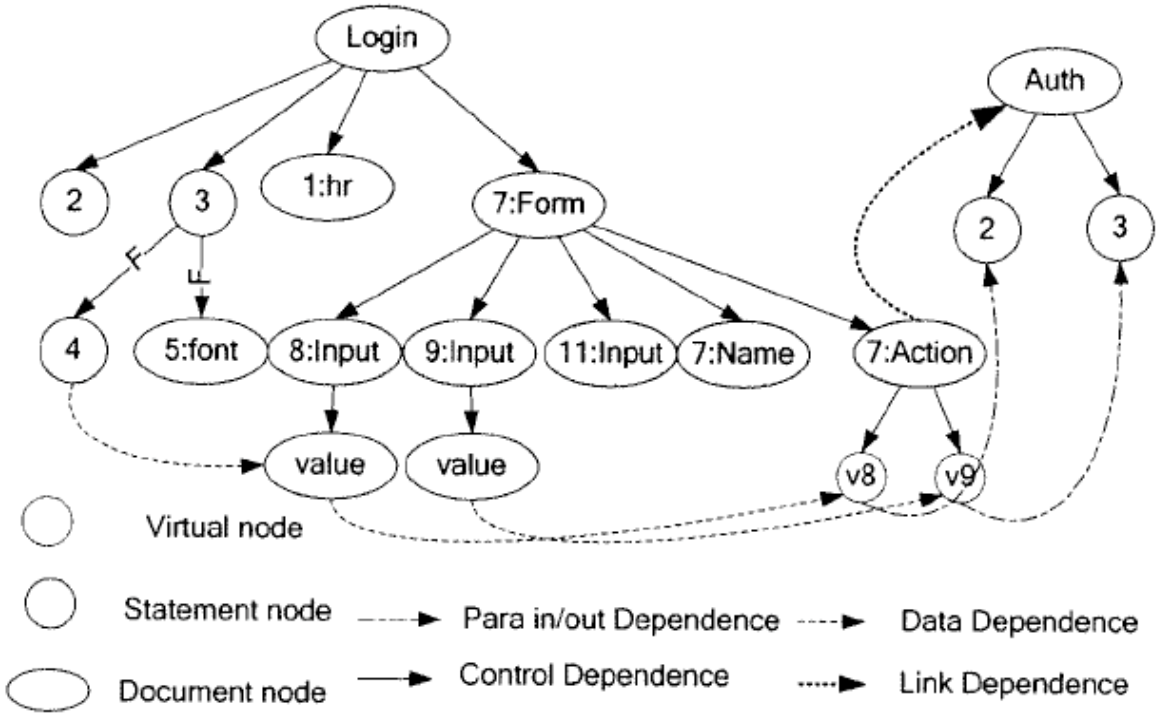
Here is an example of the input file: Login.pl

```
01 <hr><h1>Login</h1><hr>
02 % my $uname;
03 % if ($session) {
04 % $uname = $session;
05 <font color="red">You are currently login as <% $uname %>, you may want to logout first? </font> <br /> <br />
06 % }
07 <form name="login" action="auth.html" method=post align="center">
08 Username: <input type="text" name="uname" value= <% $uname %> ><br /><br />
09 Password: <input type="PASSWORD" name="passwd"><br />
10 <br />
11 <input type="submit" onclick="auth.html" value="login"><br />
12 </form>
```

Figure 1. Example 1: Login.pl.

This input file is written with html code and three Perl statements at lines 2, 3, and 4, respectively.

The output should be in a graphical format as follows. This is a WGUI Tree.



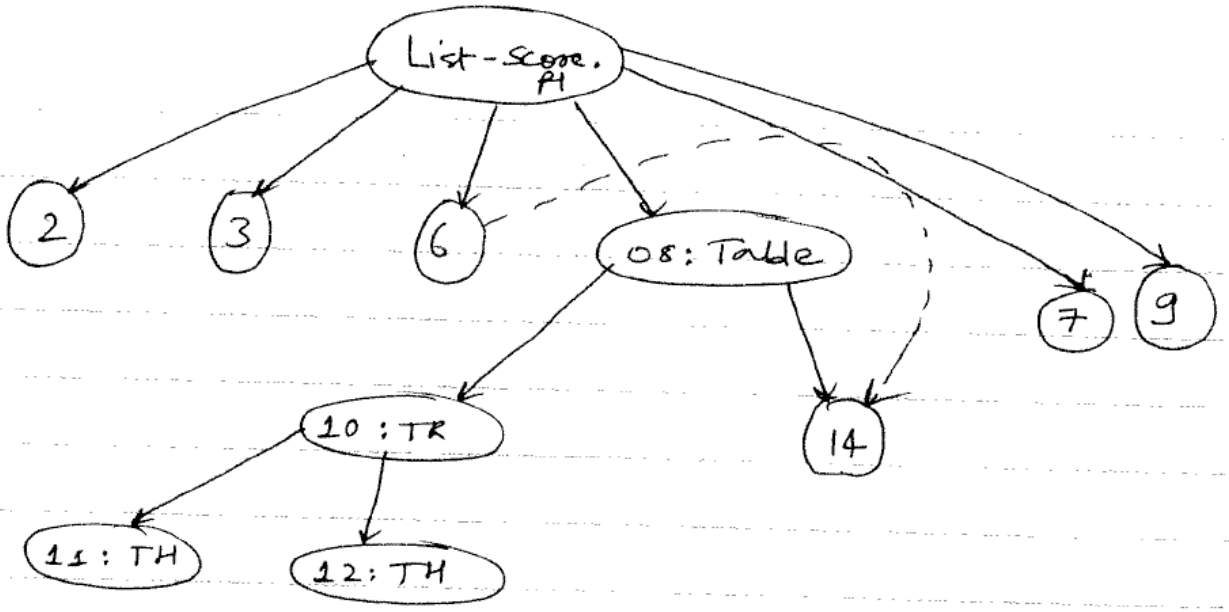
Note that two virtual nodes , v8 and v9 are added in the above Figure to indicate actual-in-nodes associated with the attribute node “Action” that is an analogy of a callsite node.

Here is another example, if the first one is not so clear:

```
01 <%args>
02 $cnum => 'cnum'
03 $term => 'term'
04 </%args>
05 <%perl>
06 my $dbh = DBI->connect('DBI:ODBC:Demo', 'sa', 'password') or die "Couldn't connect to database: " . DBI->errstr;
07 my $sth = $dbh->prepare('SELECT uname, score FROM user_course WHERE course_number = ? and term = ? and type
= \'student\' ') or die "Couldn't prepare statement: " . $dbh->errstr;
08 $sth->execute($cnum, $term);
09 my @harray;
10 while ( my $hash = $sth->fetchrow_hashref() ) {
11         push(@harray, $hash);
12 }
13 return @harray;
14 </%perl>
```

Figure 5. Example 2: list-score.pl.

And the output would again be a WGUI Tree as follows:



A WGUI Tree for Example 2.