Técnologico de Costa Rica

Departamento de Computación

Compiladores e intérpretes

Especificación del proyecto

Anthony Alfaro Sibaja
Iván Felipe Calvo Pérez

Kirstein Gätjens Soto

29 de mayo, 2017

Contents

1 Analizador léxico(Scanner)

El analizador léxico usará un autómata el cuál se cargará en la tabla de transiciones, estese lee en el proyecto como un archivo JSON, contiene 230 estados con los cuales se crea los diferentes estados y se añaden a la tabla.

2 Analizador sintáctico(Parser)

Para esta etapa del compilador, se implementó el driver de parser dado por el profesor, el cuál sigue la lógica del siguiente seudocódigo:

```
Scanner scanner = new Scanner();
int simboloInicial = Gramatica.NO_TERMINAL_INICIAL;
Stack pilaParsing = new Stack();
Token TA = scanner.DemeToken();
int EAP;
int regla;
int i;

pilaParsing.push(simboloInicial);
while(TA.familia != TokenEOF){
  EAP = pilaParsing.pop();
  if(Gramatica.esTerminal(EAP)){
    if(EAP == TA.getFamilia()){
      TA = scanner.DemeToken();
    }
    else
      // Error Sintactico
  }
  else if(Gramatica.esNoTerminal(EAP)){
    regla=Gramatica.TP[EAP-simboloInicial][TA.familia]
    if(regla < 0)
      // Error sintactico
    else{
      i = 0;
      while((Gramatica.TLD[regla][i]>-1) &&                    (i<
    Gramatica.MaxLadoDer)){
        pilaParsing.push(TLD[regla][i]);
        i++;
      }
    }
  }
  else{
    // Aqui va el switch del semantico
    // Esto se explicara en la seccion siguiente

  }
}
if(!pilaParsing.empty()){
```

```
38    // Pila parsing no esta vacia, sobraron simbolos
39    // Error sintactico
40 }
```

3 Analizador semántico

Para esta etapa se incluyó un switch en el driver de parsing el cual posee todos los símbolos semánticos implementados, los cuales ayudan a crear validaciones semánticas del programa, el cual tiene la forma:

```
1 switch(EAP){
2   case Gramatica.idPrograma:
3     ...
4     break;
5 }
```

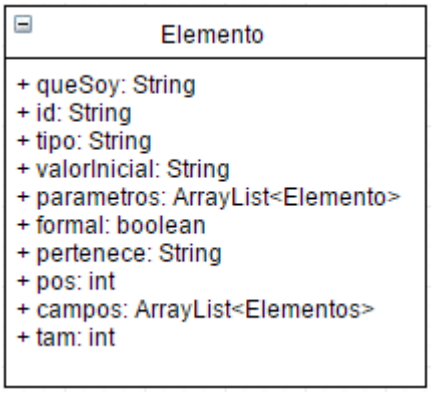
Cada case sería un símbolo semántico identificado por el driver de parsing.

3.1 Tabla Hash

Se implementó el uso de una tabla hash para llevar el control de los símbolos semánticos, en esta se añaden objetos de la clase **Elemento**, la cuál tiene todos los atributos necesarios para poder identificar desde una variable hasta una función.

3.2 Clase Elemento

Cabe destacar que la orientación a objetos no está muy bien implementada:



3.2.1 Atributo 'queSoy'

Este atributo indica qué tipo de elemento es, ya sea una variable, constante, prototipo, función, parámetro, atributo o cualquier otro dato que pueda ser nombrado.

3.2.2 Atributo 'id'

Es el que guarda el nombre de la variable.

3.2.3 Atributo 'tipo'

Es el tipo de dato que almacena, por ejemplo, si es una variable tipo 'Degh', en este atributo se almacenará 'Degh'.

3.2.4 Atributo 'valorInicial'

Es el valor inicial de la variable, por ejemplo si a una variable tipo Mi' le declaran un 5, el 5 se almacenará en ese atributo.

3.2.5 Atributo 'parametros'

Si el valor del atributo tipo es una función o rutina, este tendrpa una lista con los parámetros asignados.

3.2.6 Atributo 'formal'

Si el valor del atributo tipo es 'P'(Parámetro), el atributo formal será un booleano que indica si es o no formal usando true/false.

3.2.7 Atributo 'pertenece'

Si el valor del atributo tipo es el de una parámetro o atributo, tendrá el nombre a la función, rutina o registro al cual pertenece.

3.2.8 Atributo 'pos'

Si el valor del atributo tipo es un parámetro, este tendrá la posición en la cuál fue asignado

3.2.9 Atributo 'campos'

Si el valor del atributo tipo es registro, este tendrá una lista con los campos asignados.

3.2.10 Atributo 'tam'

Si el valor del atributo tipo es un arreglo, este tendrá el tamaño del arreglo.

3.3 Símbolos semánticos

A continuación se mostrarán los símbolos semánticos implementados en la gramática y driver de parsing:

3.3.1 idPrograma

Guarda en la TS un elemento con el nombre del programa, esto para que ese identificador no pueda ser usado después.

3.3.2 idTipos

Verifica que el tipo de variable declarada por el usuario que está instanciando exista previamente,

3.3.3 idFuncion

Verifica si el prototipo de la función que está declarando exista

3.3.4 tipoFuncion

Hace el chequeo de que los parámetros sean iguales, y en el mismo orden del prototipo, también chequea el tipo de valor de retorno, si todo esto se cumple se agrega dicha función a la TS.

3.3.5 idProtoFuncion

Si el id que le dieron al prototipo no está en uso por algún otro, entonces se agrega a la TS un nuevo prototipo

3.3.6 tipoProto

Guarda el una variable auxiliar el id del prototipo actual

3.3.7 idProcedimiento

Verifica que el procedimiento que se declare actualmente tenga su prototipo respectivo

3.3.8 idProtoProcedimiento

Verifica que el nombre del prototipo que se está declarando no este en uso.

3.3.9 reiniciarLista

Reinicia una variable auxiliar

3.3.10 parametroFormal

Le asigna true a una variable auxiliar

3.3.11 tipoParametro

Le asigna el id actual a una variable auxiliar

3.3.12 idParametro

Si es llamado en un prototipo, este le añade los parámetros a este. En otro caso si es en una función o rutina, se añaden en una lista auxiliar para luego ser comparados con los de su prototipo respectivo

3.3.13 reiniciarParametros

Borra todos los elementos de una variable auxiliar

3.3.14 incrementarPos

Incrementa en 1 una variable auxiliar

3.3.15 declaracionUsuario

Crea un nuevo tipo de dato declarado por el usuario si este no se encontraba previamente definido.

3.3.16 setTipoUsuario

Le asigna al tipo de dato declarado por el usuario actual el valor de una variable local

3.3.17 agregarUsuario

Agrega un nuevo elemento a la TS

3.3.18 idAtributo

Si el id del atributo actual no está en una lista auxiliar o si esta es vacía, se agrega

3.3.19 agregaAtributo

Agrega un nuevo elemento a la TS

3.3.20 idConstante

Si el id de la constante actual no está en uso, instancia un nuevo elemento

3.3.21 agregarConstante

Agrega una nueva constante a la TS, si el tipo de dato que se le asignó no concuerda con su tipo, se reporta un warning.

3.3.22 tipoMi

Le cambia el valor a una variable auxiliar a "Mi""

3.3.23 tipoMu

Le cambia el valor a una variable auxiliar a "Mu""

3.3.24 tipoDegh

Le cambia el valor a una variable auxiliar a "Degh"

3.3.25 tipoVit

Le cambia el valor a una variable auxiliar a "Vit"

3.3.26 tipoJaj

Le cambia el valor a una variable auxiliar a "Jaj"

3.3.27 tipoVey

Le cambia el valor a una variable auxiliar a "Vey"

3.3.28 tipoMorgh

Le cambia el valor a una variable auxiliar a "Morgh"

3.3.29 tipoUsuario

Le cambia el valor a una variable auxiliar al tipo actual

3.3.30 setVarRutinas

Modifica el estado de una bandera

3.3.31 sotVarRutinas

Modifica el estado de una bandera

3.3.32 idDeclaracion

Si una variable no existe en el segmento local o global, se añade a la TS

3.3.33 agregarVariableNI

Agrega una variable no inicializada a la TS

3.3.34 agregarVariable

Agrega una variable inicializada a la TS

3.3.35 checkConstantes

Su función es verificar que el bloque de constantes solo sea declarado una vez

3.3.36 checkProtos

Su función es verificar que el bloque de prototipos solo sea declarado una vez

3.3.37 checkRutinas

Su función es verificar que el bloque de rutinas solo sea declarado una vez

3.3.38 checkUsuarios

Su función es verificar que el bloque de tipos declarados por el usuario solo sea declarado una vez

3.3.39 checkProgramaP

Su función es verificar que el bloque principal solo sea declarado una vez

3.4 Librerías adicionales

Se implementó el uso de la librería json.simple para poder leer el archivo JSON de manera más práctica.

4 Gramática

4.1 Terminales

```
id
$$
$+
$?
&!
&+
&++
&–
&—
(
)
*
*=
+
++
+=
,
–
––
–=
/
//
//=
/=
:
:=
<
<<
<=
<>
==
>
>=
>>
@#?
@$?
@<<
@>>
[
]
^
{
}
|+=
|++=
|+++=
|#|
|##|
|###|
' ej
```

```
Be'
Bup
Chen
Choh
Chu'
Chugh
Degh
Duh
Ghap
&-=
Gho'
Ghom
Ghor
Ghu
Ghun
Hegh
Hijol
Hoch
Jaj
Jog
$+=
Latlh
Leq
Lo'
Meh
Mi'
Miw
Mo'
Morgh
Mu'
Nagh
Nawlogh
Ngeb
Ni'
Pigh
Pong
Qa'
Qap
Qo'
Qod
Qonos
Sar
Segh
Suq
Ta
Tadmoh
Tagh
Tah
Teh
Vaj
Vey
```

```
Vis
Vit
<%:
:%>
|<?
|>?
|
literalEntero
literalChar
.
literalCadena
&+=
~
$
```

4.2 Gramática

La gramática de Kaplá a continuación, se encuentra escrita en BNF:

```
<S> ::= Ghun  id  <preprograma-list>  <secciones> ^

<literal> ::= literalCadena
<literal> ::= literalEntero
<literal> ::= <literalBooleana>
<literal> ::= <literalConjunto>
<literal> ::= <literalFecha>
<literal> ::= <literalChar>
<literal> ::= <literalArreglo>
<literal> ::= <literalRegistro>

<literalBooleana> ::= Teh
<literalBooleana> ::= Ngeb

<literalChar> ::= literalChar

<literalConjunto> ::= { <literalConjunto-aux> }

<literalConjunto-aux> ::= <literalConjunto-list>
<literalConjunto-aux> ::=

<literalConjunto-list>  ::= <literalChar> <literalConjunto-list-aux>

<literalConjunto-list-aux>  ::= , <literalConjunto-list>
<literalConjunto-list-aux>  ::=

<literalFecha>  ::= | literalEntero | literalEntero | literalEntero |

<literalArreglo>  ::= [ <literalArreglo-aux>  ]
```

```
30
31 <literalArreglo-aux>  ::= <literal-list>
32 <literalArreglo-aux>  ::=
33
34 <literal-list>  ::= <literal> <literalArreglo-aux-list>
35
36 <literalArreglo-aux-list> ::= , <literal-list>
37 <literalArreglo-aux-list> ::=
38
39 <literalRegistro> ::= <<  <literalArreglo-aux>  >>
40
41 <tipos> ::= Mi'
42 <tipos> ::= Mu'
43 <tipos> ::= Jaj
44 <tipos> ::= Degh
45 <tipos> ::= Vit
46 <tipos> ::= Vey
47 <tipos> ::= Morgh [ <expresion> ] Chen  <tipos>
48 <tipos> ::= id
49
50 <declaracion-rutina> ::= Qap  <declarador>  : <tipos> <bloque-instruccion>  ^
51 <declaracion-rutina>  ::= Miw <declarador>  <bloque-instruccion>  ^
52 <declaracion-prototipo> ::= Qap <declarador>  : <tipos>
53 <declaracion-prototipo> ::= Miw <declarador>
54
55 <declarador>  ::= id  ( <parametros>  )
56
57 <parametros>  ::= <parametros-list>
58 <parametros>  ::=
59
60 <parametros-list> ::= <parametro> ^ <parametros-list-aux>
61
62 <parametros-list-aux> ::= <parametros-list>
63 <parametros-list-aux> ::=
64
65 <parametro> ::= <formal>  <tipos> <id-list>
66
67 <id-list> ::= id  <id-list-aux>
68
69 <id-list-aux> ::= , <id-list>
70 <id-list-aux> ::=
71
72 <formal>  ::= Choh
73 <formal>  ::=
74
75 <rutinas> ::= <rutinas-list>
76 <rutinas> ::=
77
78 <rutinas-list>  ::= <declaracion-rutina>  ^ <rutinas-list-aux>
79
80 <rutinas-list-aux>  ::= <rutinas-list>
```

```
<rutinas-list-aux>  ::=

<prototipos>  ::= <prototipos-list>
<prototipos>  ::=

<prototipos-list> ::= <declaracion-prototipo> ^ <prototipos-list-aux>

<prototipos-list-aux> ::= <prototipos-list>
<prototipos-list-aux> ::=

<declaracion> ::= <tipos> <declaracion-ids> ^

<declaracion-ids> ::= id  <declaracion-ids-aux>

<declaracion-ids-aux> ::= , <declaracion-ids>
<declaracion-ids-aux> ::= Tadmoh  <expresion> <declaracion-ids-aux-1>
<declaracion-ids-aux> ::=

<declaracion-ids-aux-1> ::= , <declaracion-ids>
<declaracion-ids-aux-1> ::=

<declaracion-list>  ::= <list-declaracion>
<declaracion-list>  ::=

<list-declaracion>  ::= <declaracion> <list-declaracion-aux>

<list-declaracion-aux>  ::= <list-declaracion>
<list-declaracion-aux>  ::=

<declaracion-usuario> ::= id  Pong  <declaracion-usuario-aux> ^

<declaracion-usuario-aux> ::= <tipos>
<declaracion-usuario-aux> ::= Ta  <atributos> Hegh

<atributos> ::= <tipos> id  ^ <atributos-aux>

<atributos-aux> ::= <atributos>
<atributos-aux> ::=

<declaracion-usuario-list>  ::= <list-declaracion-usuario>
<declaracion-usuario-list>  ::=

<list-declaracion-usuario>  ::= <declaracion-usuario> <list-declaracion-
    usuario-aux>

<list-declaracion-usuario-aux>  ::= <list-declaracion-usuario>
<list-declaracion-usuario-aux>  ::=

<declaracion-constante> ::= <tipos> id  Tadmoh  <expresion> ^

<declaracion-constante-list>  ::= <list-declaracion-constante>
```

```
<declaracion-constante-list>  ::=

<list-declaracion-constante>  ::= <declaracion-constante> <list-declaracion-
    constante-aux>

<list-declaracion-constante-aux>  ::= <list-declaracion-constante>
<list-declaracion-constante-aux>  ::=

<instruccion> ::= Duh <expresion> : <instruccion> ^
<instruccion> ::= Latlh : <instruccion> ^
<instruccion> ::= Qap Vis <expresion> <instruccion> ^
<instruccion> ::= Be' <instruccion> Qap Qa' <expresion> ^
<instruccion> ::= Hoch  id  Qod <expresion> Be' <instruccion> ^
<instruccion> ::= Leq <expresion> <instruccion> ^
<instruccion> ::= Lo' <expresion> Be' <instruccion> ^
<instruccion> ::= Ghor  ^
<instruccion> ::= Bup ^
<instruccion> ::= Tah ^
<instruccion> ::= Meh <for-aux1>  Mo' <expresion> <for-aux2>  Be' <instruccion
    > ^
<instruccion> ::= Chugh <expresion> <if-aux1> ^
<instruccion> ::= Hijol <return-aux>  ^
<instruccion> ::= <bloque-instruccion>  ^
<instruccion> ::= <expresion> ^

<for-aux1>  ::= id  :=  <expresion>

<for-aux2>  ::= Gho'  <expresion>
<for-aux2>  ::=

<if-aux1> ::= <if-aux2> <instruccion> <if-aux3>
<if-aux1> ::= Qod <expresion> Nawlogh <expresion> Vaj <instruccion>

<if-aux2> ::= Vaj
<if-aux2> ::=

<if-aux3> ::= Latlh <instruccion> ^
<if-aux3> ::=

<return-aux>  ::= <expresion>
<return-aux>  ::=

<bloque-instruccion>  ::= Tagh  <instruccion-list>  Hegh

<instruccion-list>  ::= <instruccion> <instruccion-list>
<instruccion-list>  ::=

<args>  ::= <args-list>
<args>  ::=

<args-list> ::= <expresion> <args-list-aux>
```

```
180
181 <args-list-aux> ::= , <args-list>
182 <args-list-aux> ::=
183
184
185 <expresion> ::= <expresion-asignacion>
186
187 <expresion-asignacion> ::= <expresion-logica> <operador-asignacion>
188
189 <operador-asignacion> ::= <asignacion> <expresion-asignacion>
190 <operador-asignacion> ::=
191
192 <asignacion> ::= :=
193 <asignacion> ::= +=
194 <asignacion> ::= -=
195 <asignacion> ::= *=
196 <asignacion> ::= /=
197 <asignacion> ::= //=
198 <asignacion> ::= |+=
199 <asignacion> ::= |++=
200 <asignacion> ::= |+++=
201 <asignacion> ::= &+=
202 <asignacion> ::= &-=
203 <asignacion> ::= $+=
204
205 <expresion-logica> ::= <expresion-relacional> <operador-logico>
206
207 <operador-logico> ::= <logico> <expresion-logica>
208 <operador-logico> ::=
209
210 <logico> ::= 'ej
211 <logico> ::= Ghap
212 <logico> ::= Jog
213
214 <expresion-relacional> ::= <expresion-trinaria> <operador-relacional>
215
216 <operador-relacional> ::= <relacional> <expresion-relacional>
217 <operador-relacional> ::=
218
219 <relacional> ::= <
220 <relacional> ::= >
221 <relacional> ::= <=
222 <relacional> ::= >=
223 <relacional> ::= ==
224 <relacional> ::= <>
225
226 <expresion-trinaria> ::= <expresion-binaria> <operador-trinario>
227
228 <operador-trinario> ::= <trinario> <expresion-binaria> $$ <expresion-binaria
    >
229 <operador-trinario> ::=
```

```
230
231 <trinario>  ::= <<
232 <trinario>  ::= >>
233
234 <expresion-binaria> ::= <expresion-unaria>  <operador-binario>
235
236 <operador-binario>  ::= <binario> <expresion-binaria>
237 <operador-binario>  ::=
238
239 <binario> ::= +
240 <binario> ::= -
241 <binario> ::= *
242 <binario> ::= /
243 <binario> ::= //
244 <binario> ::= $+
245 <binario> ::= &+
246 <binario> ::= &-
247 <binario> ::= &++
248 <binario> ::= &--
249
250 <expresion-unaria>  ::= <operador-unario> <expresion-postfijo>
251
252 <operador-unario> ::= <unario>  <operador-unario>
253 <operador-unario> ::=
254
255 <unario>  ::= $?
256 <unario>  ::= |<?
257 <unario>  ::= |>?
258 <unario>  ::= @#?
259 <unario>  ::= @$?
260 <unario>  ::= &!
261 <unario>  ::= @>>
262 <unario>  ::= @<<
263 <unario>  ::= Ni'
264 <unario>  ::= Qo'
265 <unario>  ::= --
266 <unario>  ::= ++
267 <unario>  ::= +
268 <unario>  ::= -
269
270 <expresion-postfijo>  ::= <expresion-primaria>  <operador-postfijo>
271
272 <operador-postfijo> ::= <postfijo>  <operador-postfijo>
273 <operador-postfijo> ::=
274
275 <postfijo>  ::= --
276 <postfijo>  ::= ++
277 <postfijo>  ::= ( <args>  )
278 <postfijo>  ::= [ <expresion> ]
279 <postfijo>  ::= Suq <suq-aux>
280
```

```
<suq-aux> ::= |#|
<suq-aux> ::= |##|
<suq-aux> ::= |###|

<expresion-primaria>  ::= <literal>
<expresion-primaria>  ::= id
<expresion-primaria>  ::= ( <expresion> )

<preprograma> ::= <%: <preprograma-aux> :%>

<preprograma-aux> ::= Qonos literalCadena
<preprograma-aux> ::= Pigh  literalEntero

<preprograma-list>  ::= <preprograma> <preprograma-list>
<preprograma-list>  ::=

<secciones> ::= <seccion> <secciones>
<secciones> ::=

<seccion> ::= Nagh  <seccionConstantes> ^

<seccionConstantes> ::= Tagh  <declaracion-constante-list>  Hegh  ^
<seccionConstantes> ::=

<seccion> ::= Segh  <seccionUsuarios> ^

<seccionUsuarios> ::= Tagh  <declaracion-usuario-list>  Hegh  ^
<seccionUsuarios> ::=

<seccion> ::= Sar <seccionVariables>  ^

<seccionVariables>  ::= Tagh  <declaracion-list>  Hegh  ^
<seccionVariables>  ::=

<seccion> ::= Ghu <seccionPrototipos> ^

<seccionPrototipos> ::= Tagh  <prototipos>  Hegh  ^
<seccionPrototipos> ::=

<seccion> ::= Qap Ghom  <seccionRutinas>  ^

<seccionRutinas>  ::= Tagh  <rutinas> Hegh  ^
<seccionRutinas>  ::=

<seccion> ::= Chu'  <bloque-instruccion>  ^
```

4.3 Gramática con símbolos semánticos

```
<S> ::= #idPrograma Ghun  id  <preprograma-list>  <secciones>

<literal> ::= #tipoMu literalCadena
<literal> ::= #tipoMi literalEntero
<literal> ::= #tipoVit  <literalBooleana>
<literal> ::= #tipoVey  <literalConjunto>
<literal> ::= #tipoJaj  <literalFecha>
<literal> ::= #tipoDegh literalChar
<literal> ::= #tipoMorgh  <literalArreglo>
<literal> ::= #tipoUsuario  <literalRegistro>

<literalBooleana> ::= Teh
<literalBooleana> ::= Ngeb

<literalConjunto> ::= { <literalConjunto-aux> }

<literalConjunto-aux> ::= <literalConjunto-list>
<literalConjunto-aux> ::=

<literalConjunto-list>  ::= literalChar <literalConjunto-list-aux>

<literalConjunto-list-aux>  ::= , <literalConjunto-list>
<literalConjunto-list-aux>  ::=

<literalFecha>  ::= | literalEntero | literalEntero | literalEntero |

<literalArreglo>  ::= [ <literalArreglo-aux>  ]

<literalArreglo-aux>  ::= <literal-list>
<literalArreglo-aux>  ::=

<literal-list>  ::= <literal> <literalArreglo-aux-list>

<literalArreglo-aux-list> ::= , <literal-list>
<literalArreglo-aux-list> ::=

<literalRegistro> ::= <<  <literalArreglo-aux>  >>

<tipos> ::= Mi'
<tipos> ::= Mu'
<tipos> ::= Jaj
<tipos> ::= Degh
<tipos> ::= Vit
<tipos> ::= Vey

<tipos> ::= Morgh [ <expresion> ] Chen  <tipos>

<tipos> ::= #idTipos  id

```

```
50 <declaracion-rutina> ::= Qap #idFuncion id <declarador> : #tipoFuncion <
       tipos> #setVarRutinas <seccion-variables-rutina> #sotVarRutinas <bloque-
       instruccion> ^
51 <declaracion-rutina> ::= Miw #idProcedimiento id <declarador> <seccion-
       variables-rutina> <bloque-instruccion> ^
52 <declaracion-prototipo> ::= Qap #idProtoFuncion id <declarador> : #tipoProto
        <tipos>
53 <declaracion-prototipo> ::= Miw #idProtoProcedimiento id <declarador>
54
55 <seccion-variables-rutina> ::= <seccion-variables>
56 <seccion-variables-rutina> ::=
57
58 <declarador> ::= ( <parametros> )
59
60 <parametros> ::= #reiniciarLista <parametros-list>
61 <parametros> ::=
62
63 <parametros-list> ::= <parametro> ^ <parametros-list-aux>
64
65 <parametros-list-aux> ::= , #reiniciarParametro #incrementarPos <parametros-
       list>
66 <parametros-list-aux> ::=
67
68 <parametro> ::= <formal> #tipoParametro <tipos> <id-list>
69
70 <id-list> ::= #idParametro id <id-list-aux>
71
72 <id-list-aux> ::= , #incrementarPos <id-list>
73 <id-list-aux> ::=
74
75 <formal> ::= #parametroFormal Choh
76 <formal> ::=
77
78 <rutinas> ::= <rutinas-list>
79 <rutinas> ::=
80
81 <rutinas-list> ::= <declaracion-rutina> ^ <rutinas-list-aux>
82
83 <rutinas-list-aux> ::= <rutinas-list>
84 <rutinas-list-aux> ::=
85
86 <prototipos> ::= <prototipos-list>
87 <prototipos> ::=
88
89 <prototipos-list> ::= <declaracion-prototipo> ^ <prototipos-list-aux>
90
91 <prototipos-list-aux> ::= <prototipos-list>
92 <prototipos-list-aux> ::=
93
94 <declaracion> ::= #tipoParametro <tipos> <declaracion-ids> ^
95
```

```
96 <declaracion-ids> ::= #idDeclaracion  id  <declaracion-ids-aux>
97
98 <declaracion-ids-aux> ::= #agregarVariableNI  , <declaracion-ids>
99 <declaracion-ids-aux> ::= Tadmoh  <literal> <declaracion-ids-aux-1> #
       agregarVariable
100 <declaracion-ids-aux> ::=
101
102 <declaracion-ids-aux-1> ::= , <declaracion-ids>
103 <declaracion-ids-aux-1> ::=
104
105 <declaracion-list>  ::= <list-declaracion>
106 <declaracion-list>  ::=
107
108 <list-declaracion>  ::= <declaracion> <list-declaracion-aux>
109
110 <list-declaracion-aux>  ::= <list-declaracion>
111 <list-declaracion-aux>  ::=
112
113 <declaracion-usuario> ::= #declaracionUsuario id  Pong  <declaracion-usuario-
       aux> ^
114
115 <declaracion-usuario-aux> ::= #setTipoUsuario <tipos> #agregarUsuario
116 <declaracion-usuario-aux> ::= Ta  <atributos> Hegh  #agregarUsuario
117
118 <atributos> ::= #tipoParametro  <tipos> #idAtributo id  #agregarAtributo  ^ <
       atributos-aux>
119
120 <atributos-aux> ::= <atributos>
121 <atributos-aux> ::=
122
123 <declaracion-usuario-list>  ::= <list-declaracion-usuario>
124 <declaracion-usuario-list>  ::=
125
126 <list-declaracion-usuario>  ::= <declaracion-usuario> <list-declaracion-
       usuario-aux>
127
128 <list-declaracion-usuario-aux>  ::= <list-declaracion-usuario>
129 <list-declaracion-usuario-aux>  ::=
130
131 <declaracion-constante> ::= #tipoParametro  <tipos> #idConstante  id  Tadmoh
       <literal> #agregarConstante ^
132
133 <declaracion-constante-list>  ::= <list-declaracion-constante>
134 <declaracion-constante-list>  ::=
135
136 <list-declaracion-constante>  ::= <declaracion-constante> <list-declaracion-
       constante-aux>
137
138 <list-declaracion-constante-aux>  ::= <list-declaracion-constante>
139 <list-declaracion-constante-aux>  ::=
140
```

```
<instruccion> ::= Duh <expresion> : <instruccion> ^
<instruccion> ::= Latlh : <instruccion> ^
<instruccion> ::= Qap Vis <expresion> <instruccion> ^
<instruccion> ::= Be' <instruccion> Qap Qa' <expresion> ^
<instruccion> ::= Hoch id Qod <expresion> Be' <instruccion> ^
<instruccion> ::= Leq <expresion> <instruccion> ^
<instruccion> ::= Lo' <expresion> Be' <instruccion> ^
<instruccion> ::= Ghor ^
<instruccion> ::= Bup ^
<instruccion> ::= Tah ^
<instruccion> ::= Meh <for-aux1> Mo' <expresion> <for-aux2> Be' <instruccion
    > ^
<instruccion> ::= Chugh <expresion> <if-aux1> ^
<instruccion> ::= Hijol <return-aux> ^
<instruccion> ::= ~ <expresion> ^
<instruccion> ::= <bloque-instruccion> ^

<for-aux1> ::= id := <expresion>

<for-aux2> ::= Gho' <expresion>
<for-aux2> ::=

<if-aux1> ::= <if-aux2> <instruccion> <if-aux3>
<if-aux1> ::= Qod <expresion> Nawlogh <expresion> Vaj <instruccion>

<if-aux2> ::= Vaj
<if-aux2> ::=

<if-aux3> ::= Latlh <instruccion> ^
<if-aux3> ::=

<return-aux> ::= <expresion>
<return-aux> ::=

<bloque-instruccion> ::= Tagh <instruccion-list> Hegh

<instruccion-list> ::= <instruccion> <instruccion-list>
<instruccion-list> ::=

<args> ::= <args-list>
<args> ::=

<args-list> ::= <expresion> <args-list-aux>

<args-list-aux> ::= , <args-list>
<args-list-aux> ::=

<expresion> ::= <expresion-asignacion>

<expresion-asignacion> ::= <expresion-logica> <operador-asignacion>

```

```
<operador-asignacion> ::= <asignacion> <expresion-asignacion>
<operador-asignacion> ::=

<asignacion> ::= :=
<asignacion> ::= +=
<asignacion> ::= -=
<asignacion> ::= *=
<asignacion> ::= /=
<asignacion> ::= //=
<asignacion> ::= |+=
<asignacion> ::= |++=
<asignacion> ::= |+++=
<asignacion> ::= &+=
<asignacion> ::= &-=
<asignacion> ::= $+=

<expresion-logica> ::= <expresion-relacional> <operador-logico>

<operador-logico> ::= <logico> <expresion-logica>
<operador-logico> ::=

<logico> ::= 'ej
<logico> ::= Ghap
<logico> ::= Jog

<expresion-asignacion> ::= <expresion-logica> <operador-asignacion>

<operador-asignacion> ::= <asignacion> <expresion-asignacion>
<operador-asignacion> ::=

<expresion-trinaria> ::= <expresion-binaria> <operador-trinario>

<operador-trinario> ::= <trinario> <expresion-binaria> $$ <expresion-binaria
    >
<operador-trinario> ::=

<trinario> ::= <<
<trinario> ::= >>

<expresion-relacional> ::= <expresion-trinaria> <operador-relacional>

<operador-relacional> ::= <relacional> <expresion-relacional>
<operador-relacional> ::=

<relacional> ::= <
<relacional> ::= >
<relacional> ::= <=
<relacional> ::= >=
<relacional> ::= ==
<relacional> ::= <>

```

```
<expresion-binaria> ::= <expresion-unaria>  <operador-binario>

<operador-binario>  ::= <binario> <expresion-binaria>
<operador-binario>  ::=

<binario> ::= +
<binario> ::= -
<binario> ::= *
<binario> ::= /
<binario> ::= //
<binario> ::= $+
<binario> ::= &+
<binario> ::= &-
<binario> ::= &++
<binario> ::= &--
<binario> ::= $

<expresion-unaria>  ::= <operador-unario> <expresion-postfijo>

<operador-unario> ::= <unario>  <operador-unario>
<operador-unario> ::=

<unario>  ::= $?
<unario>  ::= |<?
<unario>  ::= |>?
<unario>  ::= @#?
<unario>  ::= @$?
<unario>  ::= &!
<unario>  ::= @>>
<unario>  ::= @<<
<unario>  ::= Ni'
<unario>  ::= Qo'
<unario>  ::= --
<unario>  ::= ++
<unario>  ::= +
<unario>  ::= -

<expresion-postfijo>  ::= <expresion-primaria>  <operador-postfijo>

<operador-postfijo> ::= <postfijo>  <operador-postfijo>
<operador-postfijo> ::=

<postfijo>  ::= --
<postfijo>  ::= ++
<postfijo>  ::= ( <args>  )
<postfijo>  ::= #arreglo1 [ <expresion> ]
<postfijo>  ::= Suq <suq-aux>
<postfijo>  ::= . id
<postfijo>  ::= <tipos> ( <expresion> )


```

```
<suq-aux> ::= |#|
<suq-aux> ::= |##|
<suq-aux> ::= |###|

<expresion-primaria>  ::= <literal>
<expresion-primaria>  ::= #idAux  id
<expresion-primaria>  ::= ( <expresion> )

<preprograma> ::= <%: <preprograma-aux> :%>

<preprograma-aux> ::= Qonos literalCadena
<preprograma-aux> ::= Pigh  literalEntero

<preprograma-list>  ::= <preprograma> <preprograma-list>
<preprograma-list>  ::=

<secciones> ::= <seccion> <secciones>
<secciones> ::=

<seccion> ::= #checkConstantes  Nagh  <seccionConstantes> ^

<seccionConstantes> ::= Tagh  <declaracion-constante-list>  Hegh  ^
<seccionConstantes> ::=

<seccion> ::= #checkUsuarios  Segh  <seccionUsuarios> ^

<seccionUsuarios> ::= Tagh  <declaracion-usuario-list>  Hegh  ^
<seccionUsuarios> ::=

<seccion-variables> ::= Sar <seccionVariables>  ^

<seccionVariables>  ::= Tagh  <declaracion-list>  Hegh  ^
<seccionVariables>  ::=

<seccion> ::= <seccion-variables>

<seccion> ::= #checkProtos  Ghu <seccionPrototipos> ^

<seccionPrototipos> ::= Tagh  <prototipos>  Hegh  ^
<seccionPrototipos> ::=

<seccion> ::= #checkRutinas Qap Ghom  <seccionRutinas>  ^

<seccionRutinas>  ::= Tagh  <rutinas> Hegh  ^
<seccionRutinas>  ::=

<seccion> ::= #checkProgramaP Chu'  <bloque-instruccion>  ^
```

5 Autómata, estados aceptadores y diagramas

5.1 Estados aceptadores

A continuación se listaran los estados aceptables del autómata:

- **Reconocedor de id:** estado q500
- **Reconocedor de $$:** estado q501
- **Reconocedor de $+:** estado q502
- **Reconocedor de $?:** estado q503
- **Reconocedor de &!:** estado q504
- **Reconocedor de &+:** estado q505
- **Reconocedor de &++:** estado q506
- **Reconocedor de &-:** estado q507
- **Reconocedor de &–:** estado q508
- **Reconocedor de (:** estado q509
- **Reconocedor de):** estado q510
- **Reconocedor de *:** estado q511
- **Reconocedor de *=:** estado q512
- **Reconocedor de +:** estado q513
- **Reconocedor de ++:** estado q514
- **Reconocedor de +=:** estado q515
- **Reconocedor de ,:** estado q516
- **Reconocedor de -:** estado q517
- **Reconocedor de –:** estado q518
- **Reconocedor de -=:** estado q519
- **Reconocedor de /:** estado q520
- **Reconocedor de //:** estado q521
- **Reconocedor de //=:** estado q522
- **Reconocedor de /=:** estado q523
- **Reconocedor de ::** estado q524
- **Reconocedor de :=:** estado q525
- **Reconocedor de <:** estado q526
- **Reconocedor de <<:** estado q527
- **Reconocedor de <=:** estado q528
- **Reconocedor de <>:** estado q529
- **Reconocedor de ==:** estado q530
- **Reconocedor de >:** estado q531
- **Reconocedor de >=:** estado q532
- **Reconocedor de >>:** estado q533
- **Reconocedor de @#?:** estado q534
- **Reconocedor de @$?:** estado q535
- **Reconocedor de @<<:** estado q536
- **Reconocedor de @>>:** estado q537
- **Reconocedor de [:** estado q538
- **Reconocedor de]:** estado q539
- **Reconocedor de circunflex(ˆ) :** estado q540
- **Reconocedor de {:** estado q541
- **Reconocedor de }:** estado q542
- **Reconocedor de |+=:** estado q543
- **Reconocedor de |++=:** estado q544

- **Reconocedor de de:** estado |+++=q545
- **Reconocedor de |#|:** estado q546
- **Reconocedor de |##|:** estado q547
- **Reconocedor de |###|:** estado q548
- **Reconocedor de 'ej:** estado q549
- **Reconocedor de Be':** estado q550
- **Reconocedor de Bup:** estado q551
- **Reconocedor de Chen:** estado q552
- **Reconocedor de Choh:** estado q553
- **Reconocedor de Chu':** estado q554
- **Reconocedor de Chugh:** estado q555
- **Reconocedor de Degh:** estado q556
- **Reconocedor de Duh:** estado q557
- **Reconocedor de Ghap:** estado q558
- **Reconocedor de Ghitlh:** estado q559
- **Reconocedor de Gho':** estado q560
- **Reconocedor de Ghom:** estado q561
- **Reconocedor de Ghor:** estado q562
- **Reconocedor de Ghu:** estado q563
- **Reconocedor de Ghun:** estado q564
- **Reconocedor de Hegh:** estado q565
- **Reconocedor de Hijol:** estado q566
- **Reconocedor de Hoch:** estado q567
- **Reconocedor de Jaj:** estado q568
- **Reconocedor de Jog:** estado q569
- **Reconocedor de Lad:** estado q570
- **Reconocedor de Latlh:** estado q571
- **Reconocedor de Leq:** estado q572
- **Reconocedor de Lo':** estado q573
- **Reconocedor de Meh:** estado q574
- **Reconocedor de Mi':** estado q575
- **Reconocedor de Miw:** estado q576
- **Reconocedor de Mo':** estado q577
- **Reconocedor de Morgh:** estado q578
- **Reconocedor de Mu':** estado q579
- **Reconocedor de Nagh:** estado q580
- **Reconocedor de Nawlogh:** estado q581
- **Reconocedor de Ngeb:** estado q582
- **Reconocedor de Ni':** estado q583
- **Reconocedor de Pigh:** estado q584
- **Reconocedor de Pong:** estado q585
- **Reconocedor de Qa':** estado q586
- **Reconocedor de Qap:** estado q587
- **Reconocedor de Qo':** estado q588
- **Reconocedor de Qod:** estado q589
- **Reconocedor de Qonos:** estado q590
- **Reconocedor de Sar:** estado q591
- **Reconocedor de Segh:** estado q592
- **Reconocedor de Suq:** estado q593
- **Reconocedor de Ta:** estado q594

- **Reconocedor de Tadmoh:** estado q595
- **Reconocedor de Tagh:** estado q596
- **Reconocedor de Tah:** estado q597
- **Reconocedor de Teh:** estado q598
- **Reconocedor de Vaj:** estado q599
- **Reconocedor de Vey:** estado q600
- **Reconocedor de Vis:** estado q601
- **Reconocedor de Vit:** estado q602
- **Reconocedor de <%::** estado q603
- **Reconocedor de :%>:** estado q604
- **Reconocedor de | <?:** estado q605
- **Reconocedor de | >?:** estado q606
- **Reconocedor de |:** estado q607
- **Reconocedor de numero:** estado q608
- **Reconocedor de 'ej:** estado q609
- **Reconocedor de caracter:** estado q610
- **Reconocedor de ER:** estado q611
- **Reconocedor de ":** estado q612
- **Reconocedor de =:** estado q613

5.2 Estados aceptadores de errores

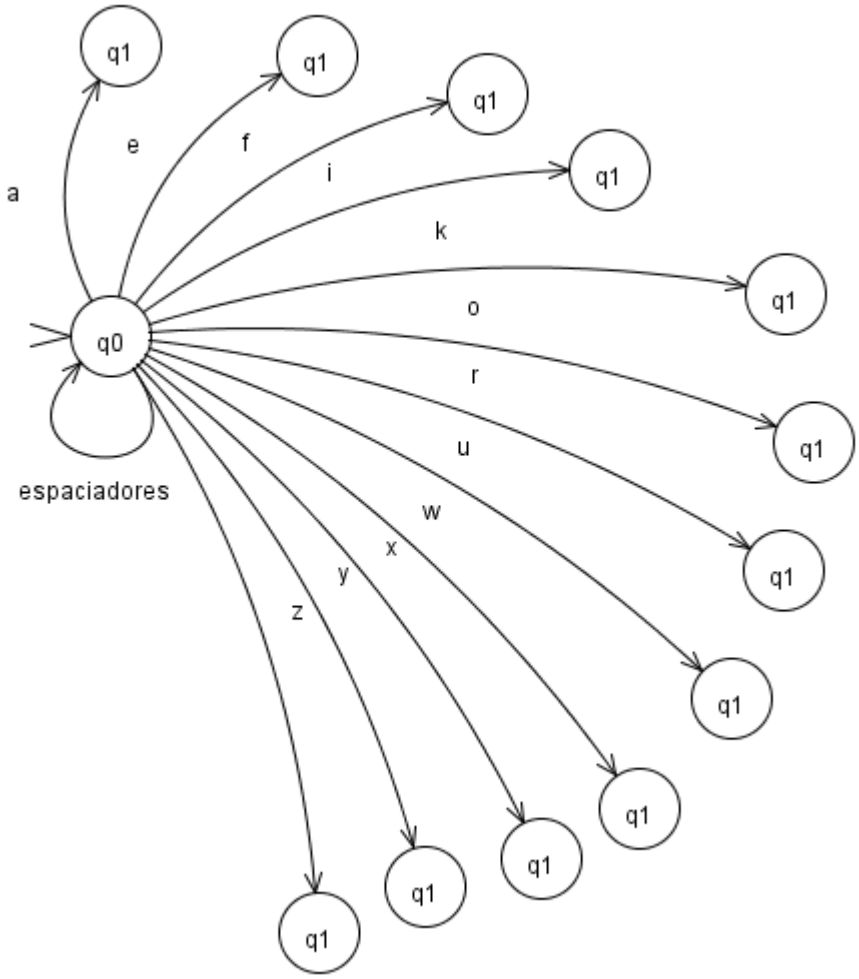
- **Estado 800:** Caracter monstruo
- **Estado 801:** Letra, caractér o número fuera de lugar
- **Estado 802:** Espaciador fuera de lugar
- **Estado 803:** Número demasiado grande
- **Estado 804:** Caracter no reconocido(La mayoría asciis menores a 32)

5.3 Diagramas

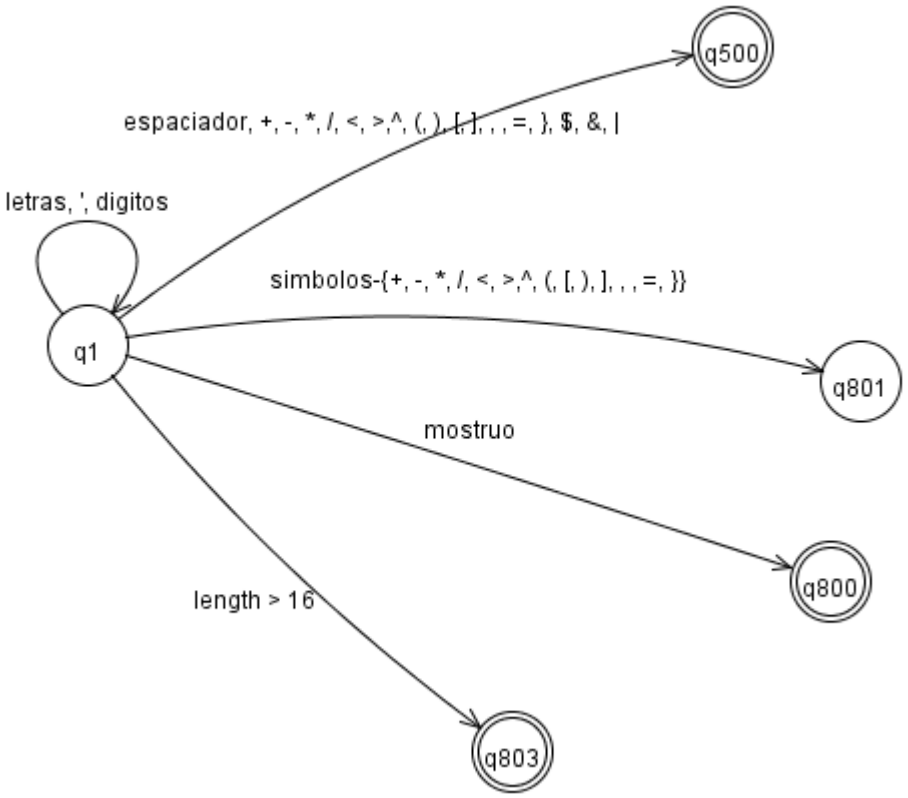
Antes de continuar con los diagramas, hay que resaltar algunas cosas:

- Todos los estados tienen una transición al estado 800 con un caracter monstruo
- Para el estado 1 que es el que está en proceso de reconocer un id, se mueve ya sea con un numero, flecha o comilla(').
- Para el estado 500 que es el que reconoce un id, se mueve con los espaciadores y estos simbolos:
  - +
  - -
  - *
  - /
  -)
  -]
  - ,
  - <
  - =
  - >
  - |
  - ^

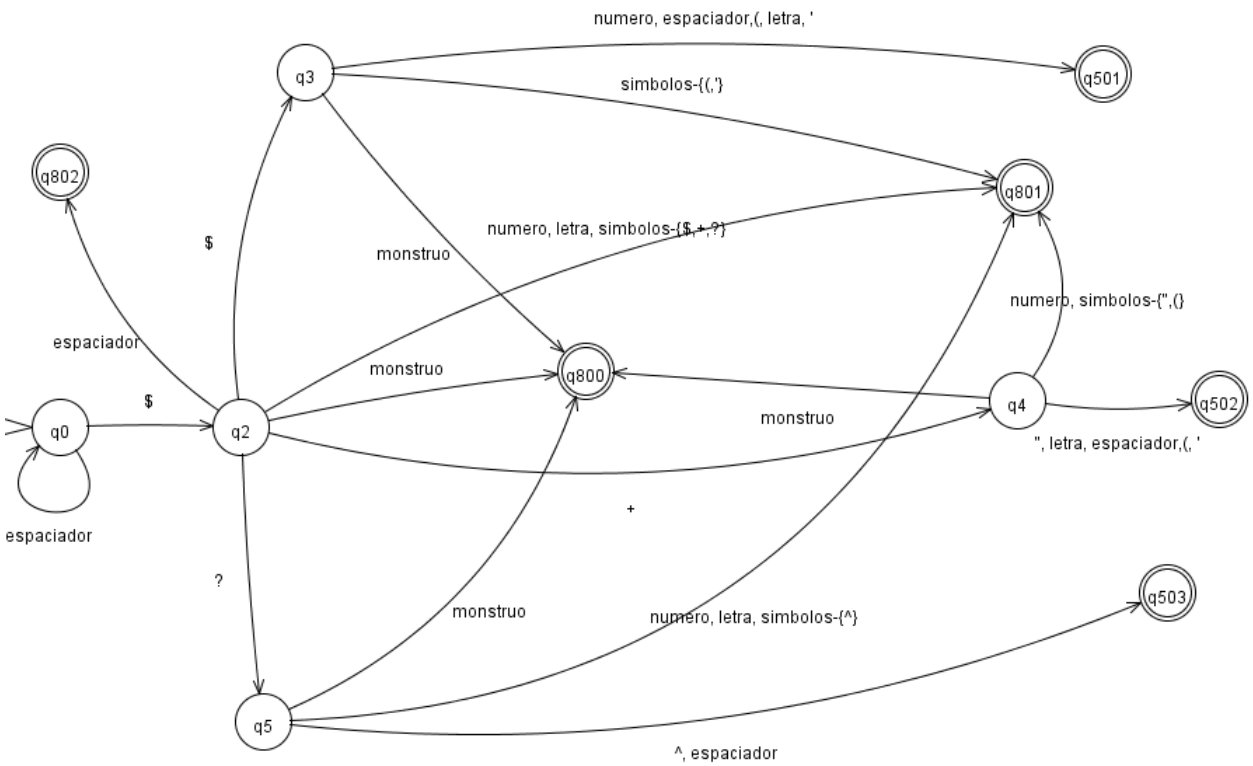
5.3.1 Reconocedor de las letras que no se usan en las palabras reservadas, estado 0



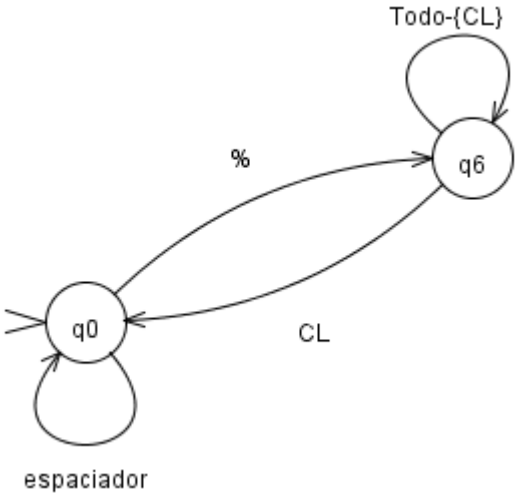
5.3.2 Reconocedor de ids en formación, estado 1



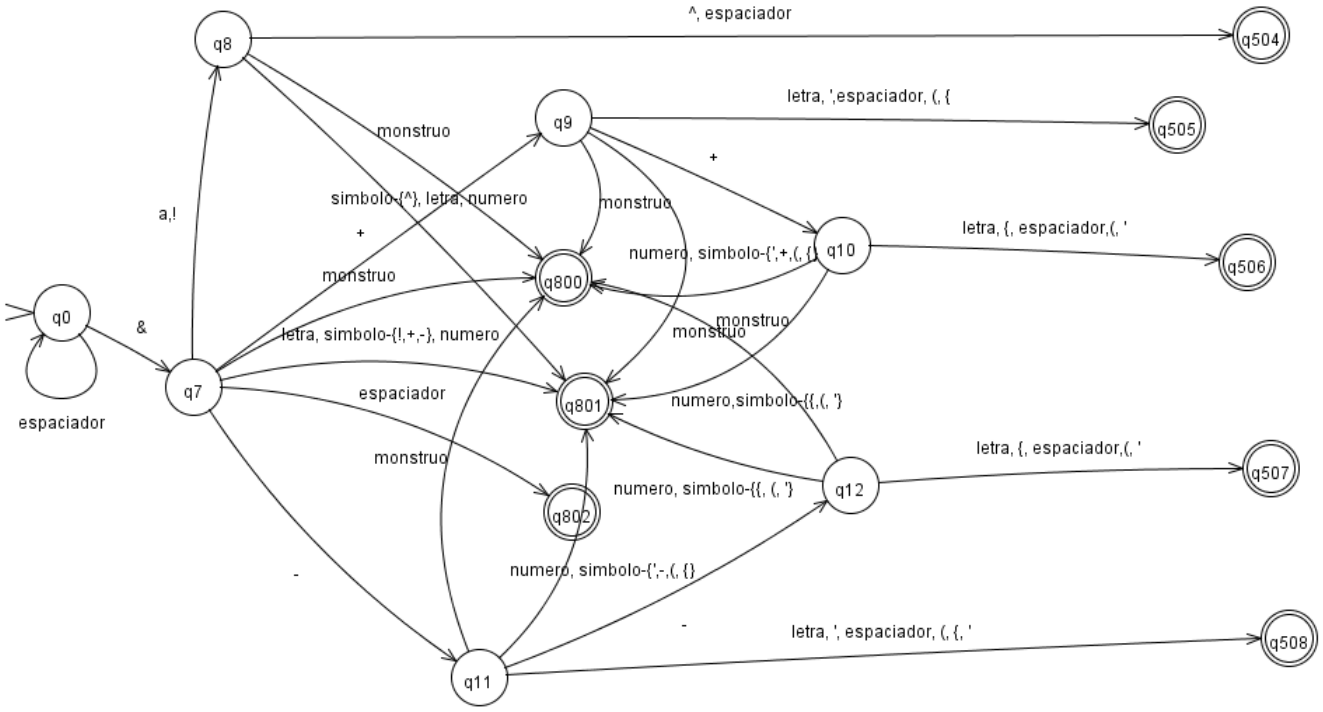
5.3.3 Reconocedor de $, estados q2-q5



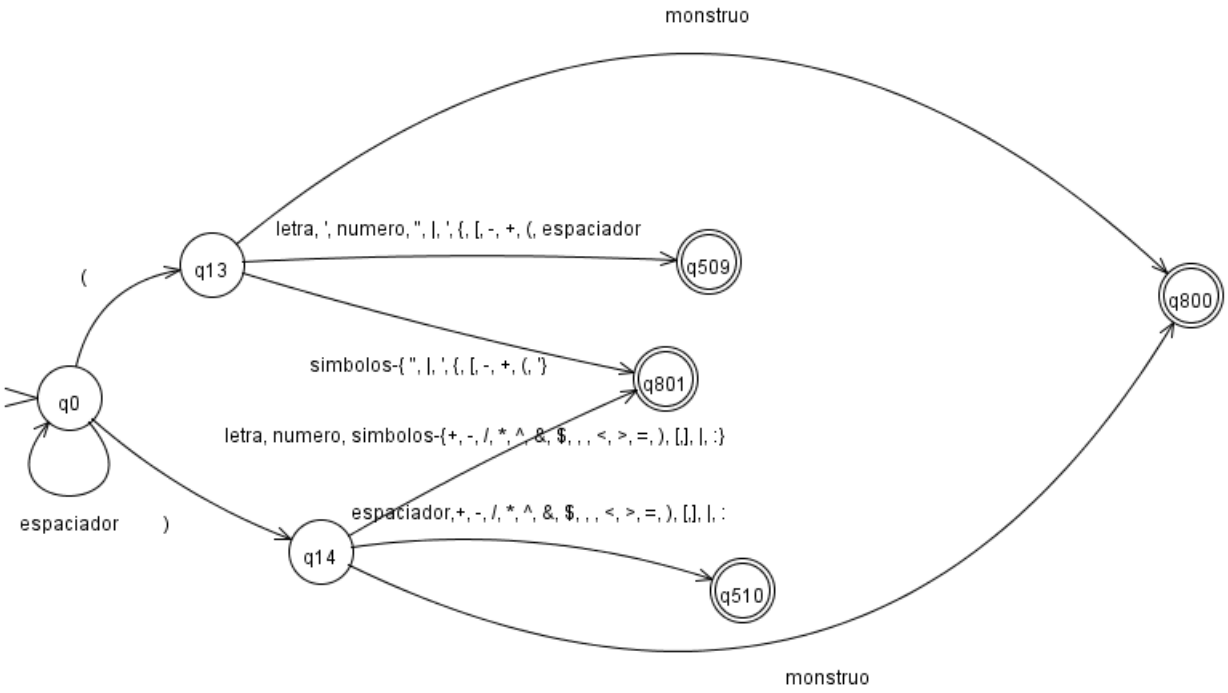
5.3.4 Reconocedor de %, estado q6



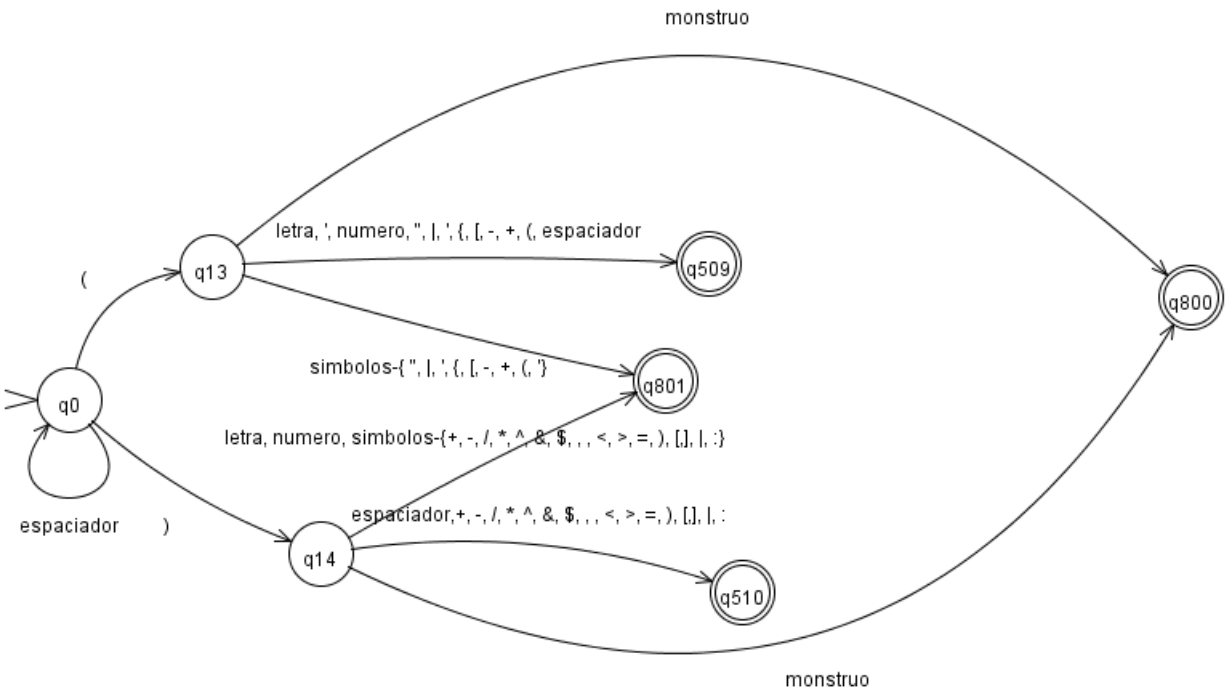
5.3.5 Reconocedor de &, estado q7-q12



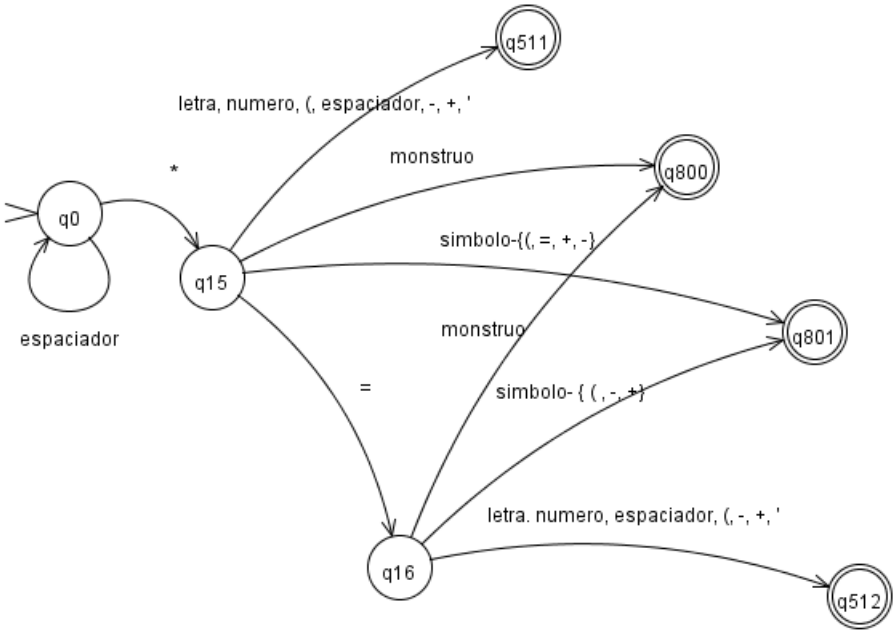
5.3.6 Reconocedor de (y), estado q13-q14



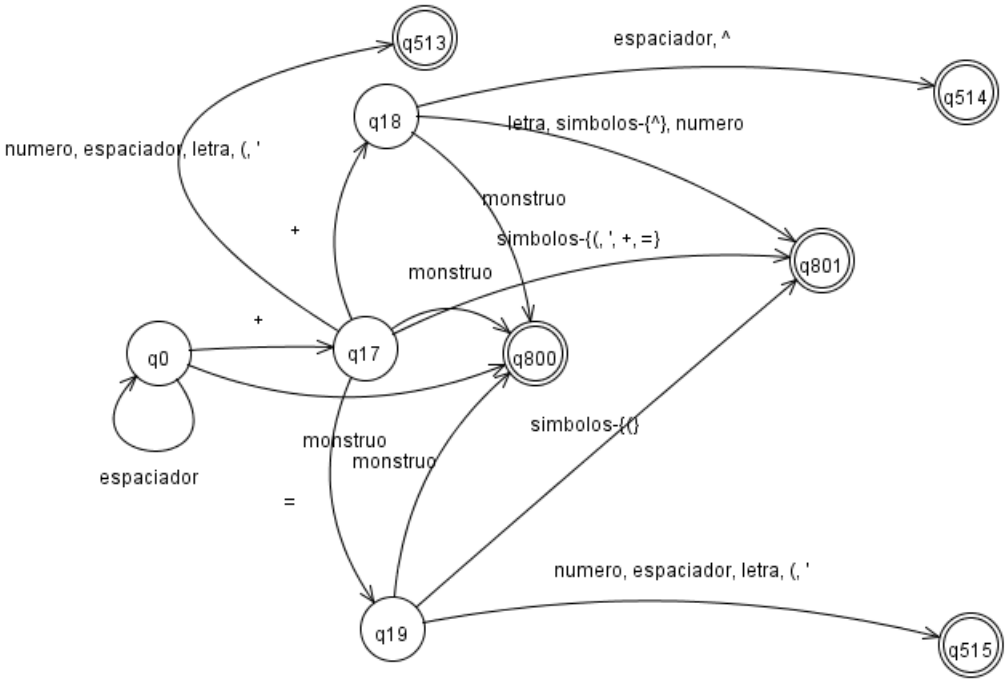
5.3.7 Reconocedor de (y), estado q13-q14



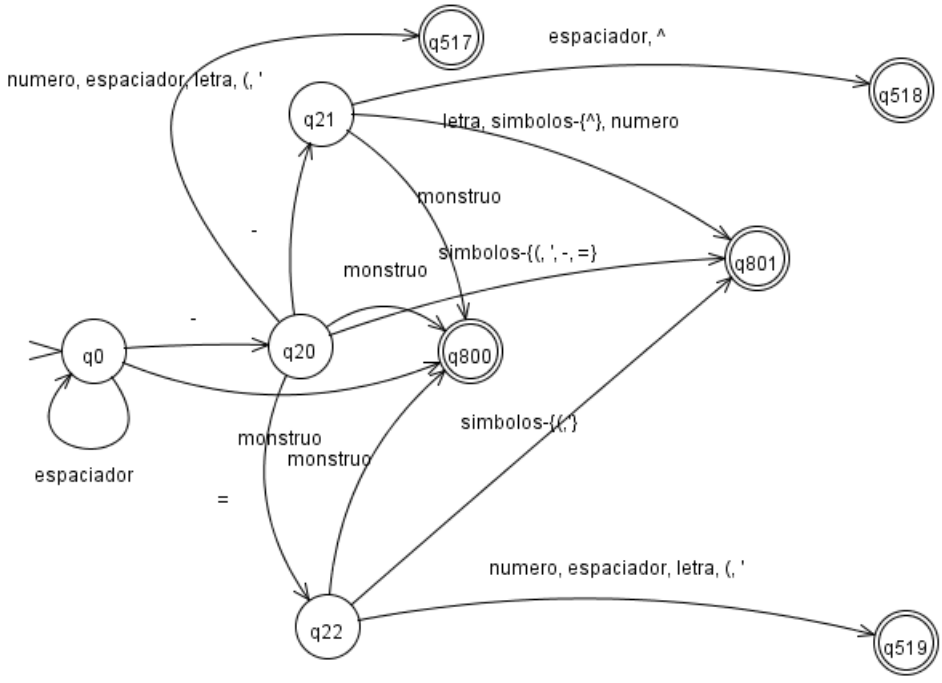
5.3.8 Reconocedor de *, estado q15-q16



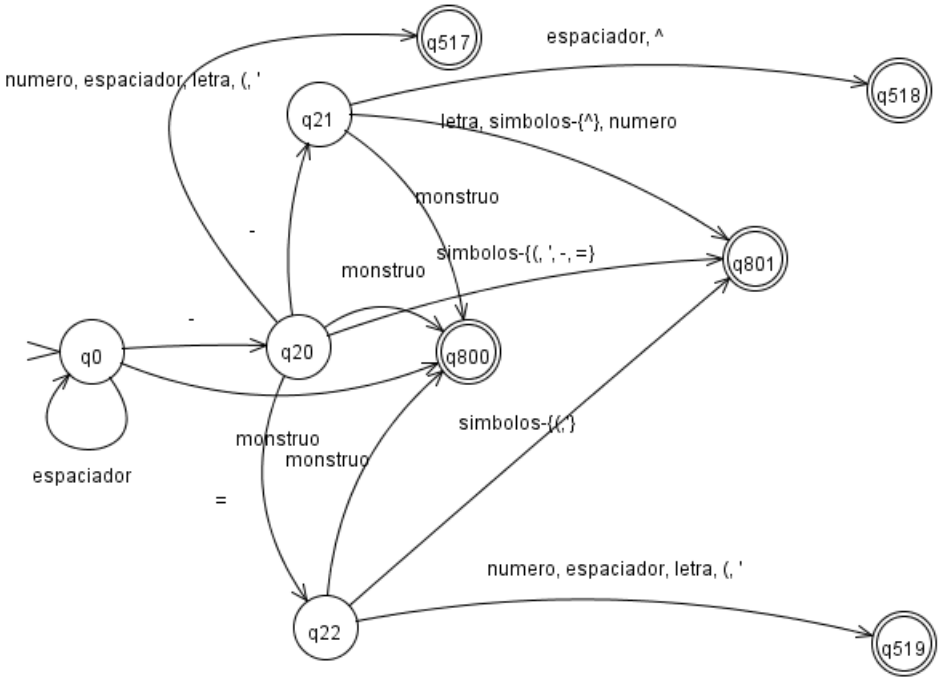
5.3.9 Reconocedor de +, estado q17-q19



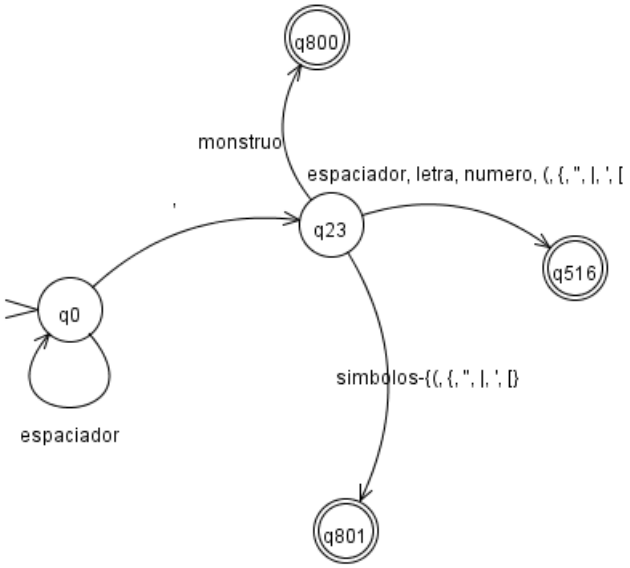
5.3.10 Reconocedor de -, estado q20-q22



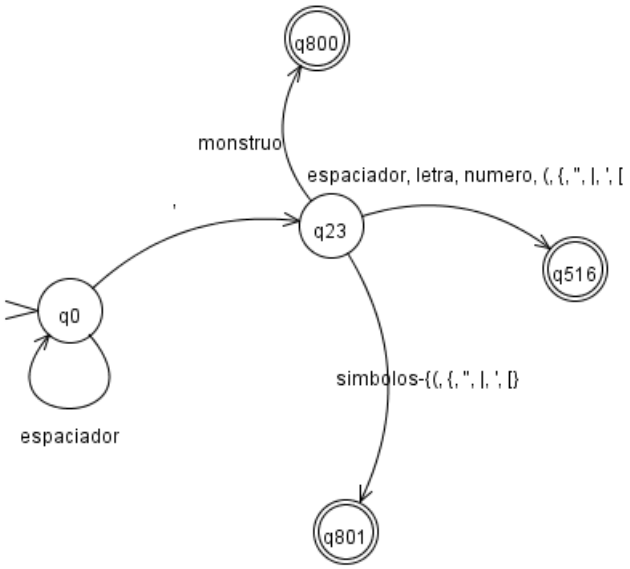
5.3.11 Reconocedor de -, estado q20-q22



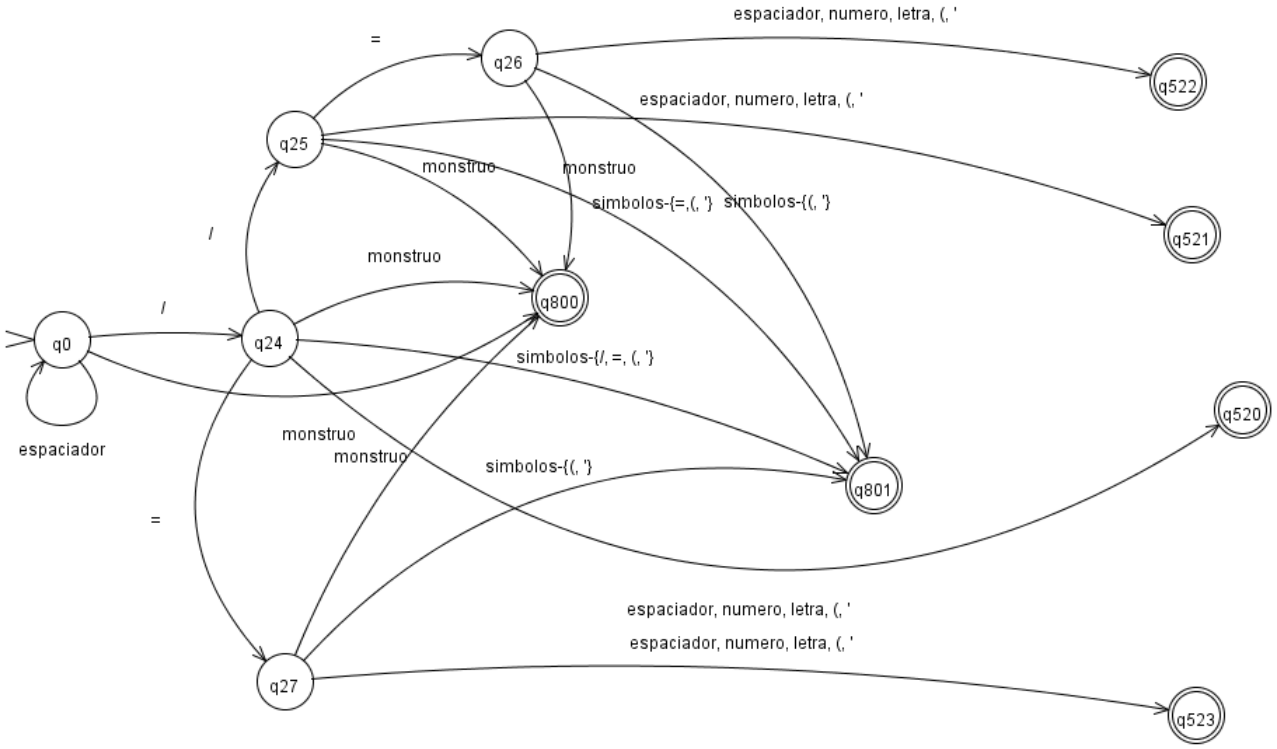
5.3.12 Reconocedor de coma(,), estado q23



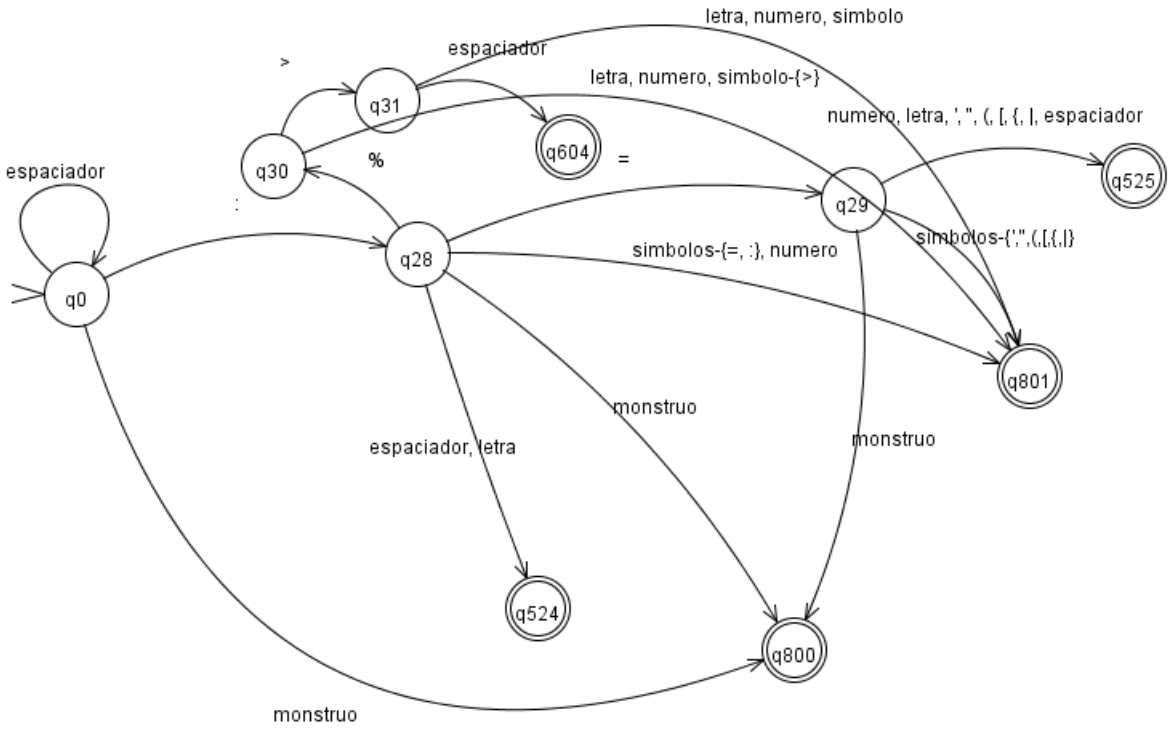
5.3.13 Reconocedor de coma(,), estado q23



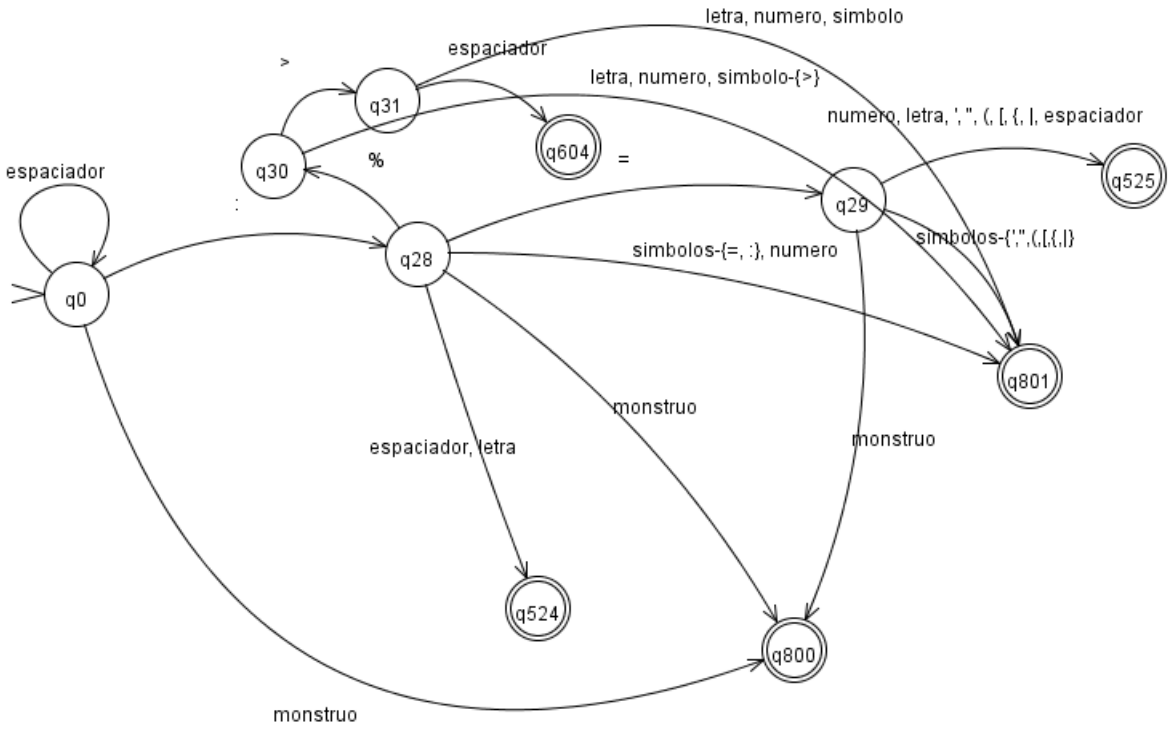
5.3.14 Reconocedor de /, estado q24-q27



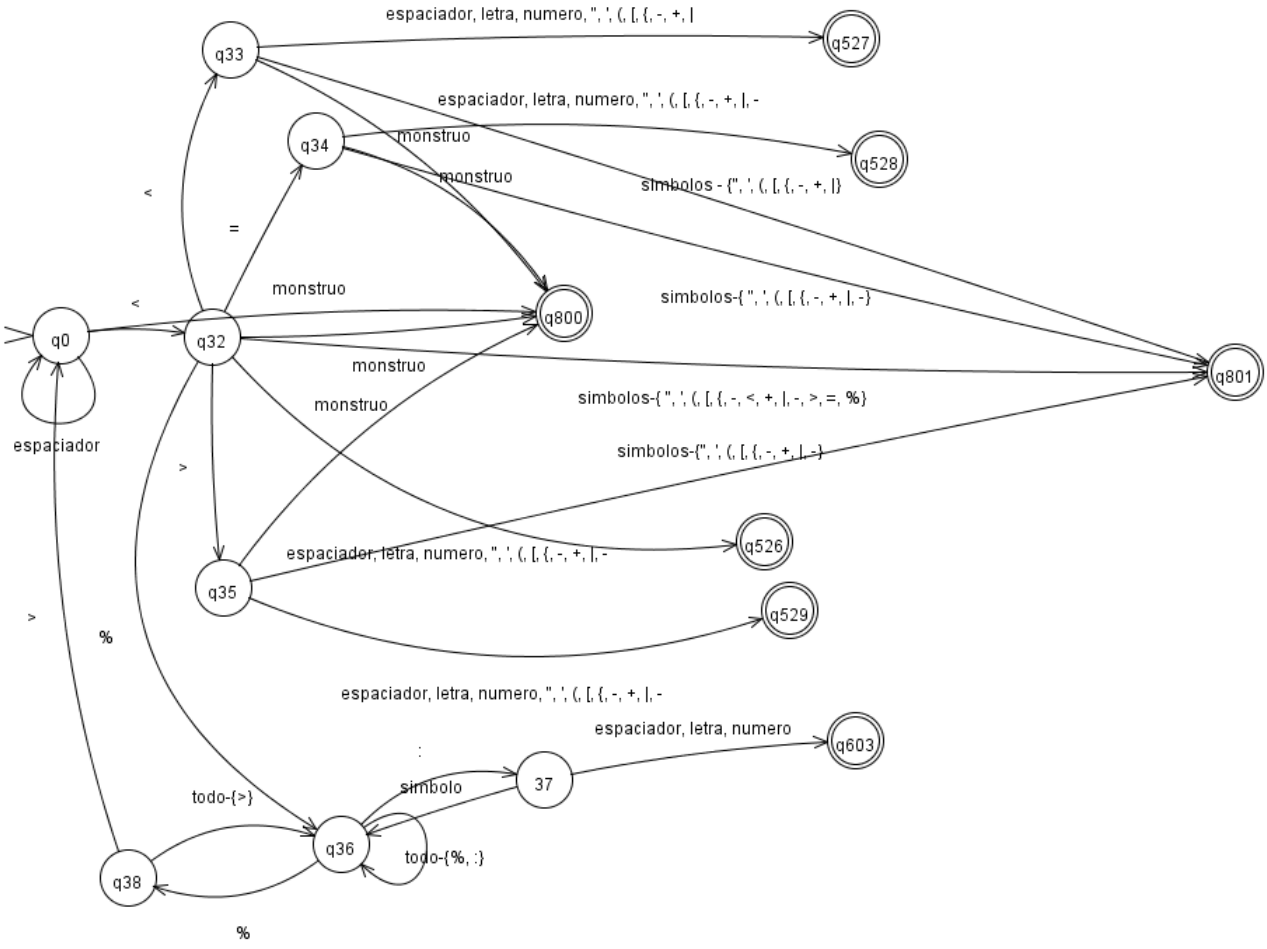
5.3.15 Reconocedor de :, estado q28-q31



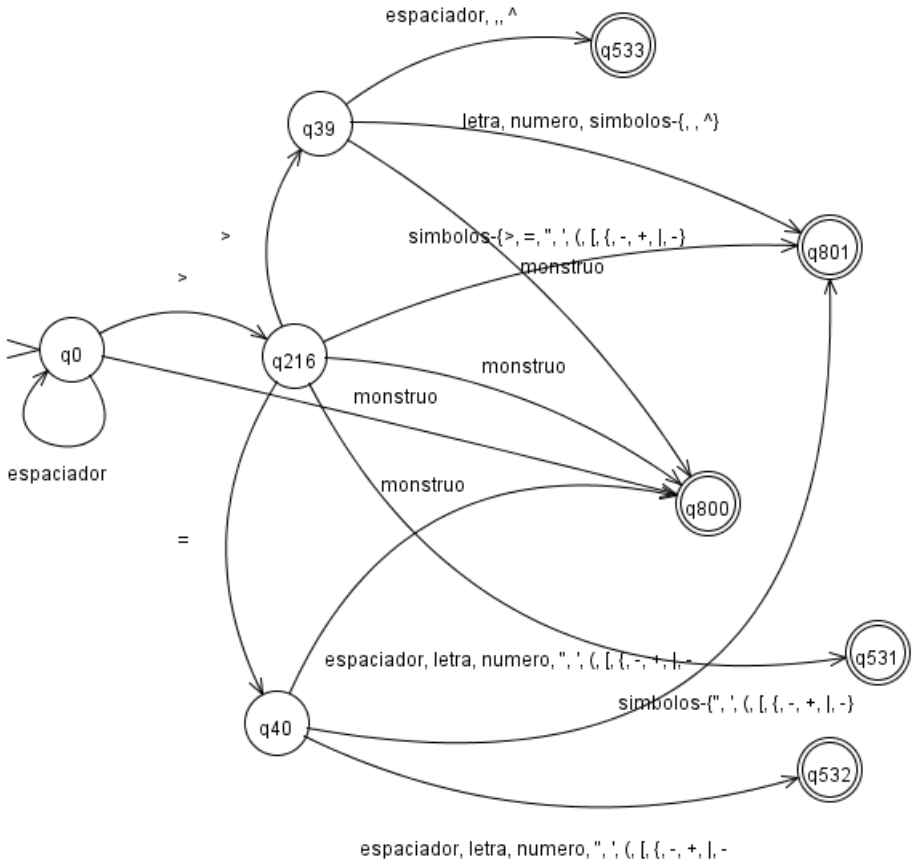
5.3.16 Reconocedor de :, estado q28-q31



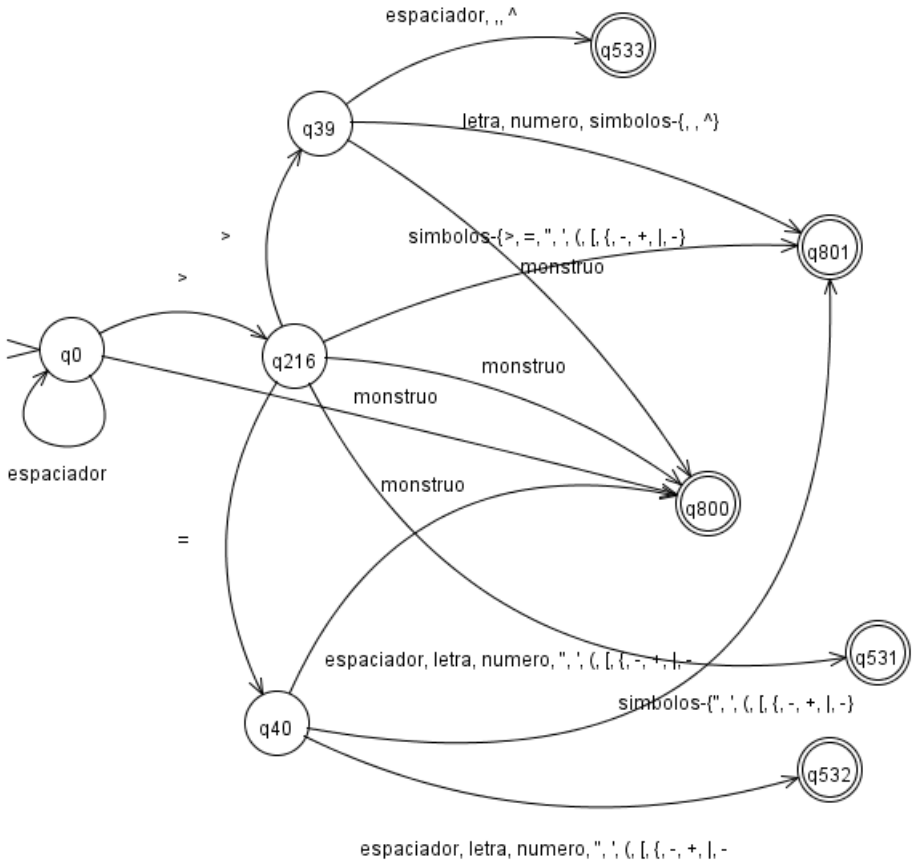
5.3.17 Reconocedor de <, estado q32-q38



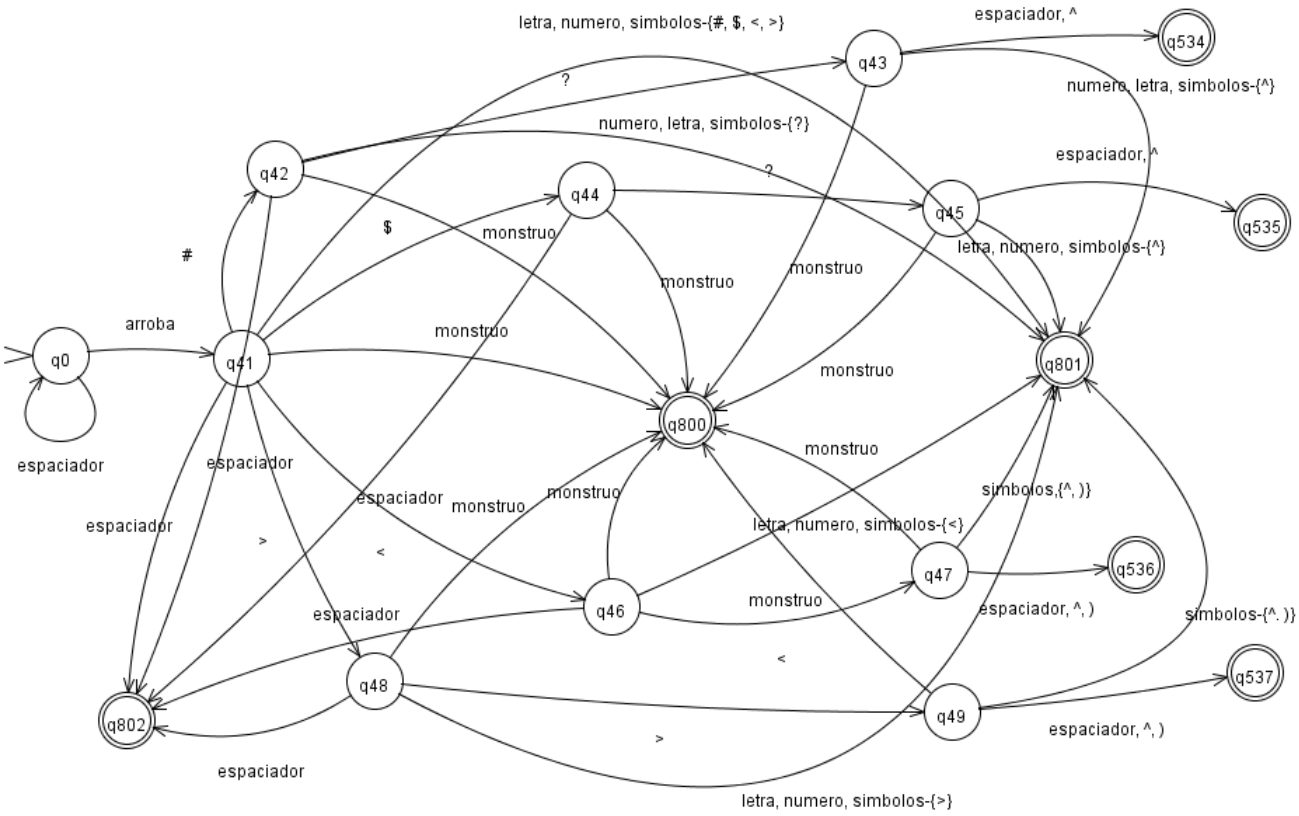
5.3.18 Reconocedor de >, estado q216-q40



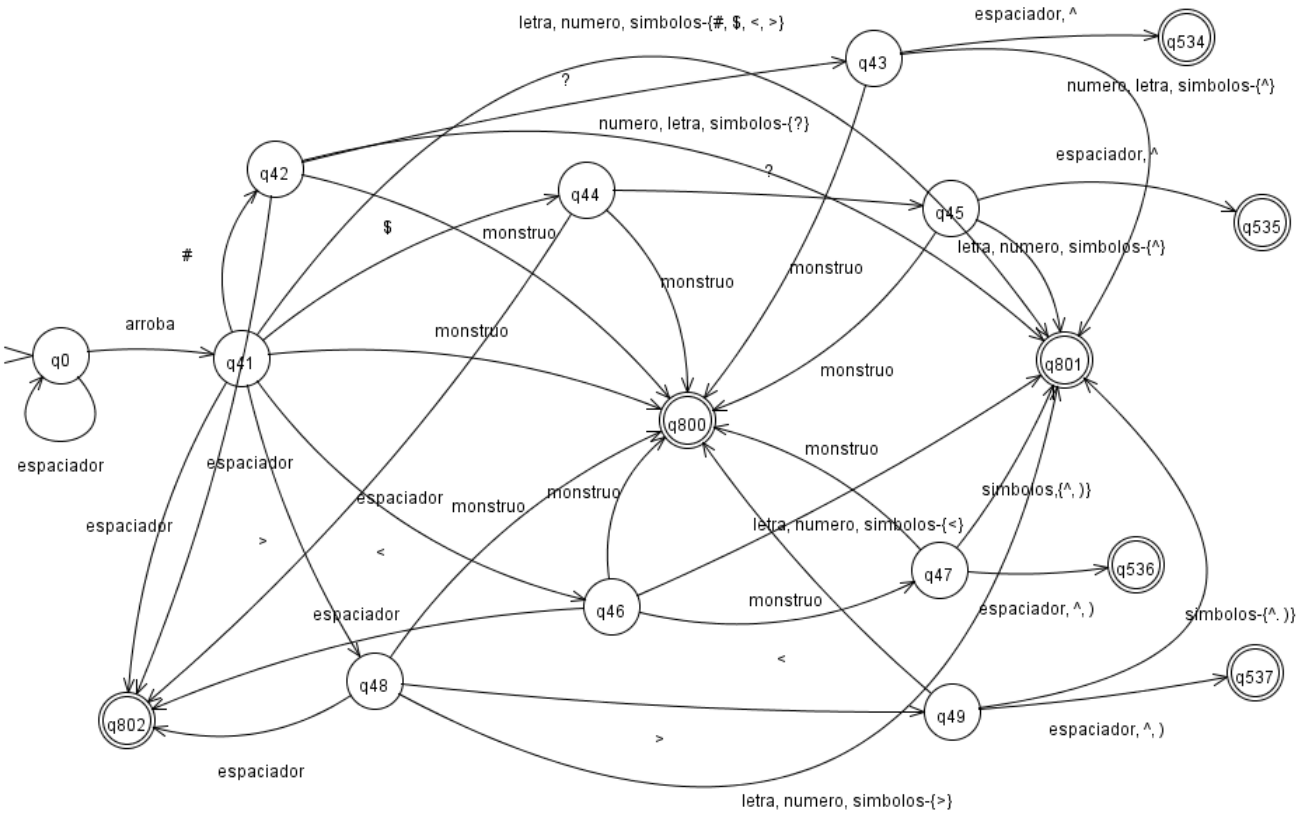
5.3.19 Reconocedor de >, estado q216-q40



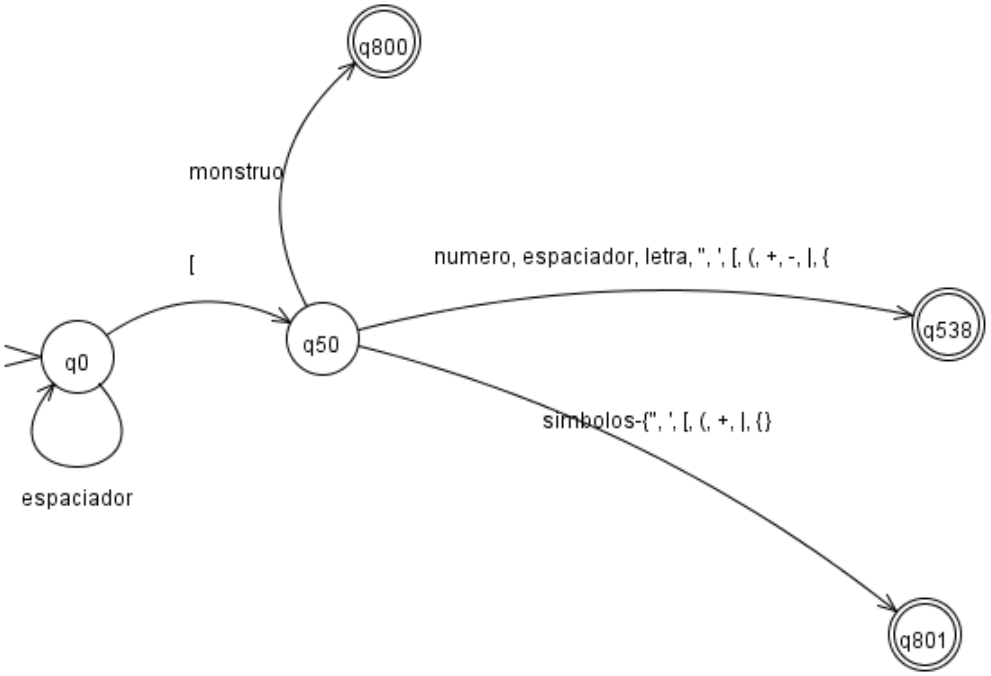
5.3.20 Reconocedor de @, estado q41-q49



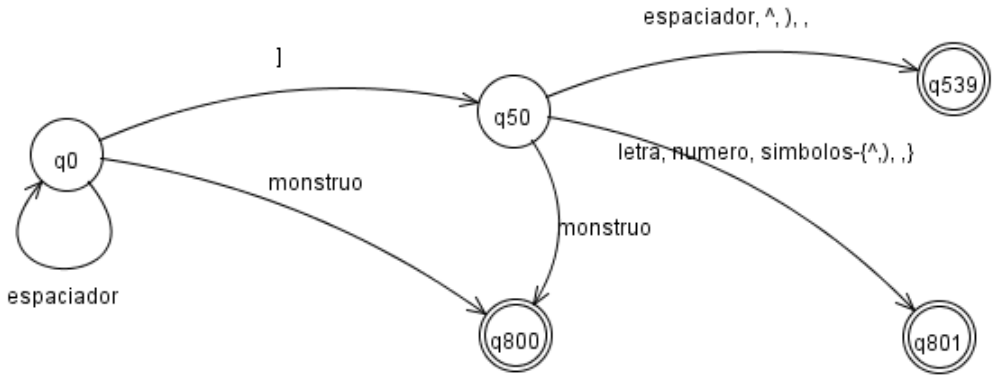
5.3.21 Reconocedor de @, estado q41-q49



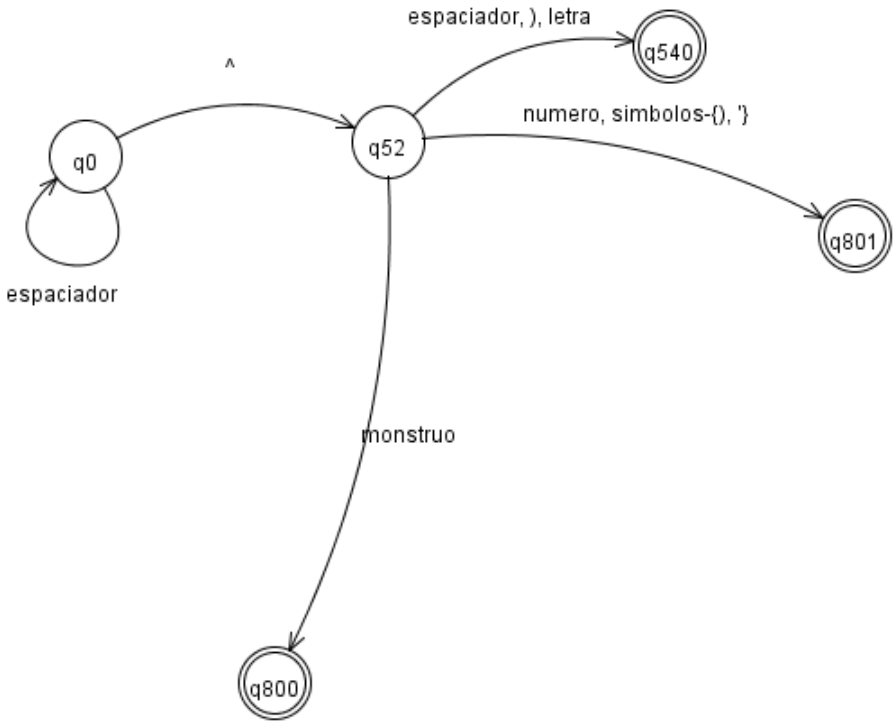
5.3.22 Reconocedor de [, estado q50



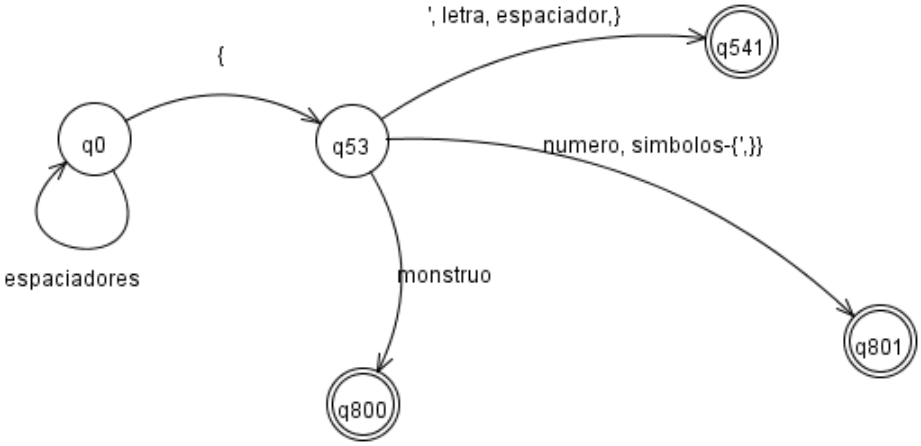
5.3.23 Reconocedor de], estado q51



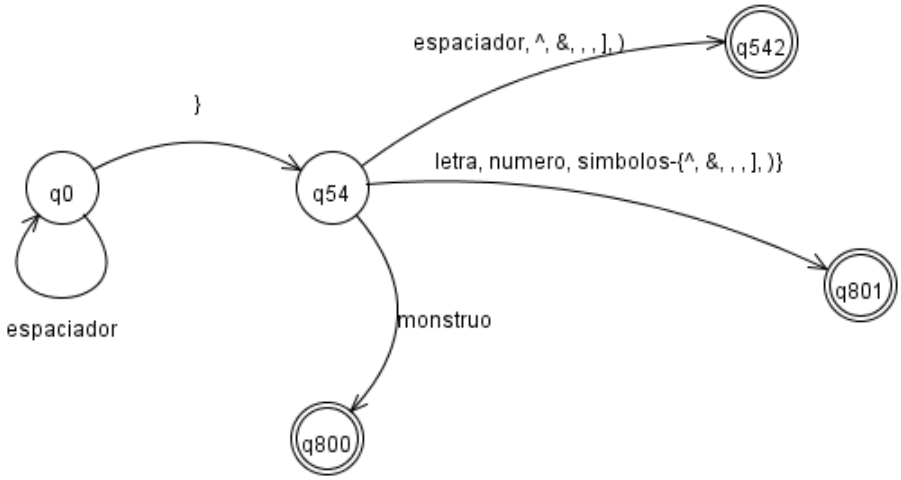
5.3.24 Reconocedor de circunflex(ˆ), estado q52



5.3.25 Reconocedor de {, estado q53



5.3.26 Reconocedor de }, estado q54



5.3.27 Reconocedor de |, estado q55-q71

Figure 1: Simple

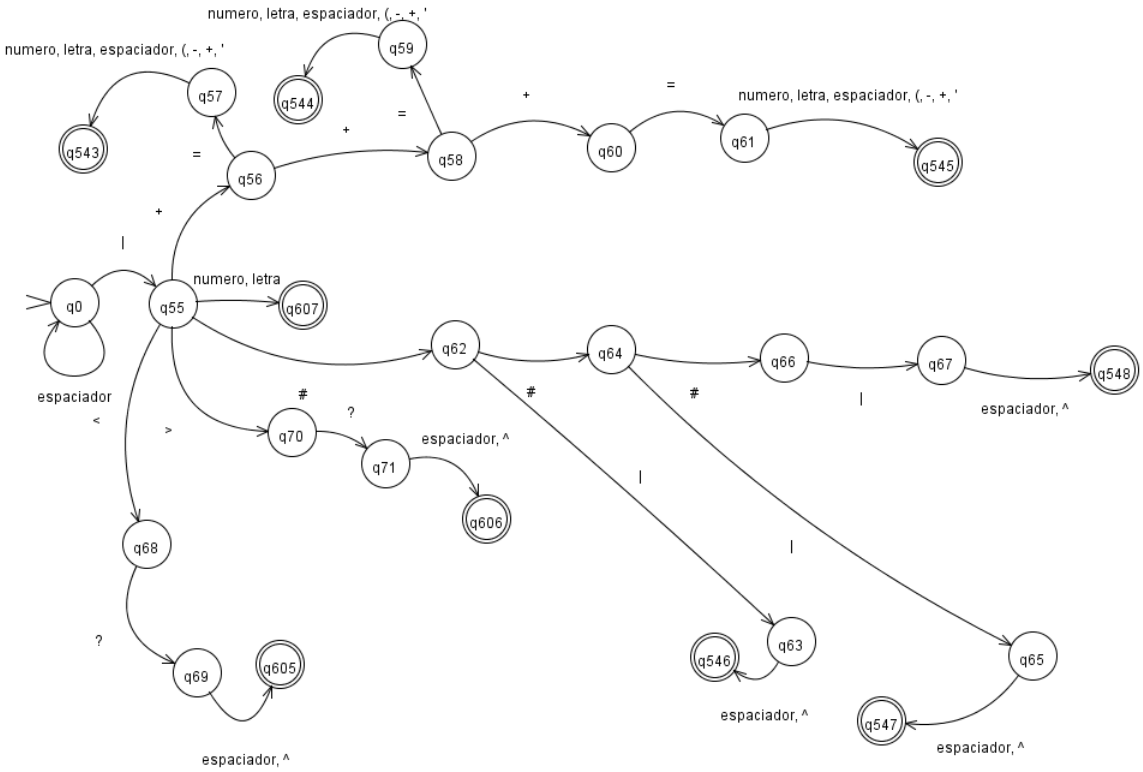


Figure 2: Error 800

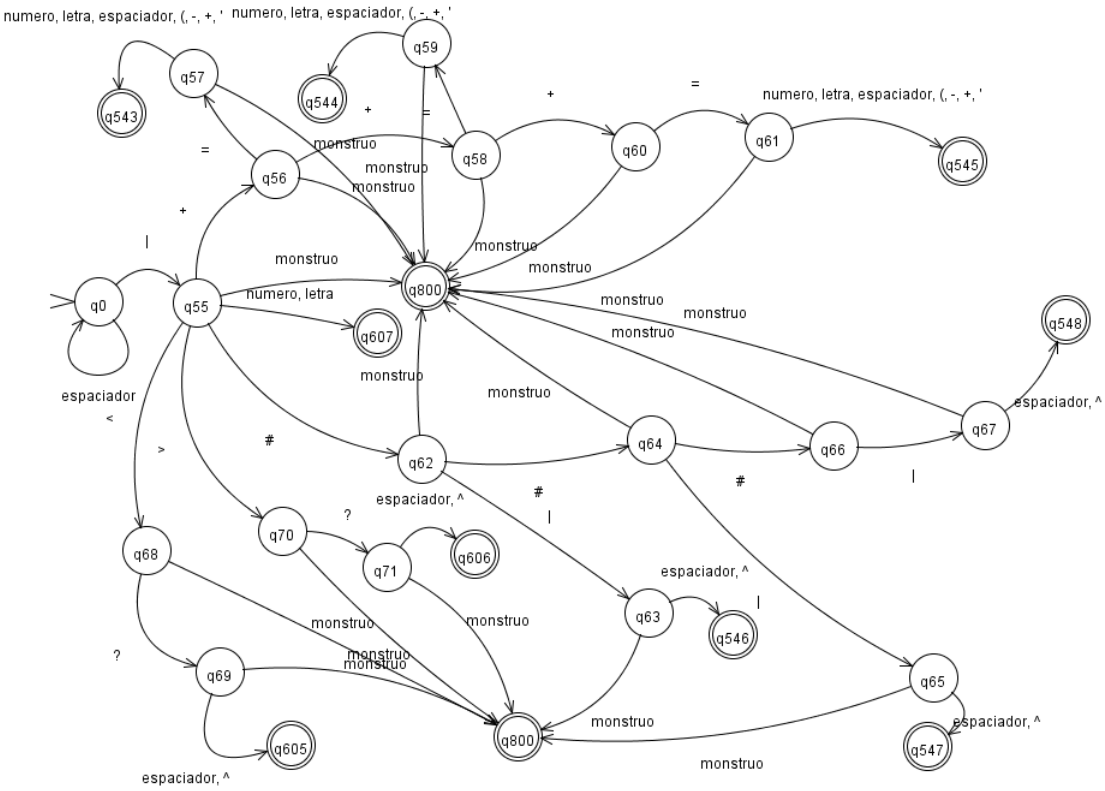


Figure 3: Error 801

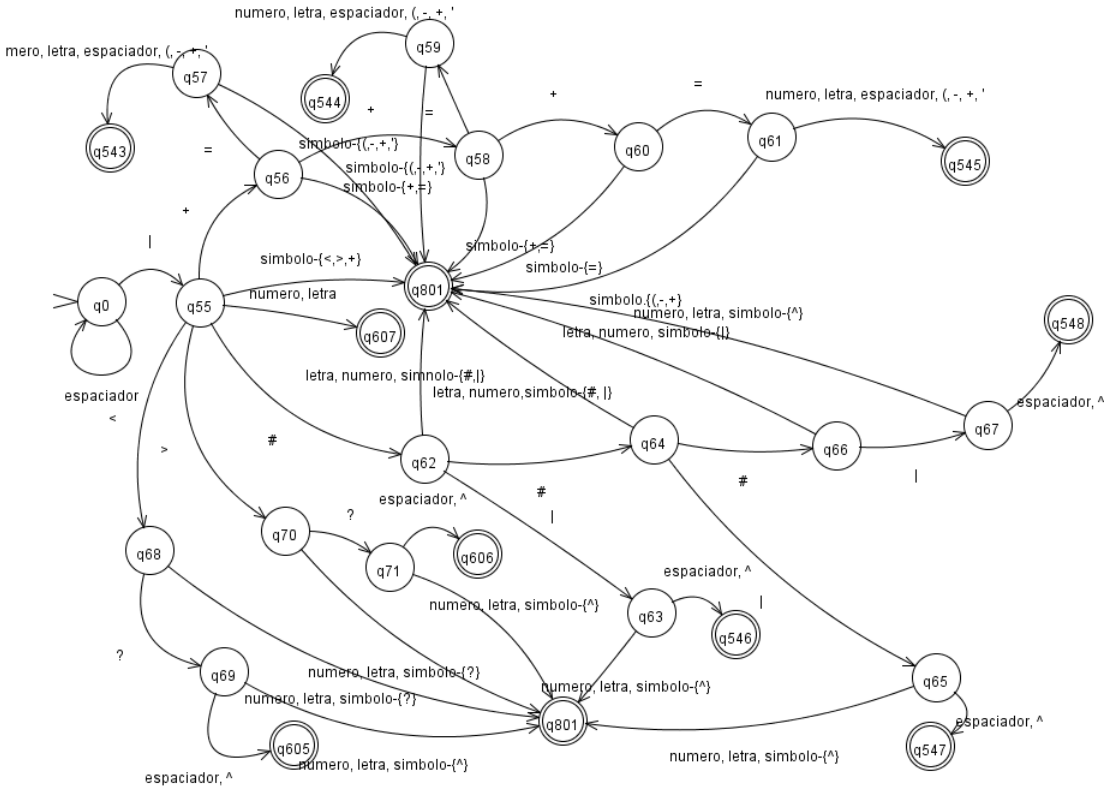
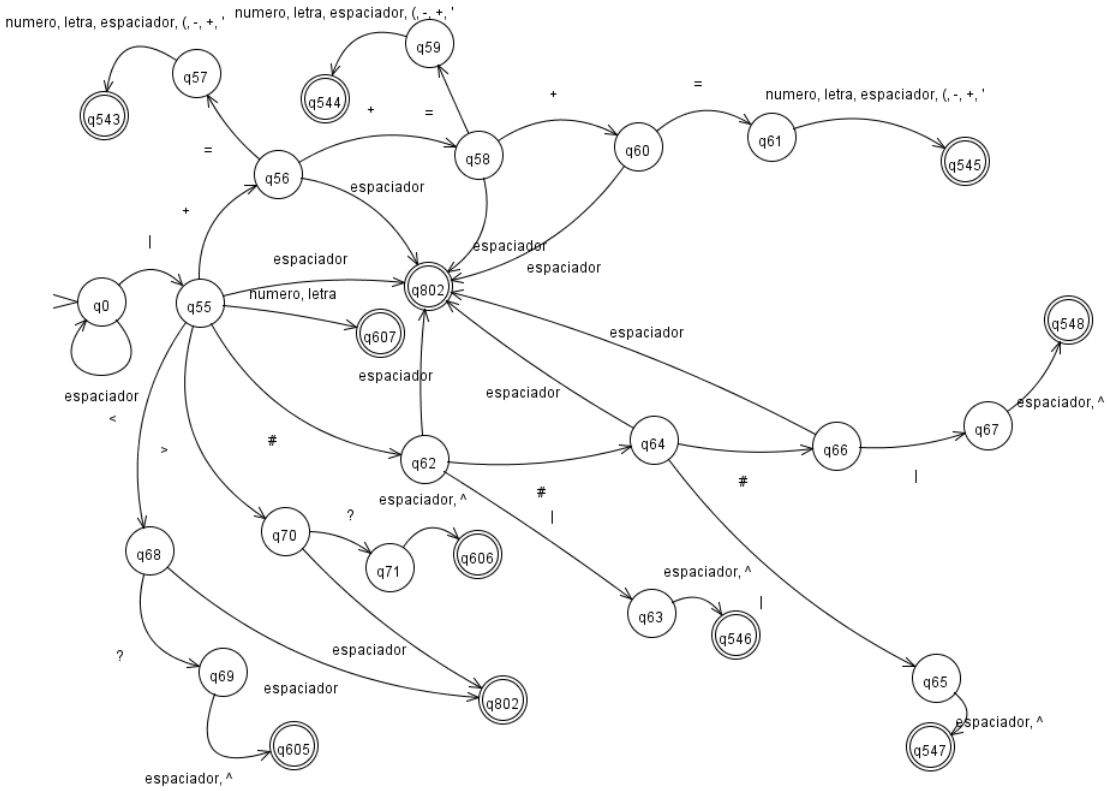


Figure 4: Error 802



5.3.28 Reconocedor de b, estado q72-q76

Figure 5: Estado 1 y 800

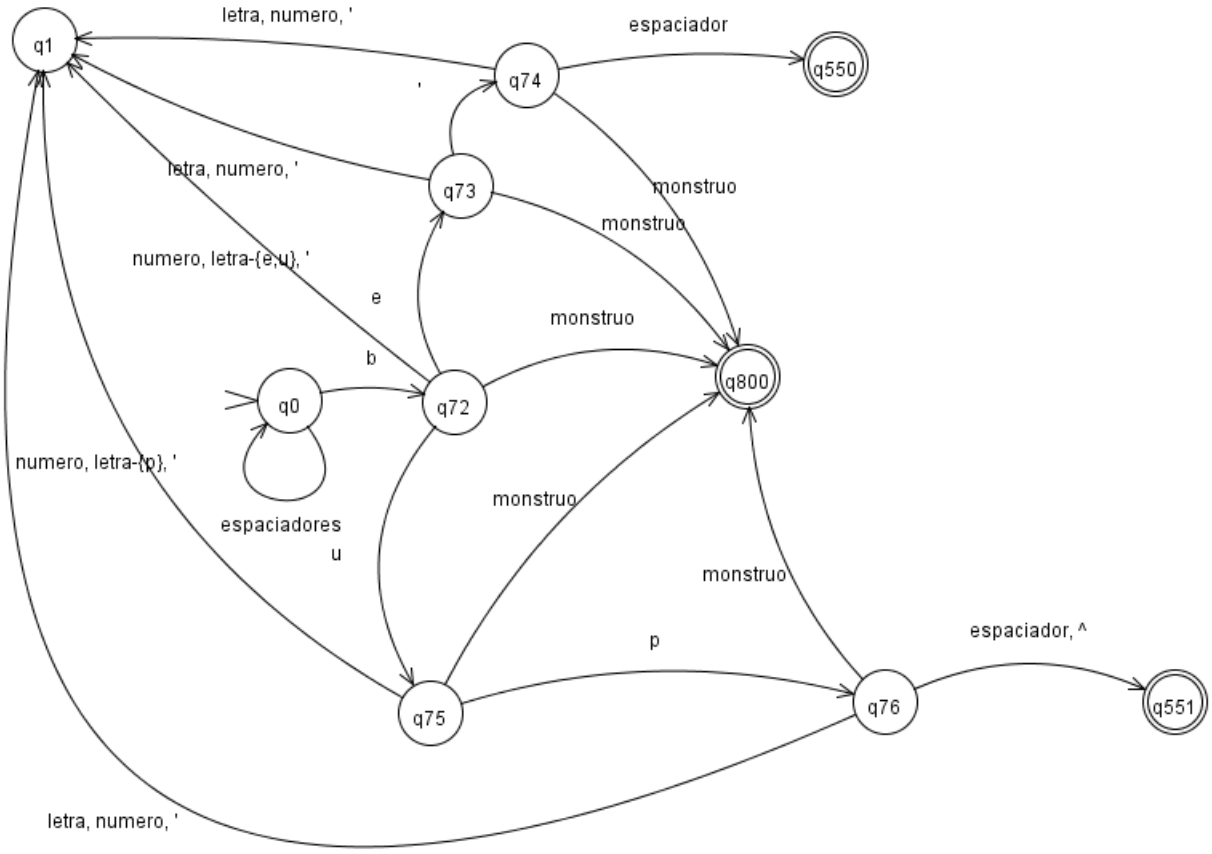
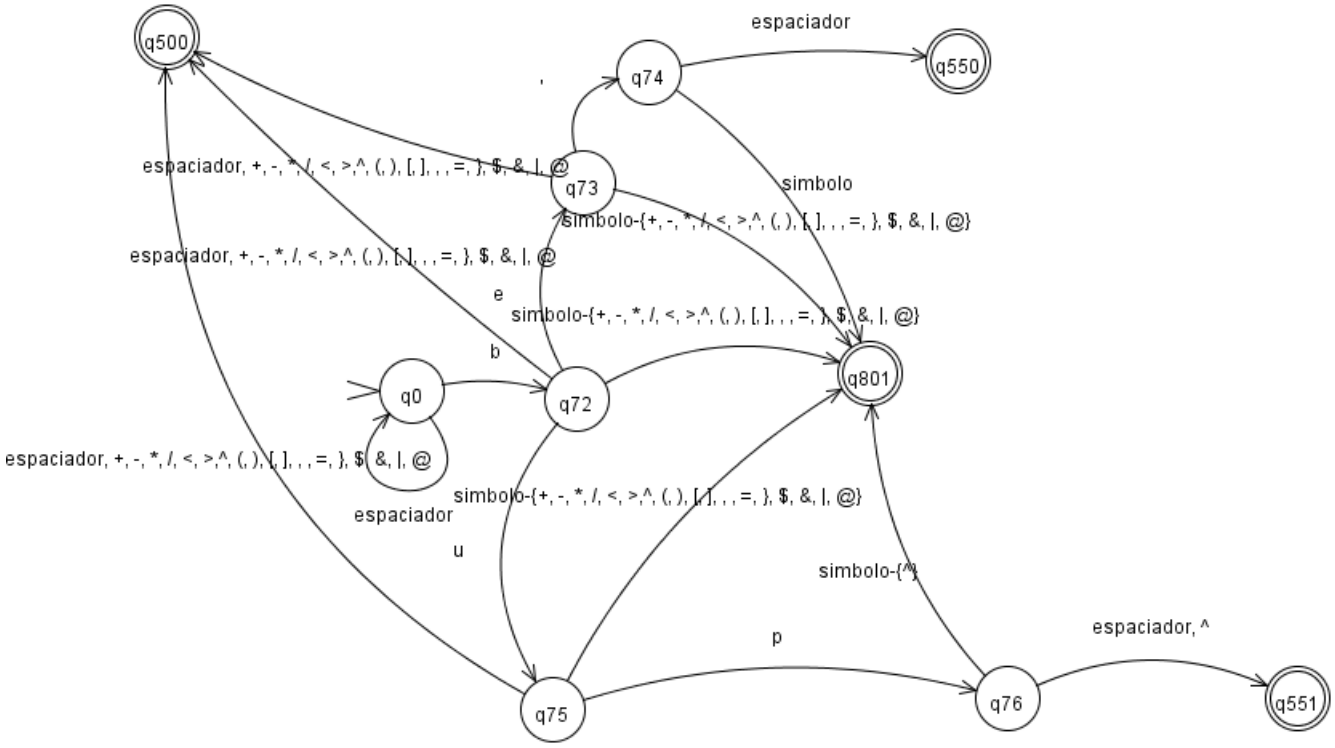


Figure 6: Estado 500 y 801



5.3.29 Reconocedor de c, estado q77-q86

Figure 7: Estado 1

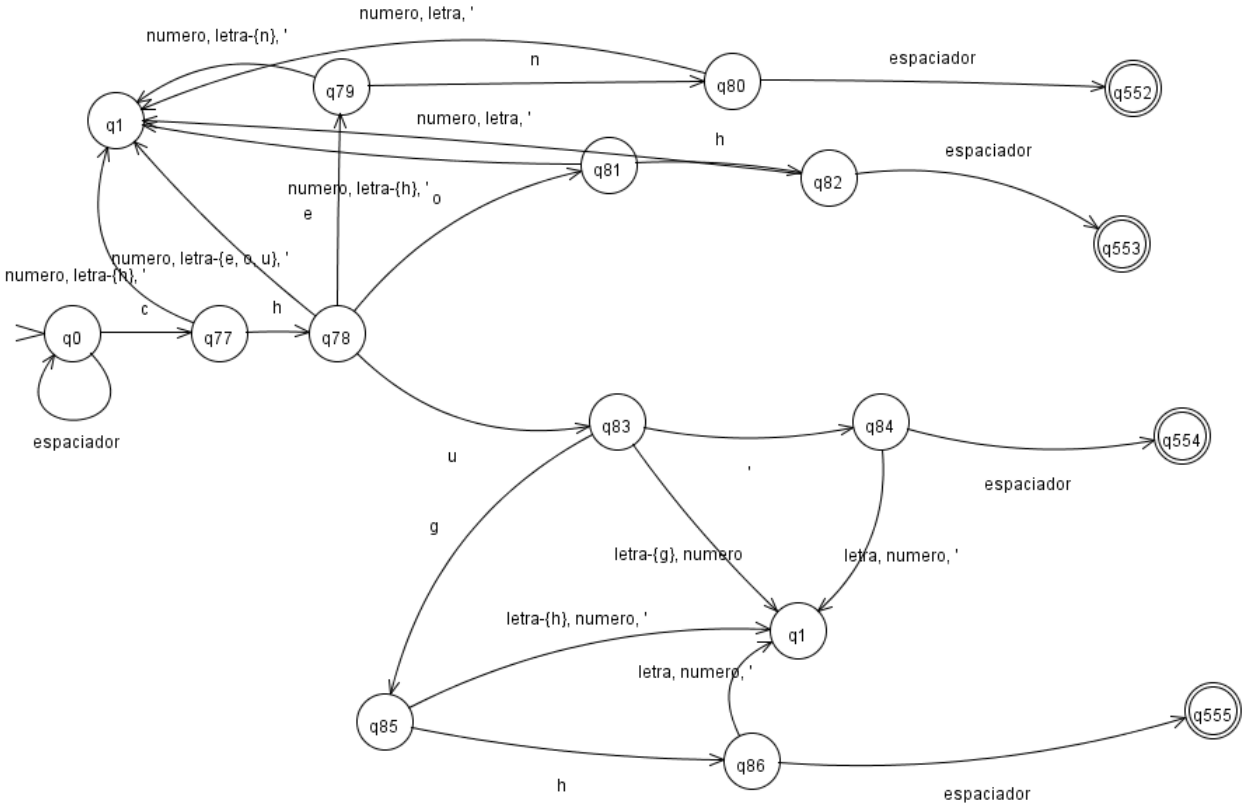


Figure 8: Estado 500

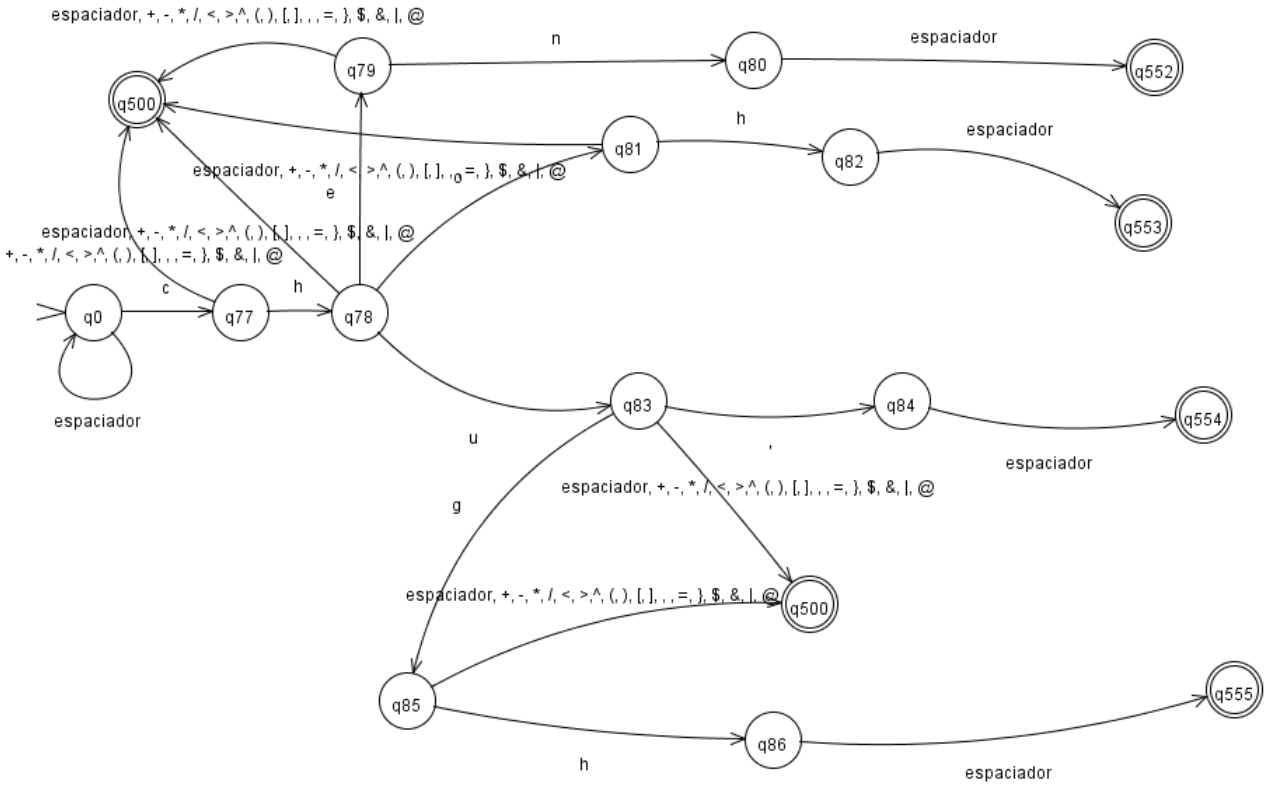


Figure 9: Error 800

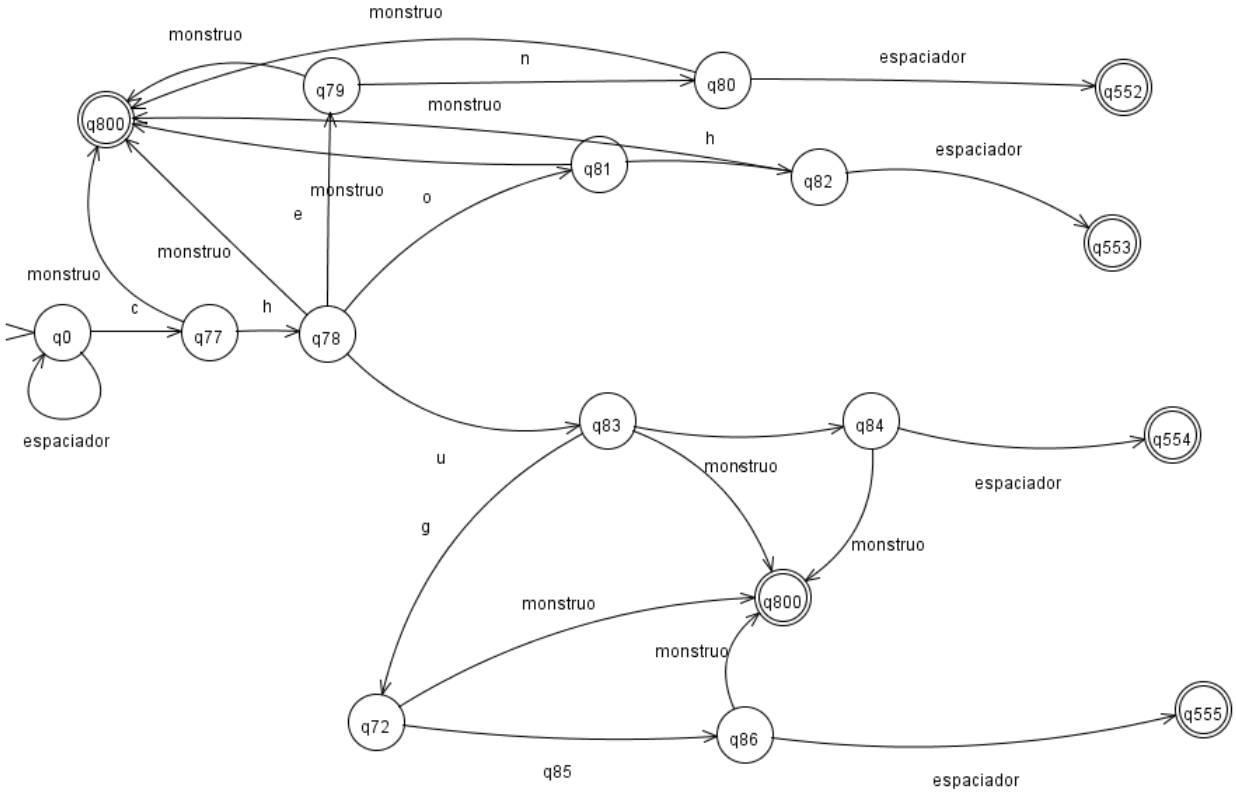
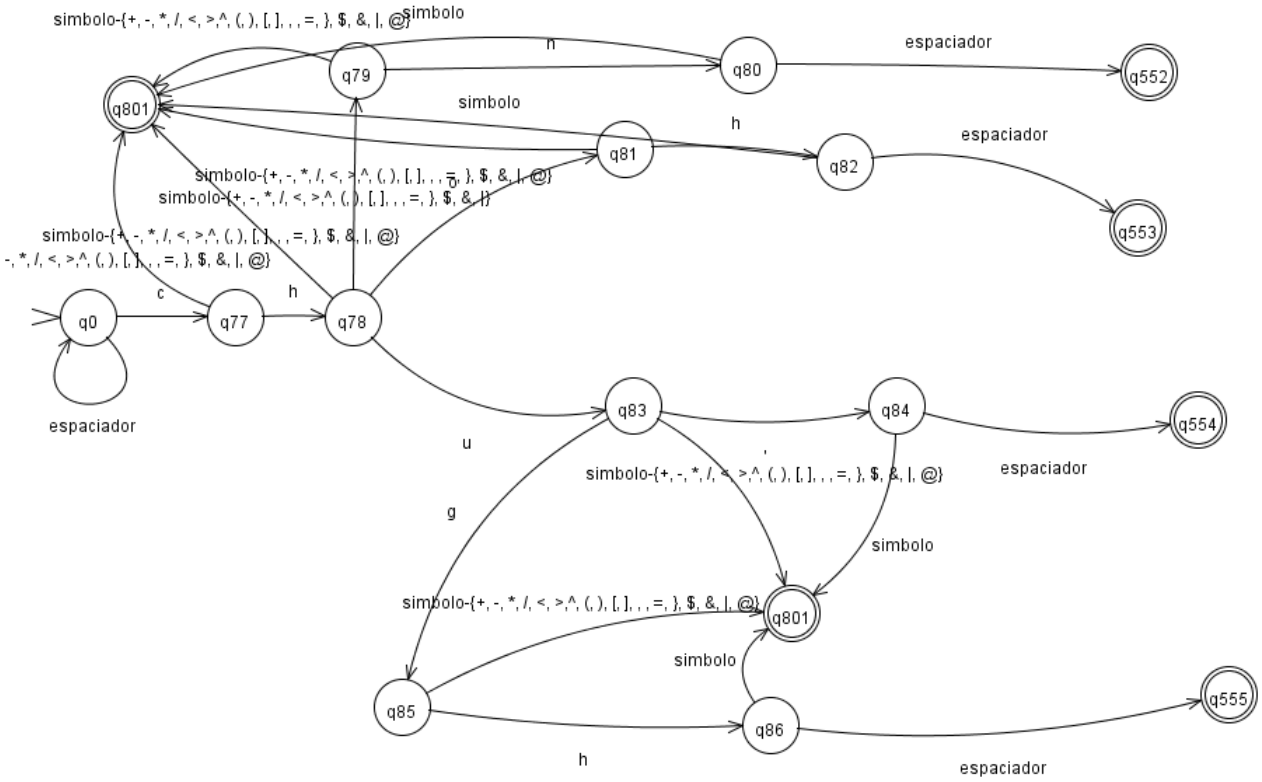


Figure 10: Error 801



5.3.30 Reconocedor de d, estado q87-q92

Figure 11: Estado 1

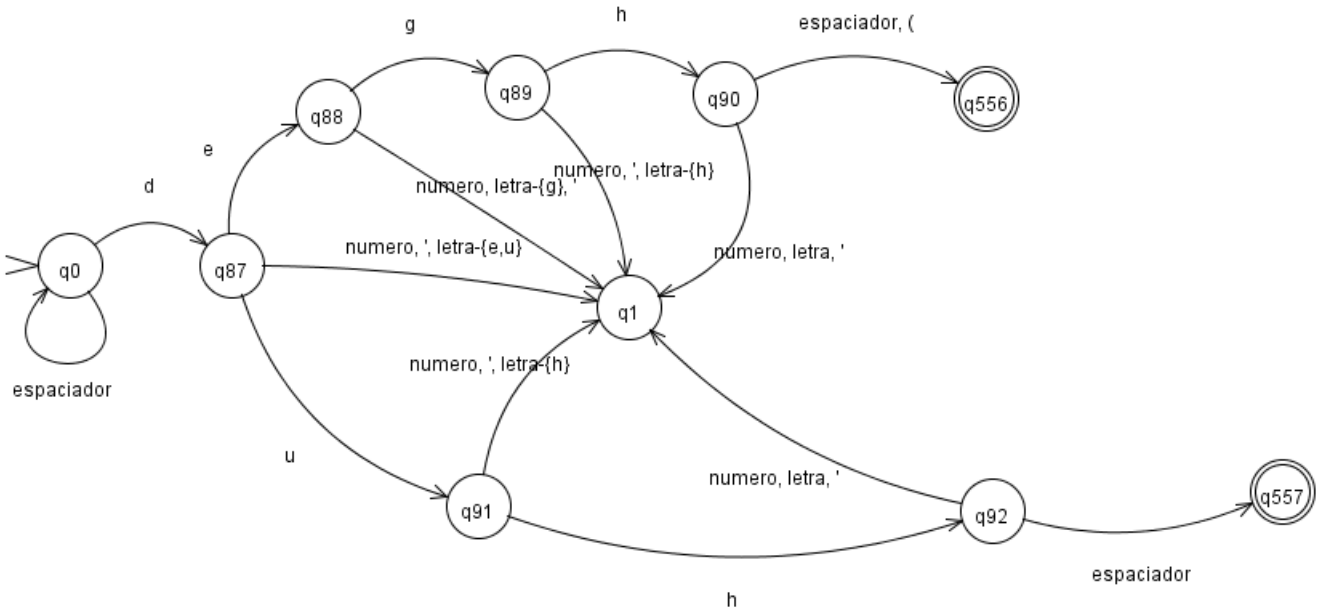


Figure 12: Estado 500

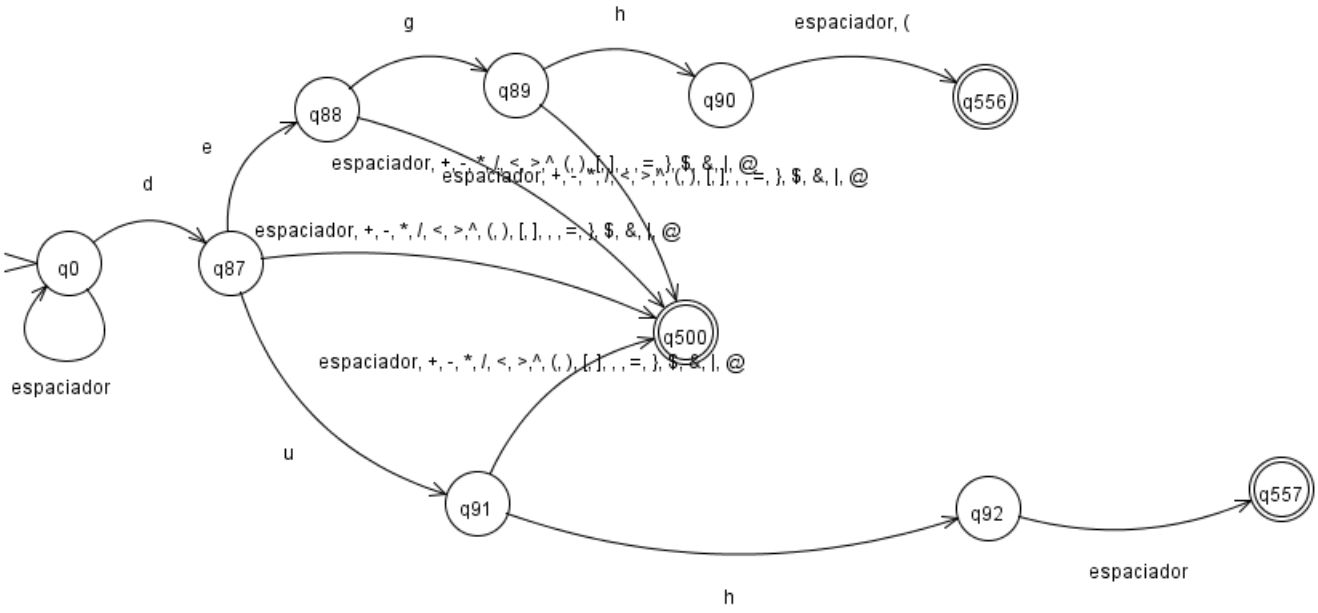


Figure 13: Error 800

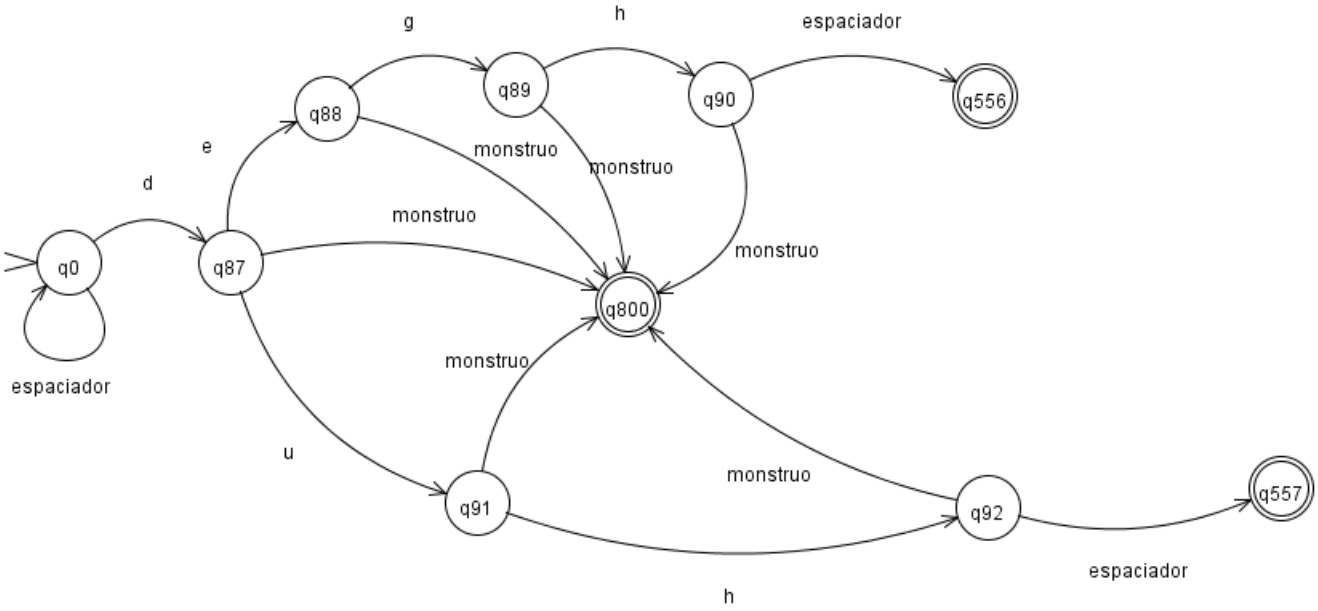
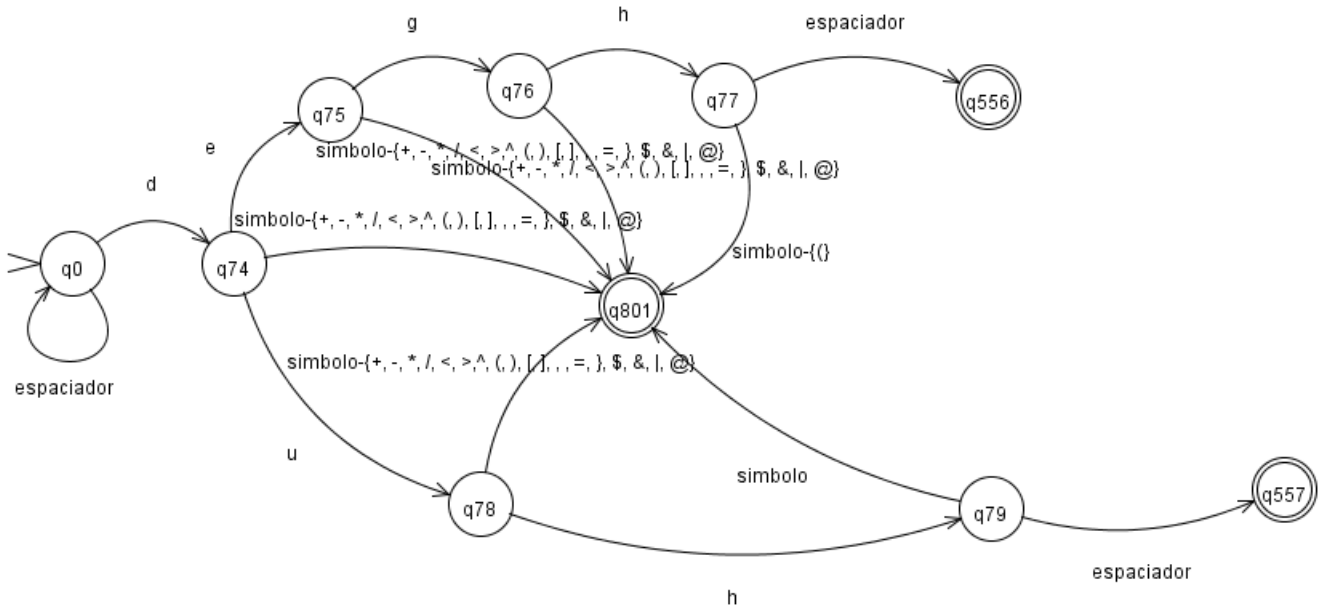


Figure 14: Error 801



5.3.31 Reconocedor de g, estado q93-q106

Figure 15: Estado 1

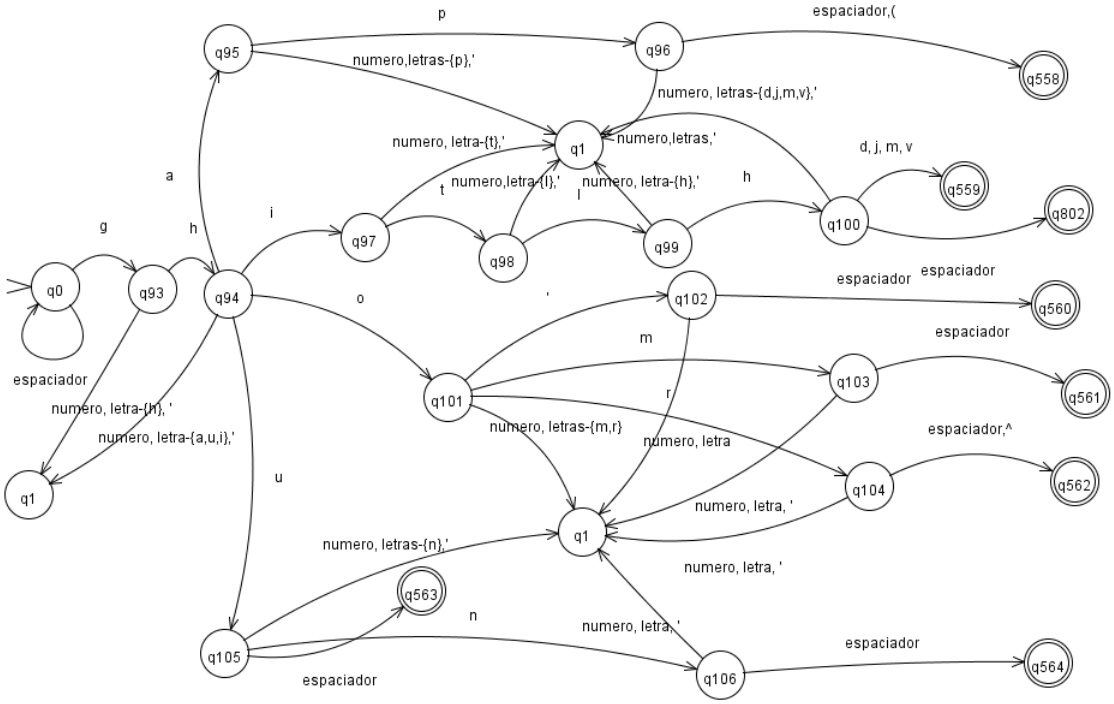


Figure 16: Estado 500

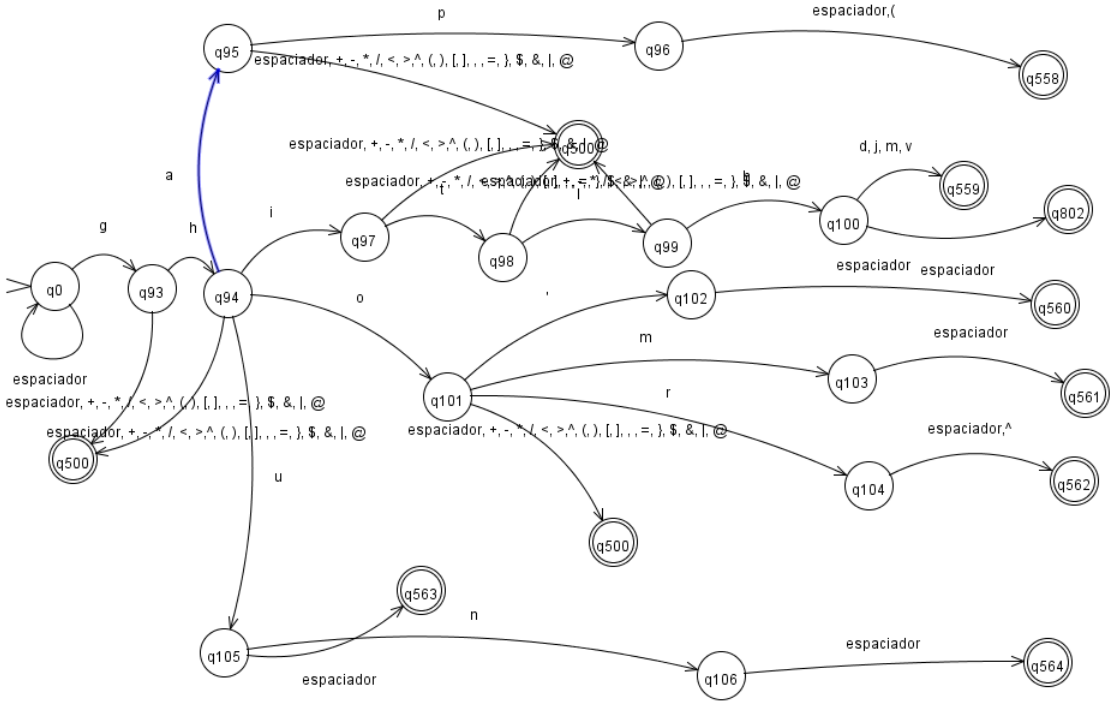


Figure 17: Error 800

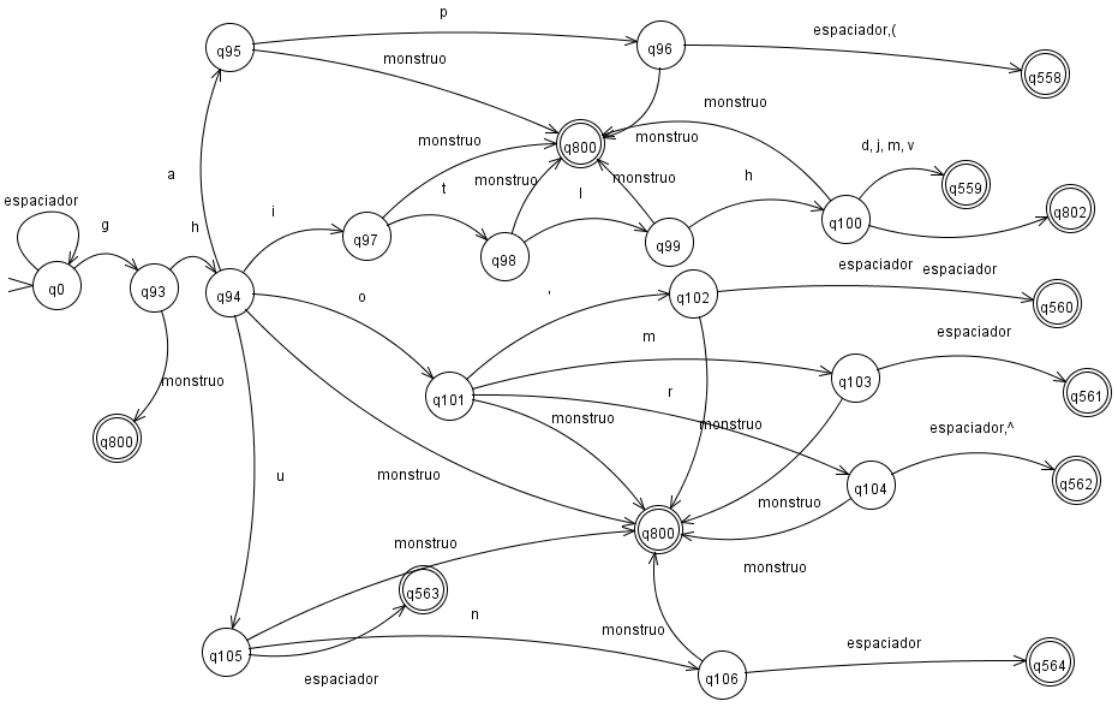
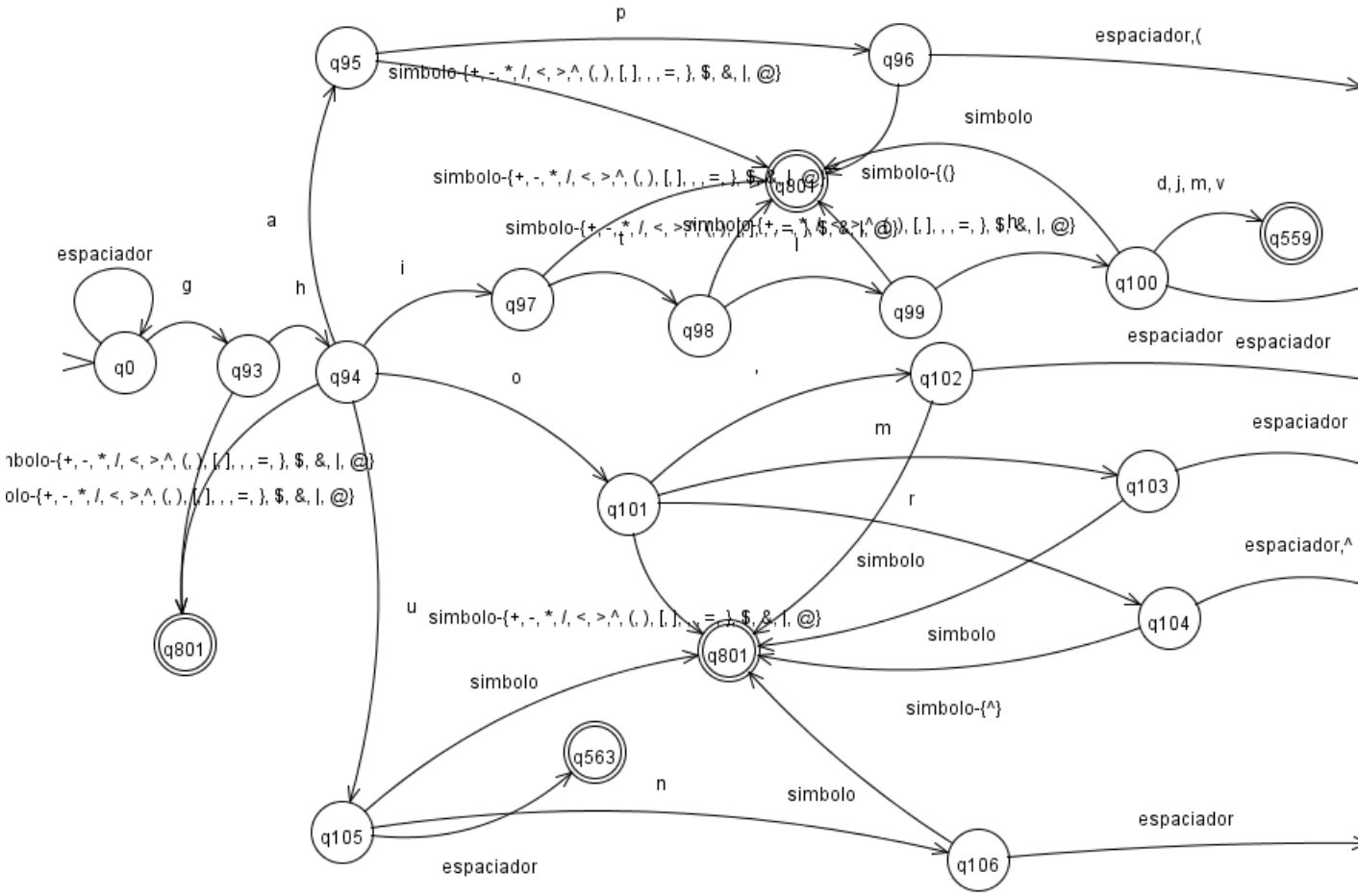


Figure 18: Error 801



5.3.32 Reconocedor de h, estado q107-q117

Figure 19: Estado 1

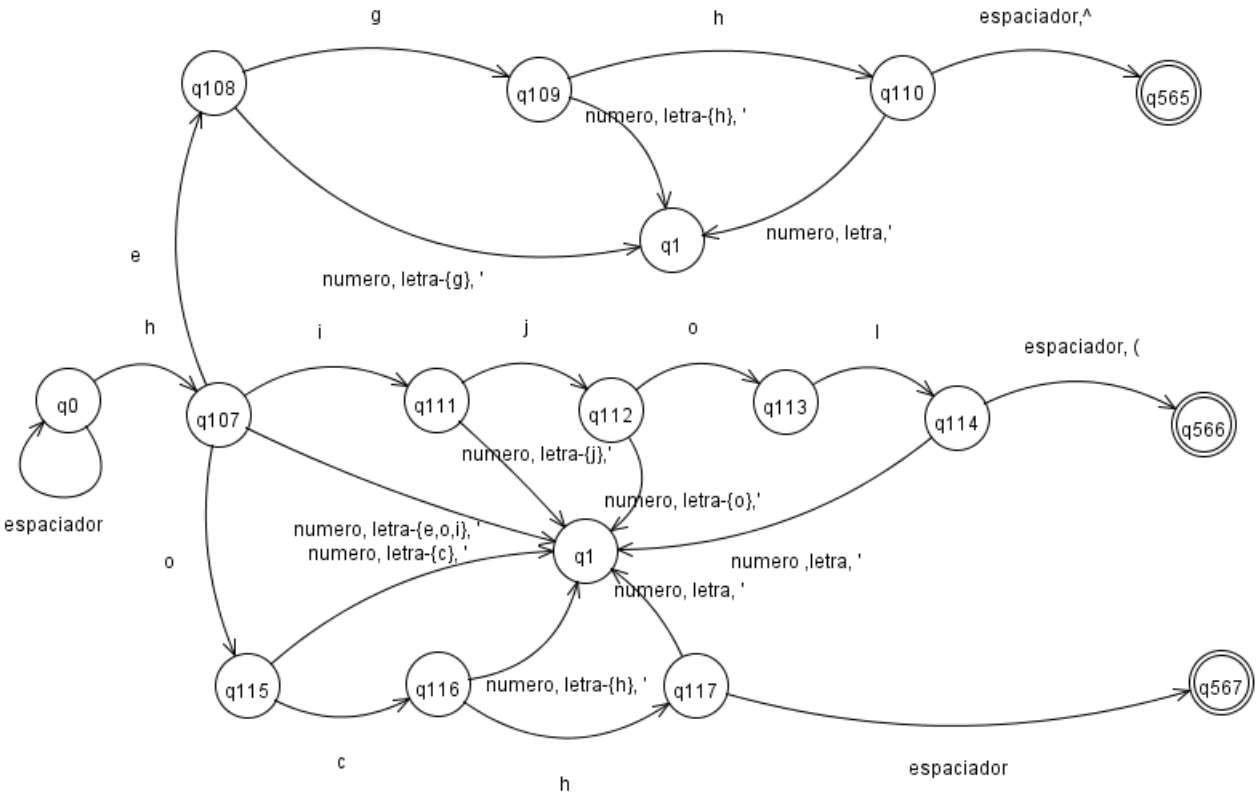


Figure 20: Estado 500

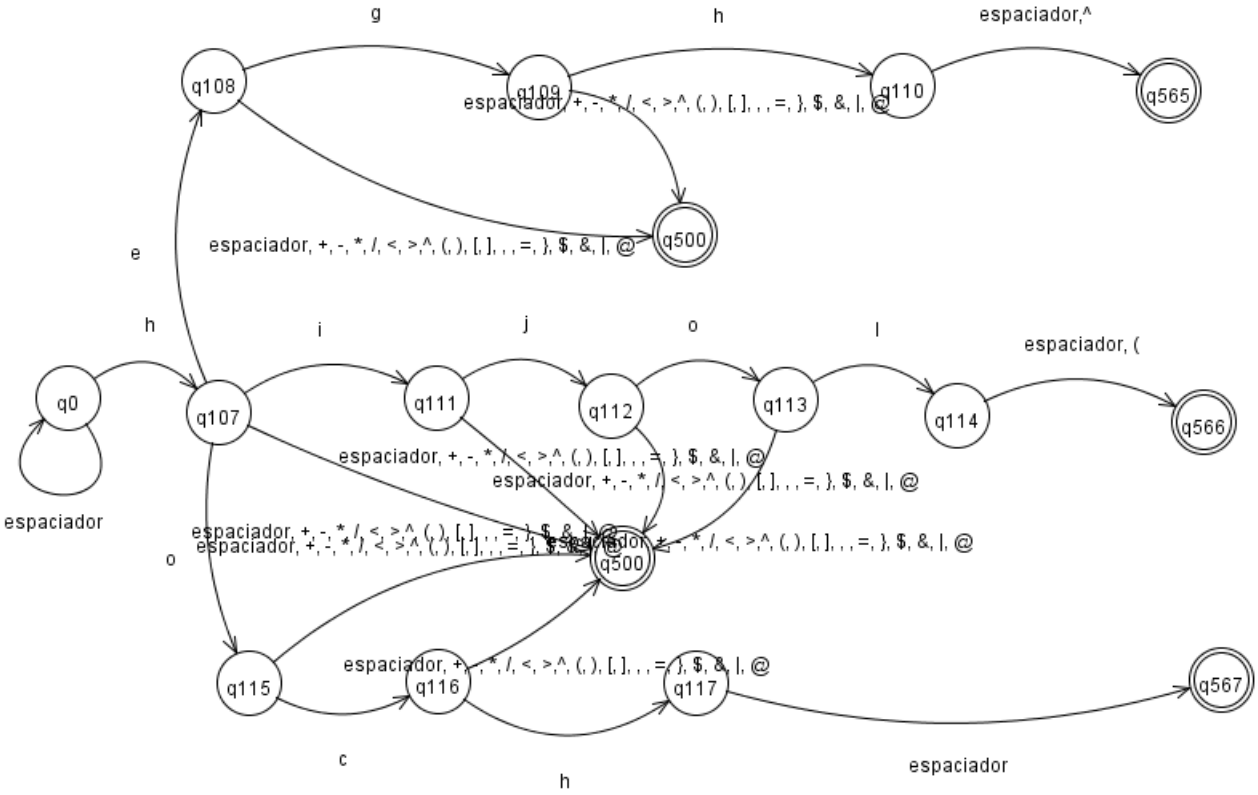


Figure 21: Error 800

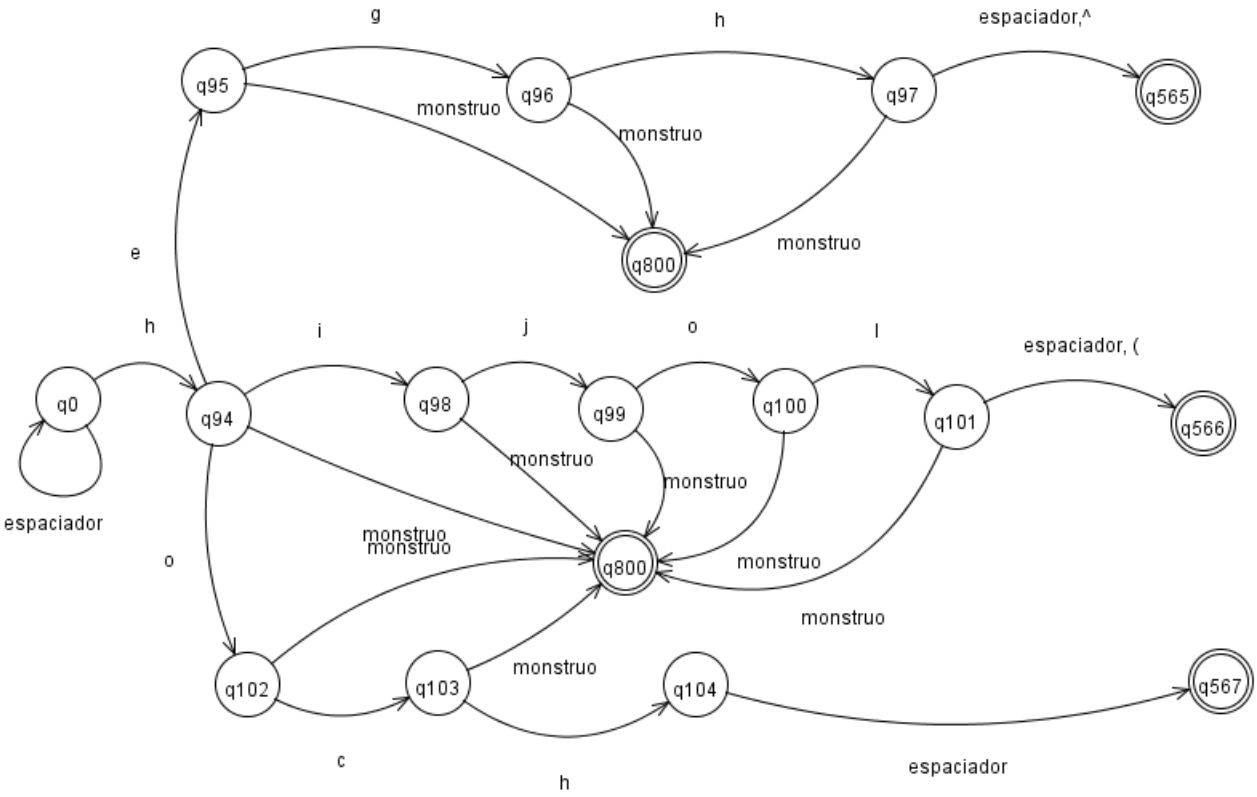
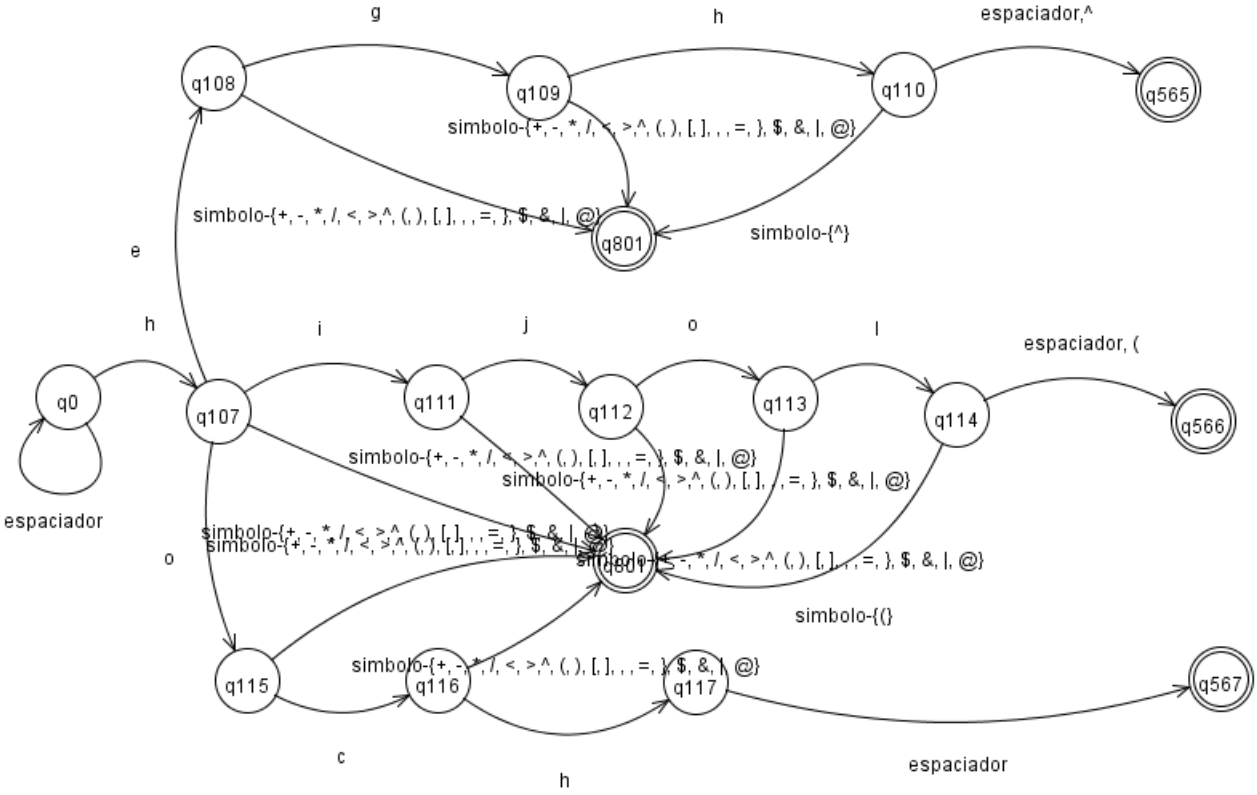


Figure 22: Error 801



5.3.33 Reconocedor de j, estado q118-q122

Figure 23: Estado 1

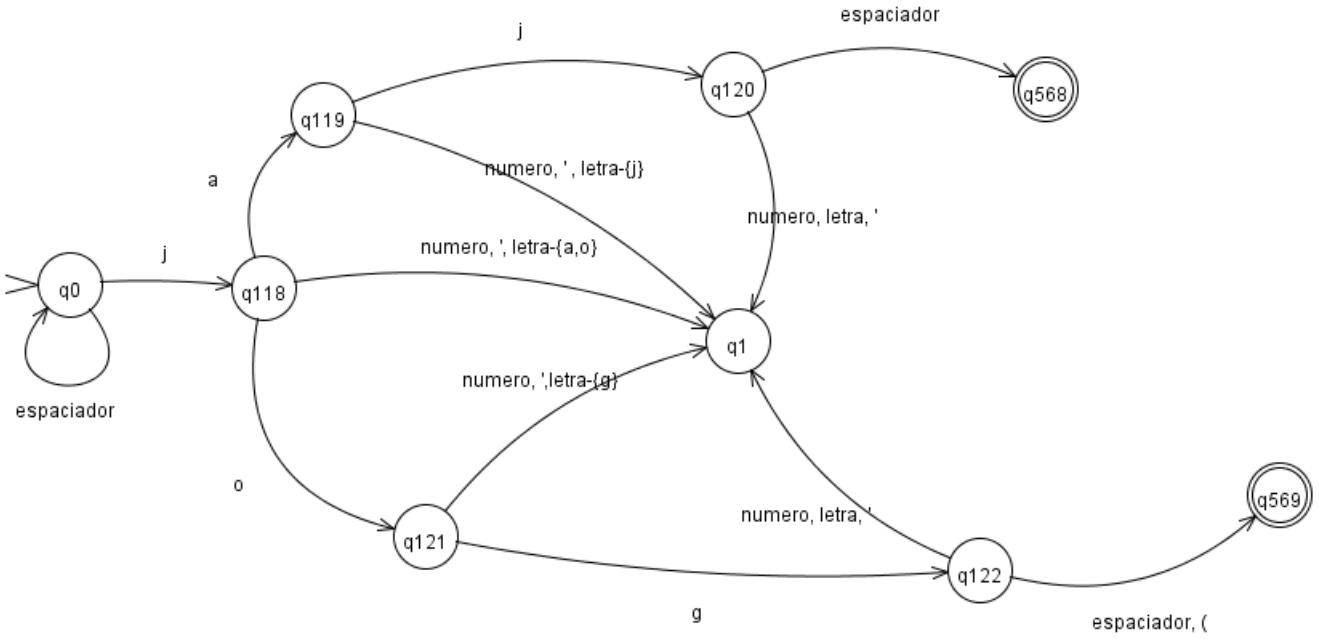


Figure 24: Estado 500

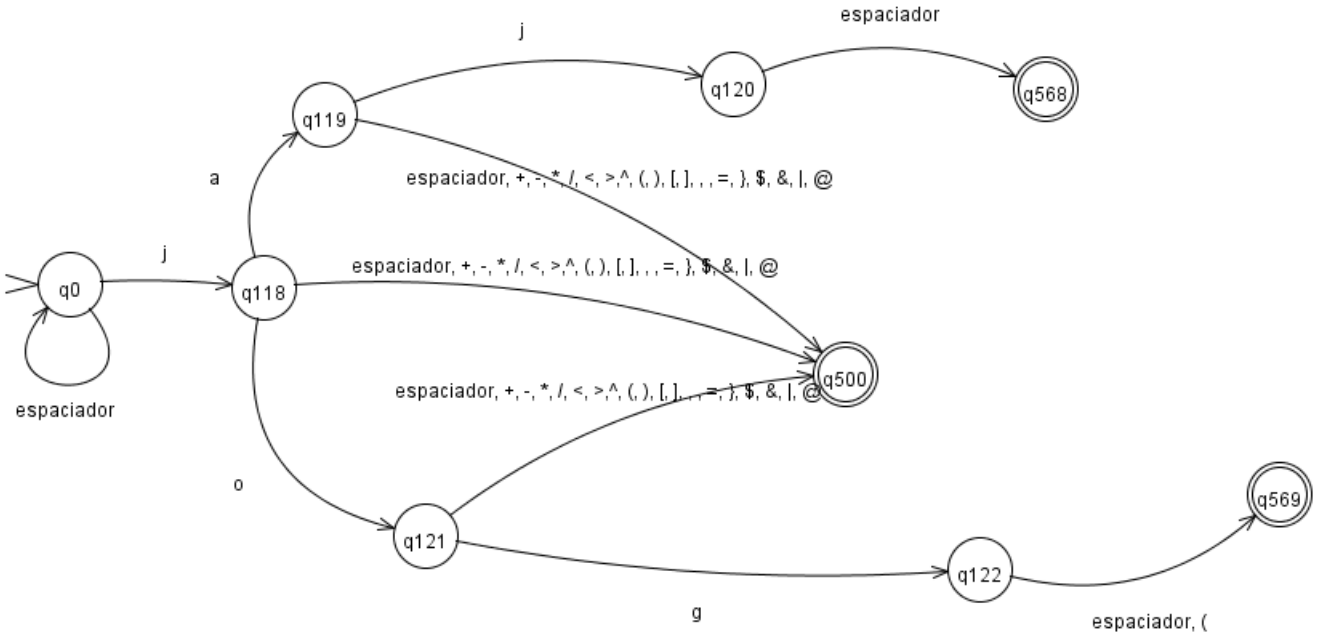


Figure 25: Error 800

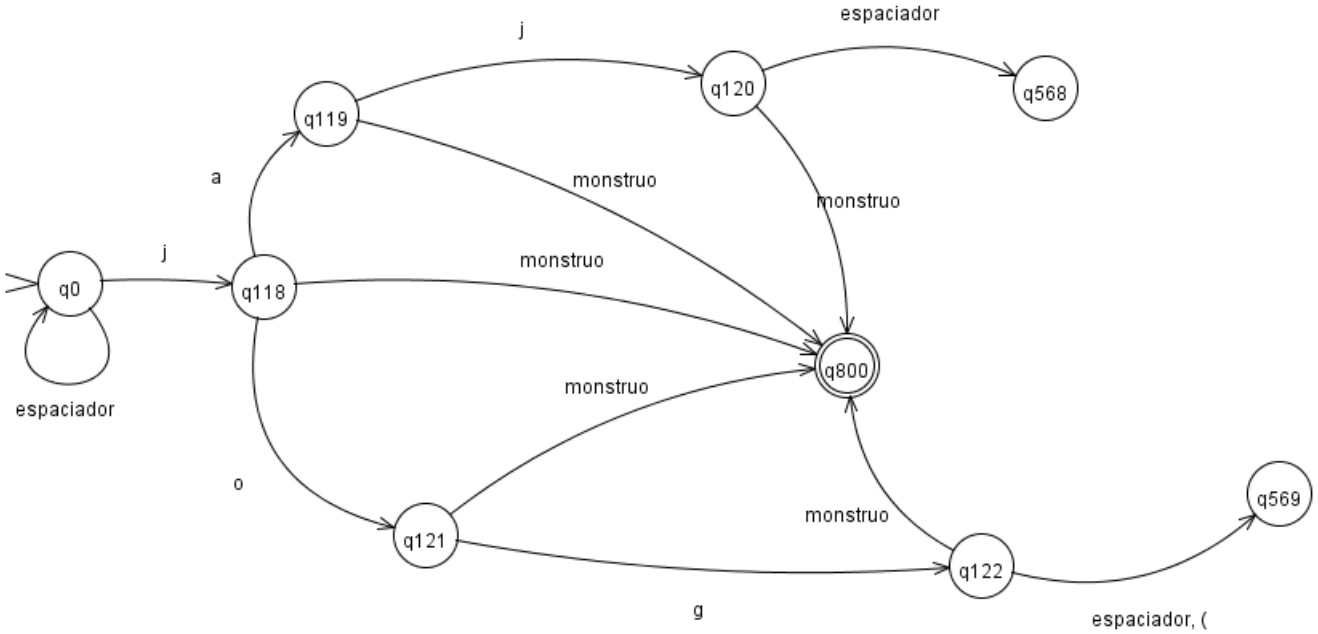
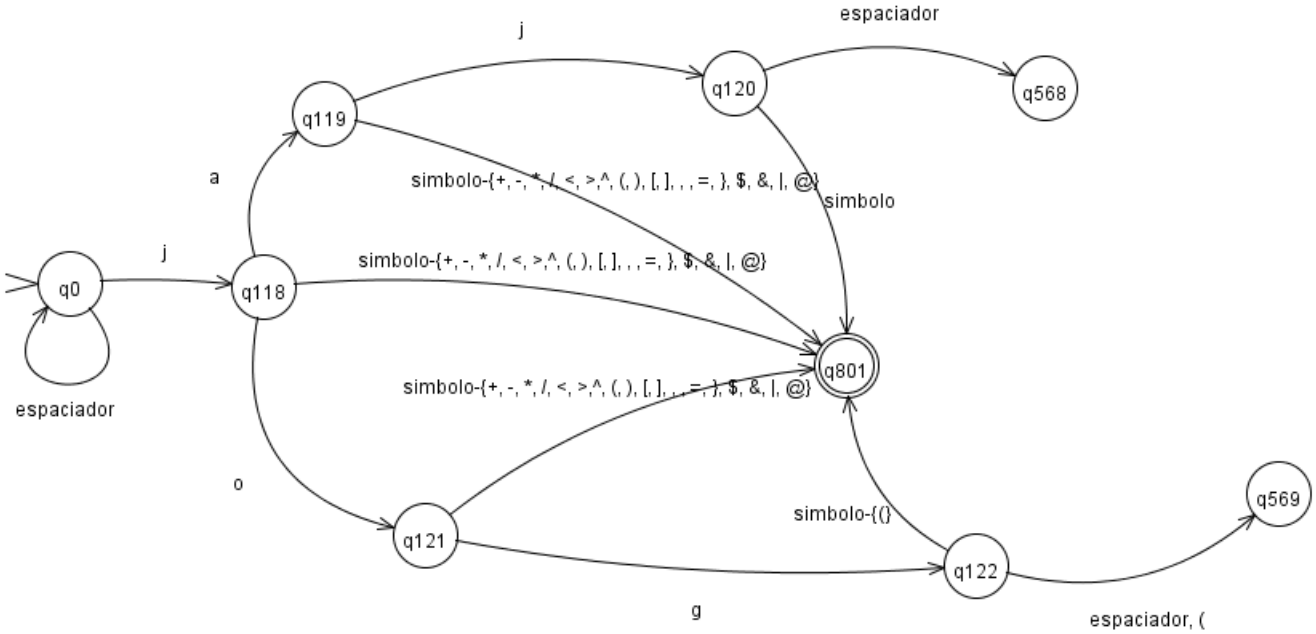


Figure 26: Error 801



5.3.34 Reconocedor de l, estado q123-q132

Figure 27: Estado 1

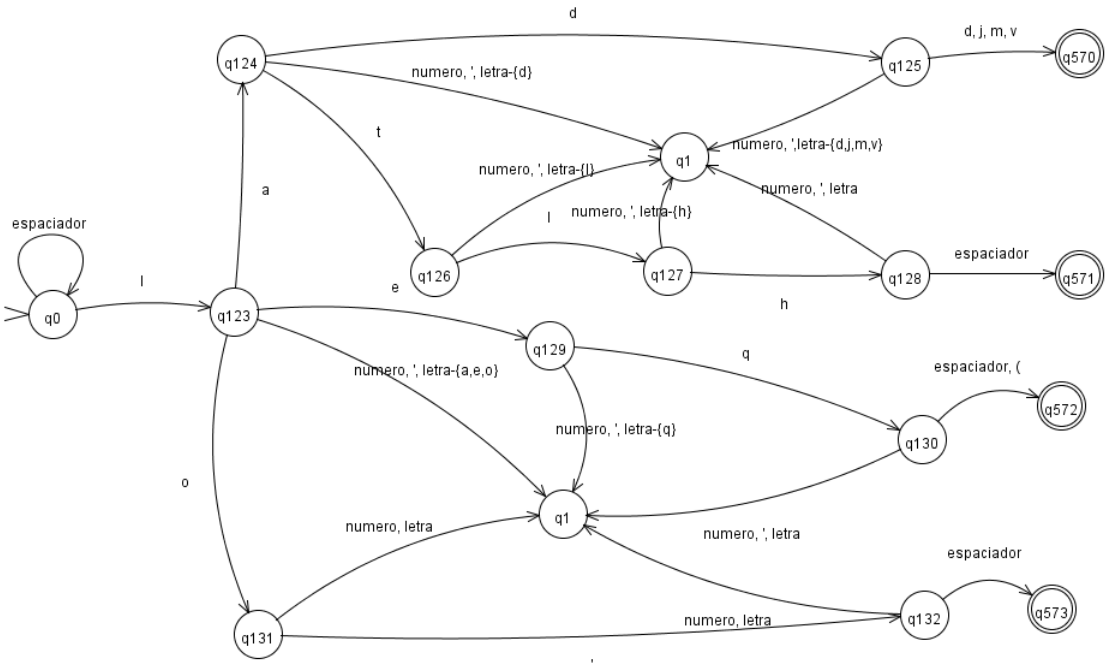


Figure 28: Estado 500

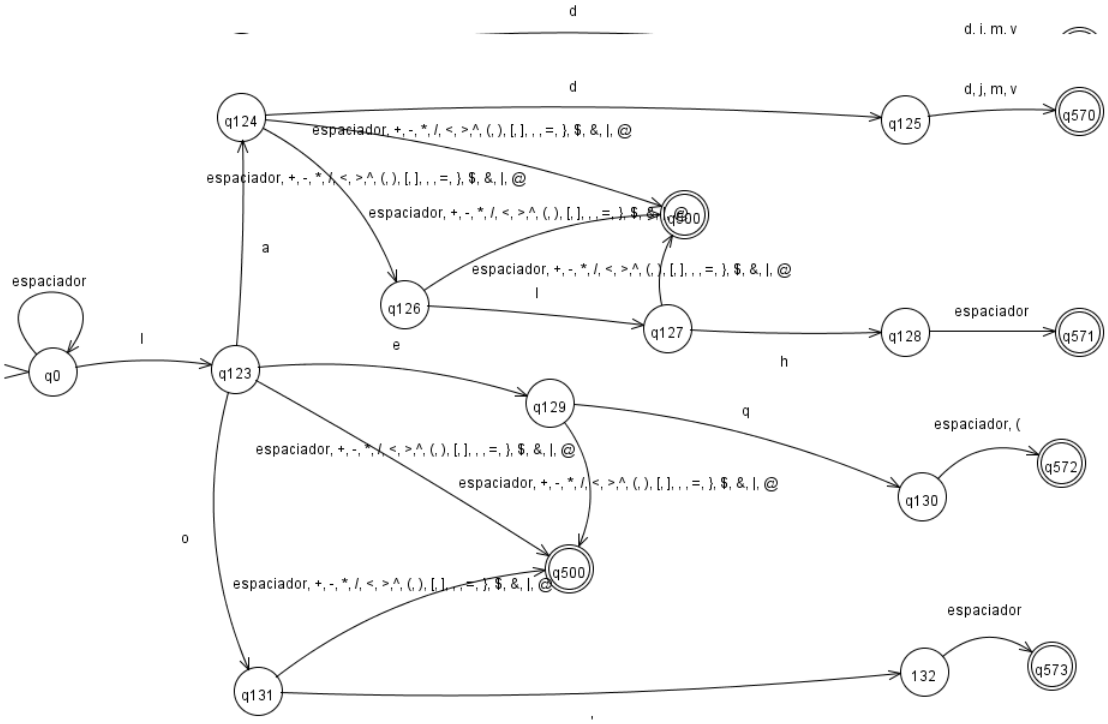


Figure 29: Error 800

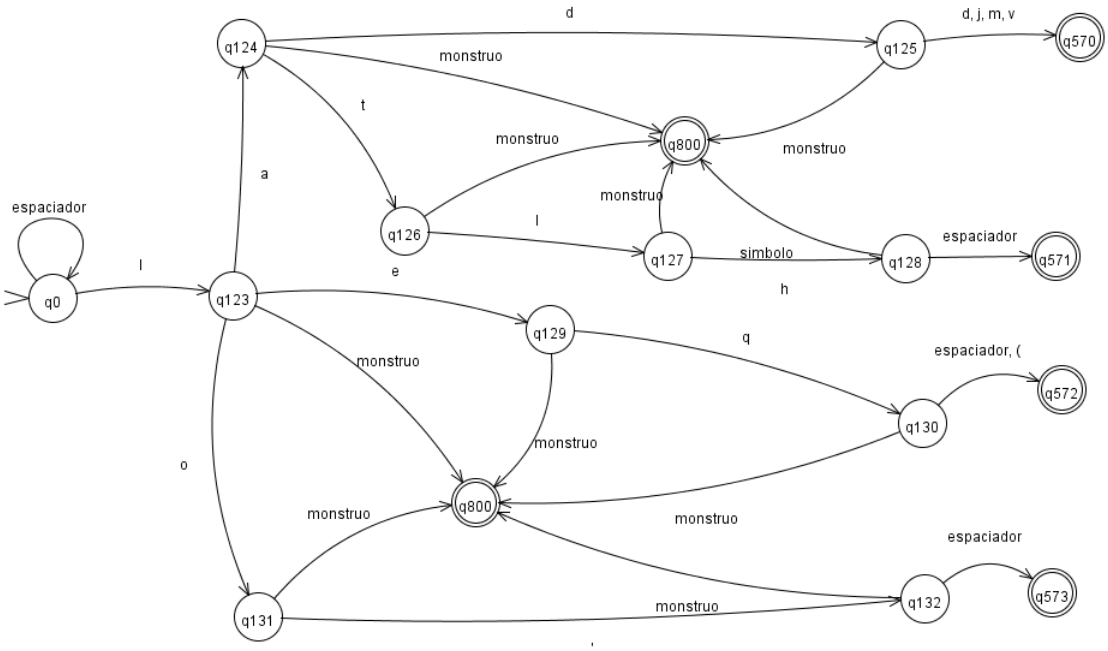
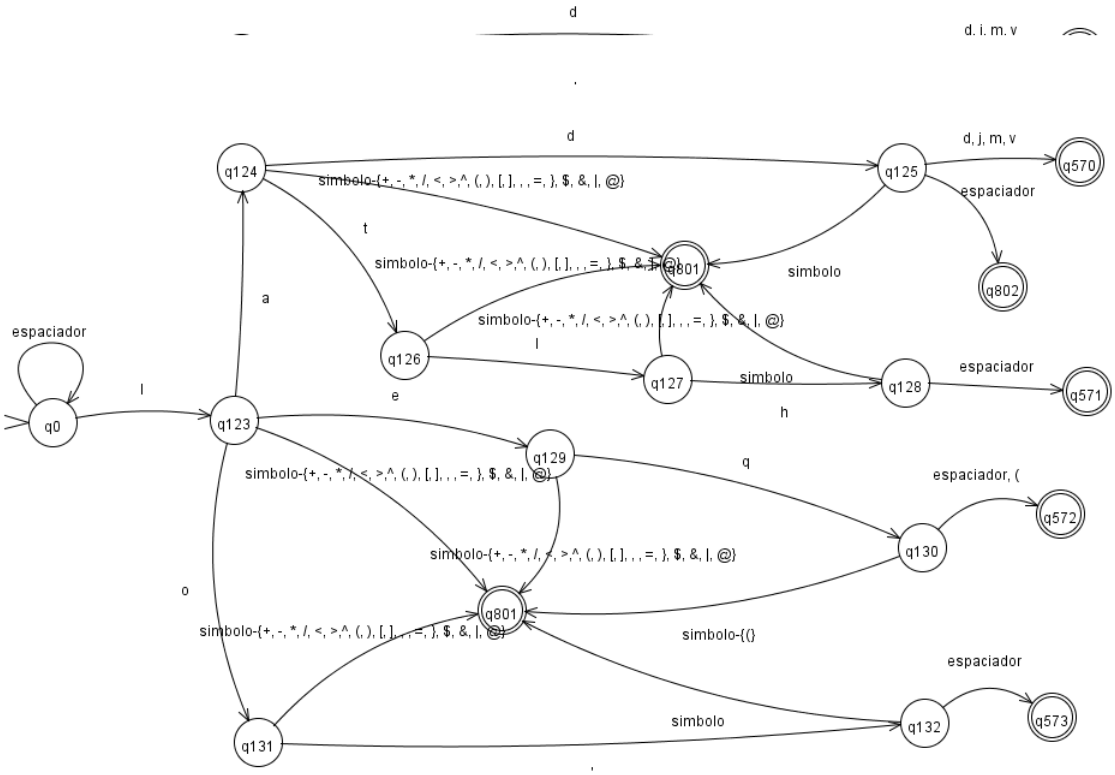


Figure 30: Error 801



5.3.35 Reconocedor de m, estado q133-q145

Figure 31: Estado 1

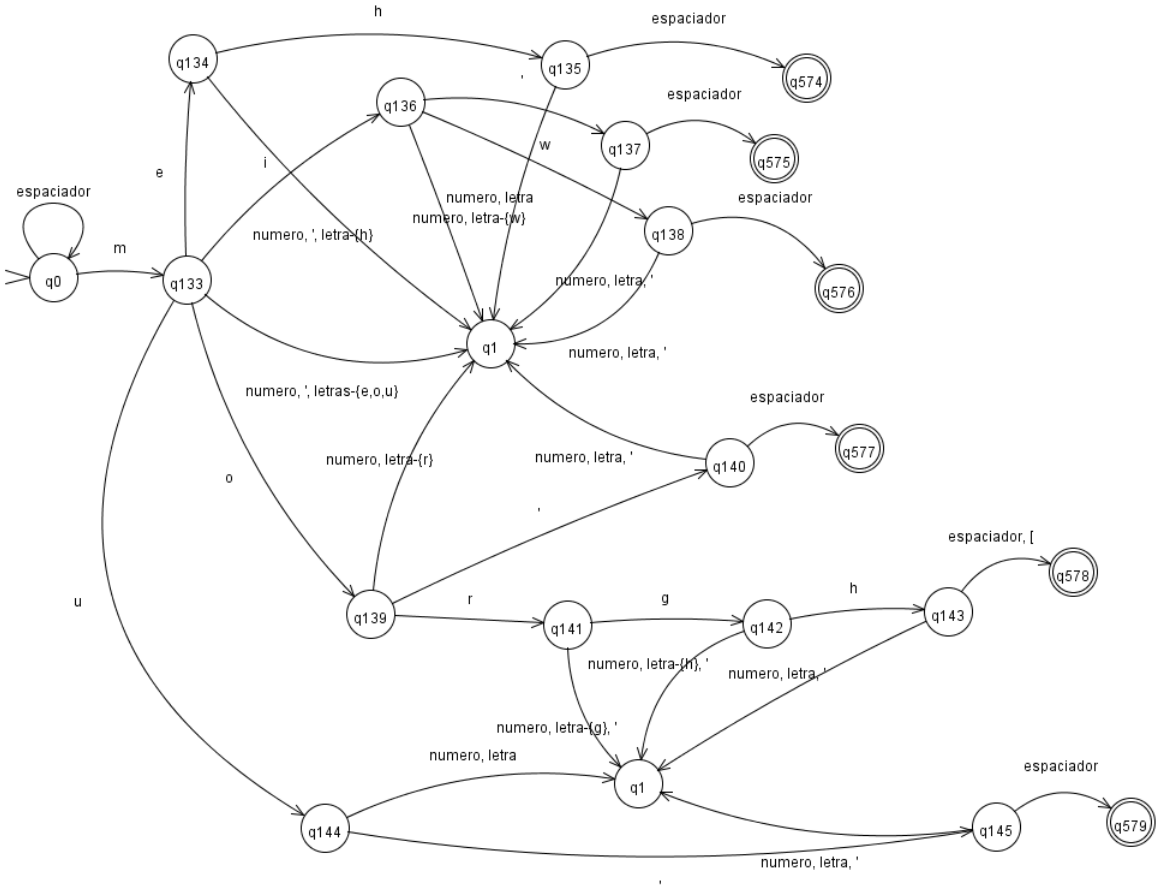


Figure 32: Estado 500

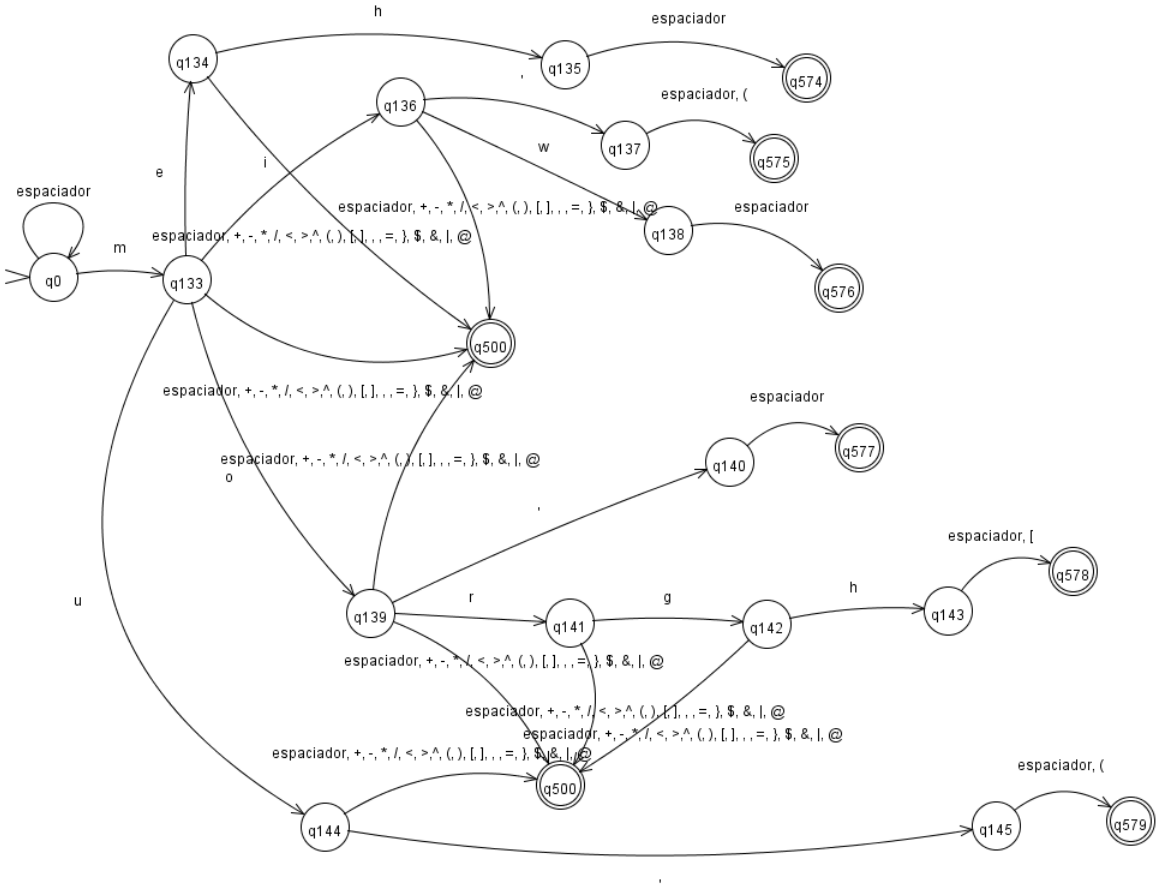


Figure 33: Error 800

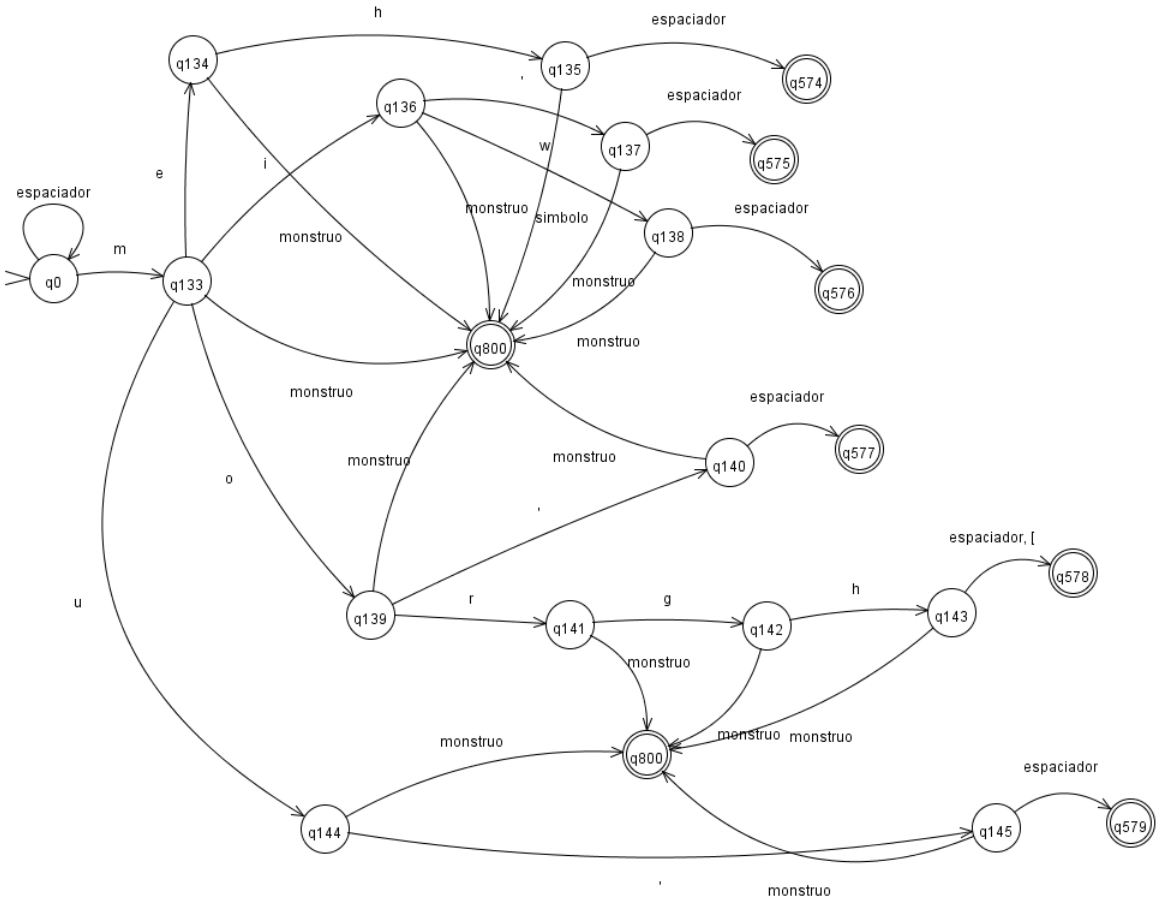
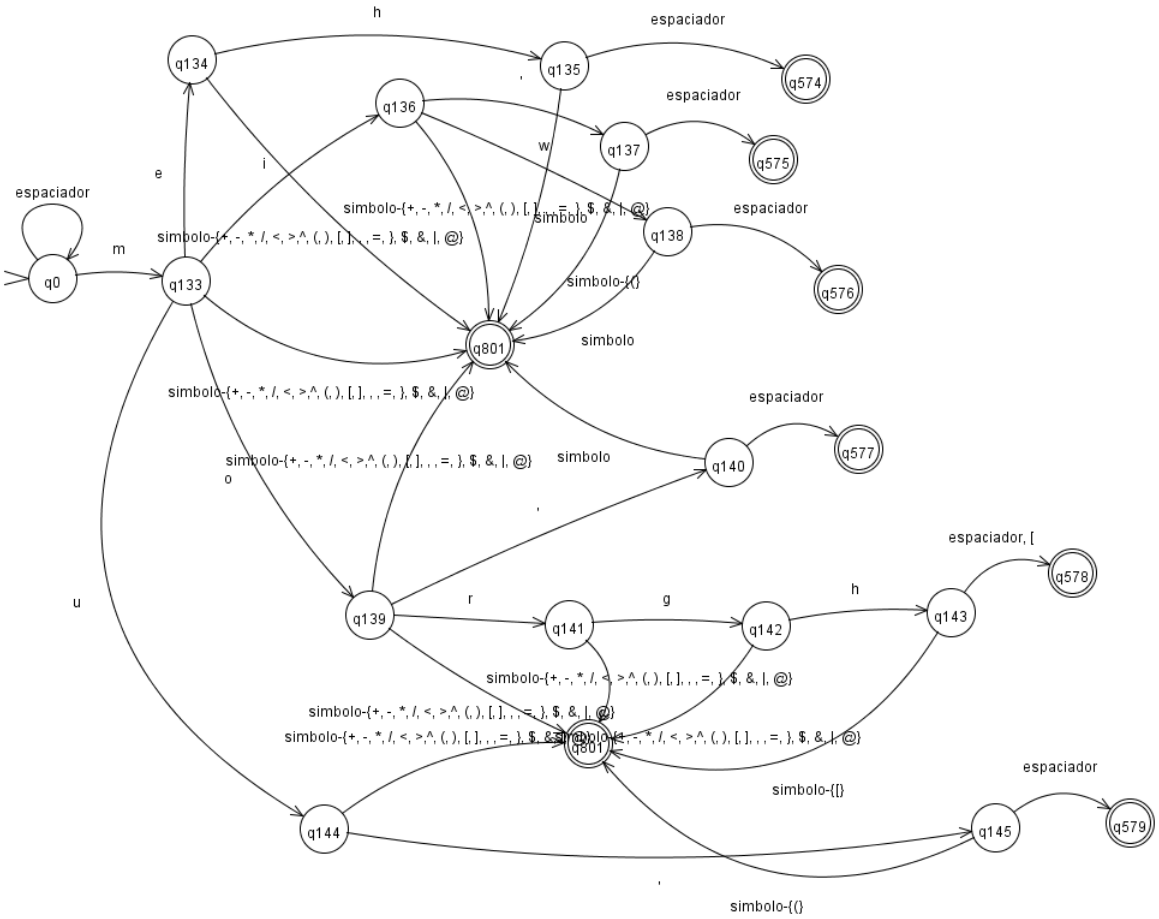


Figure 34: Error 801



5.3.36 Reconocedor de n, estado q146-q159

Figure 35: Estado 1

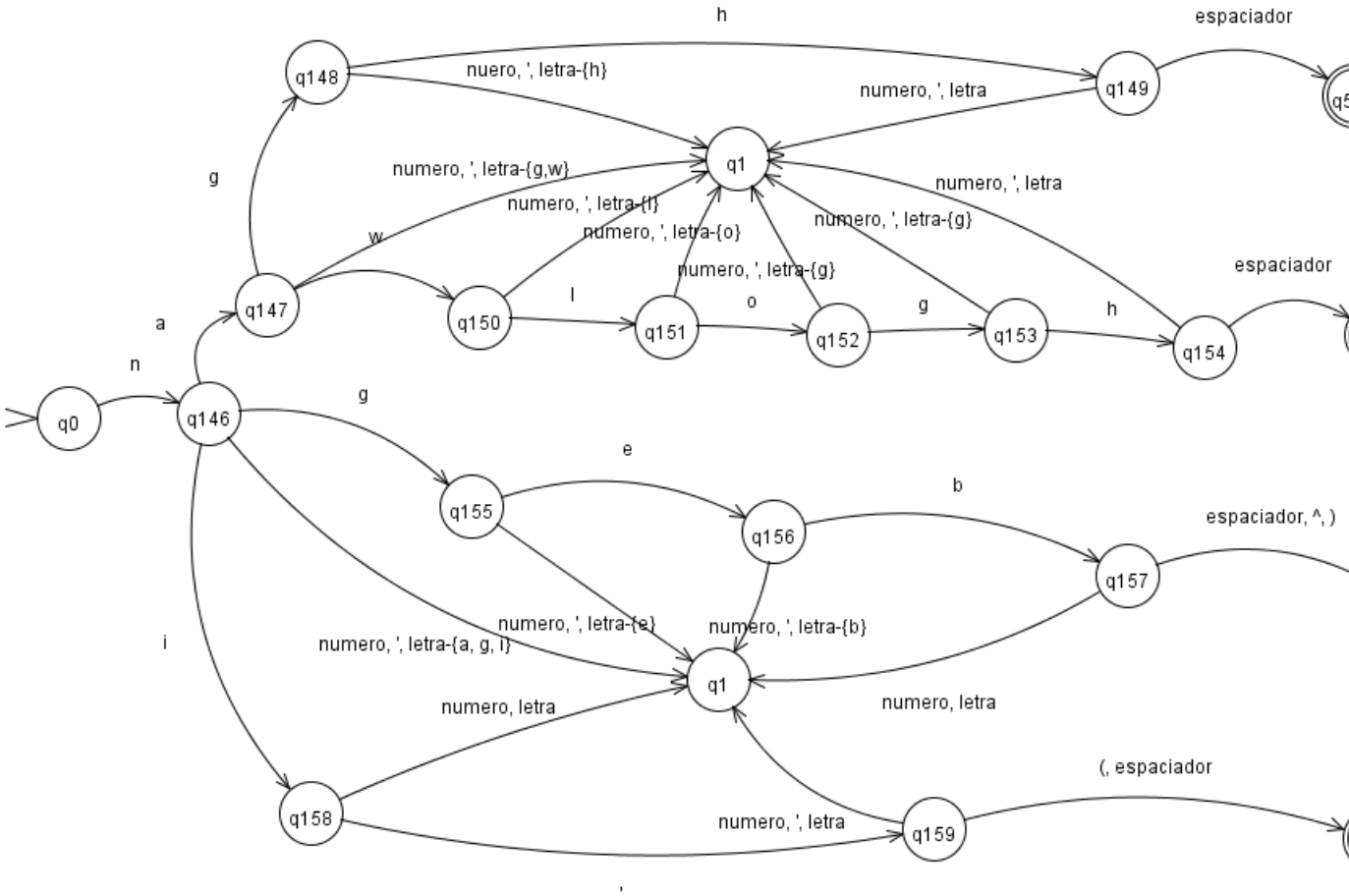


Figure 36: Estado 500

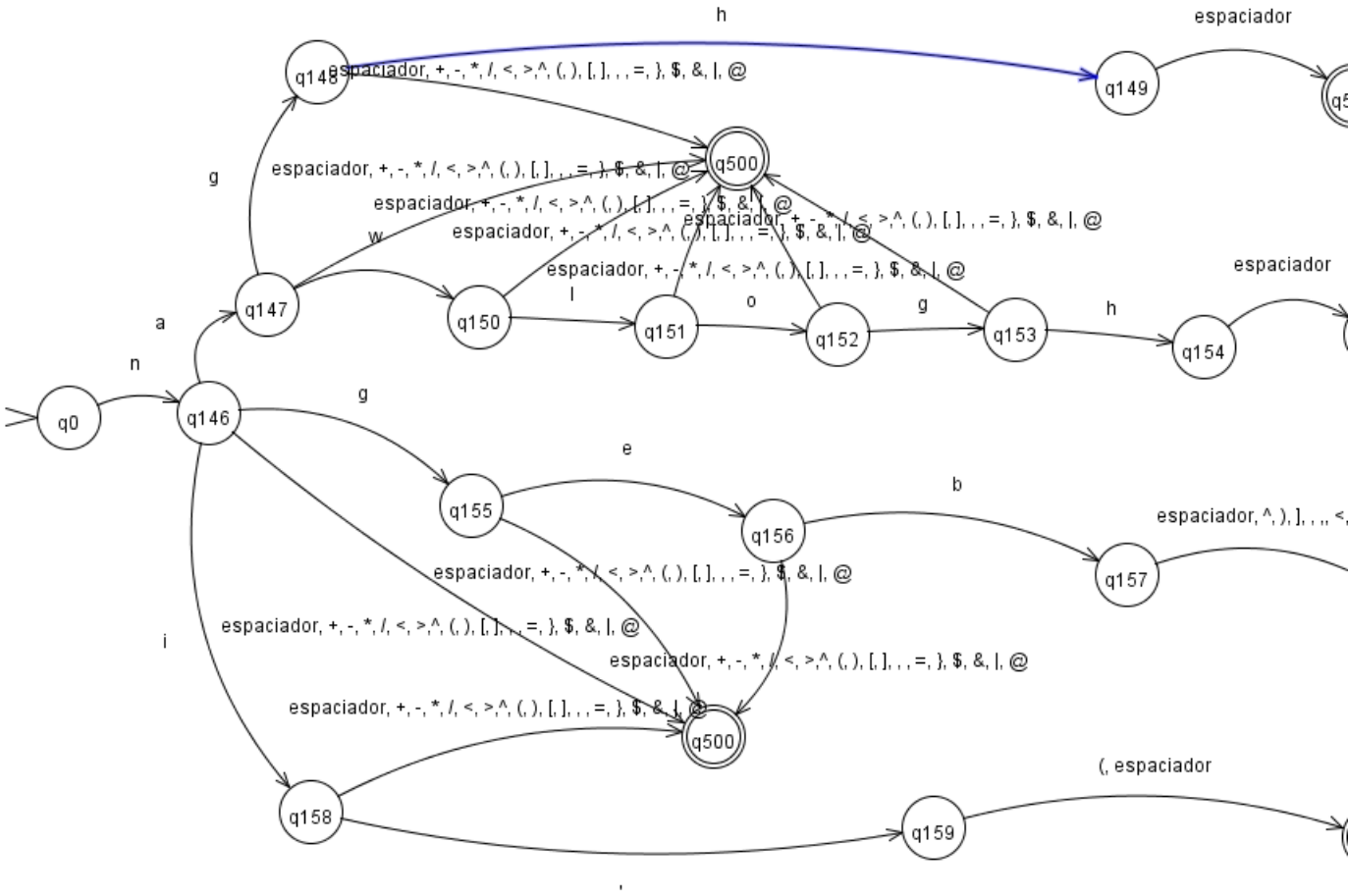


Figure 37: Error 800

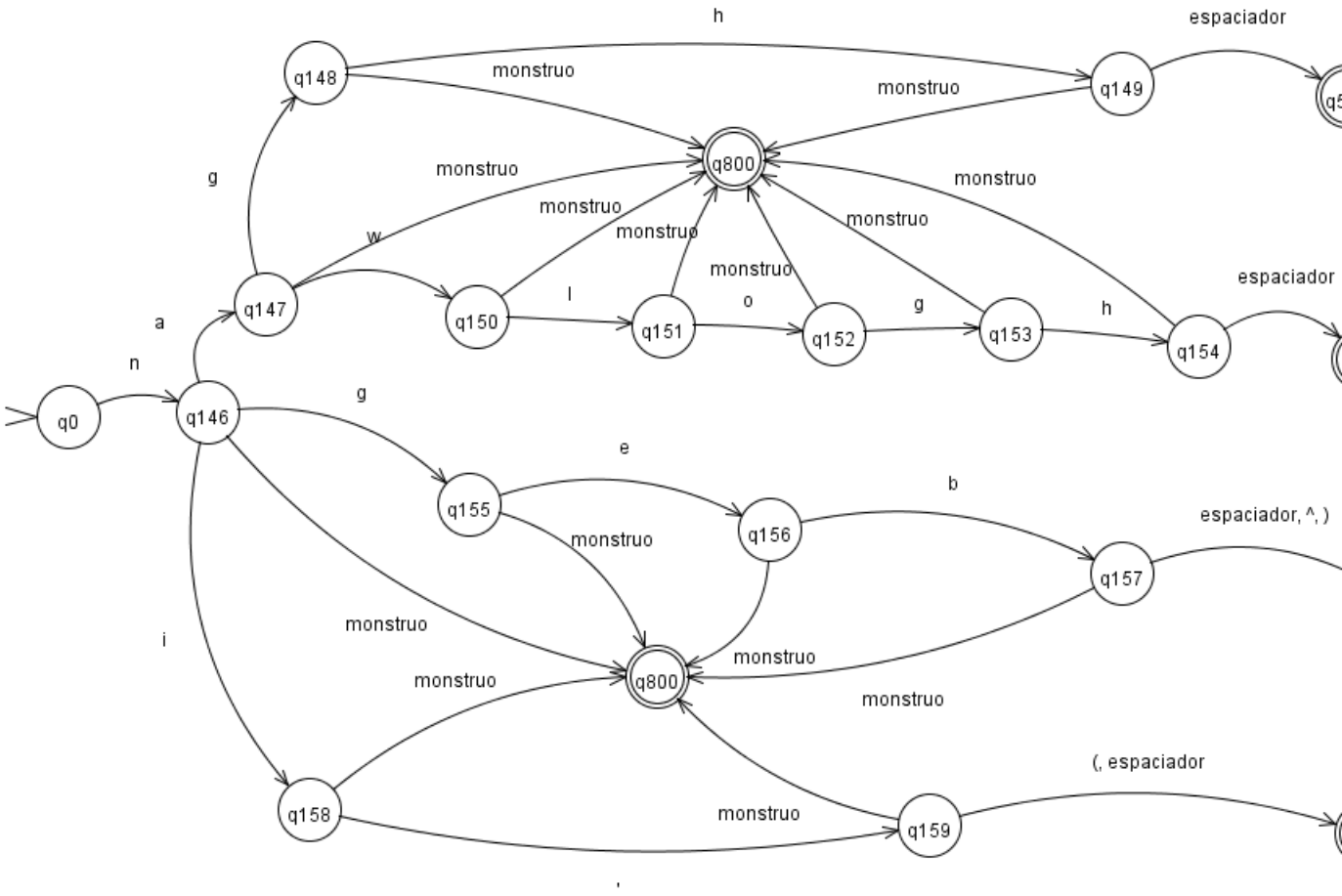
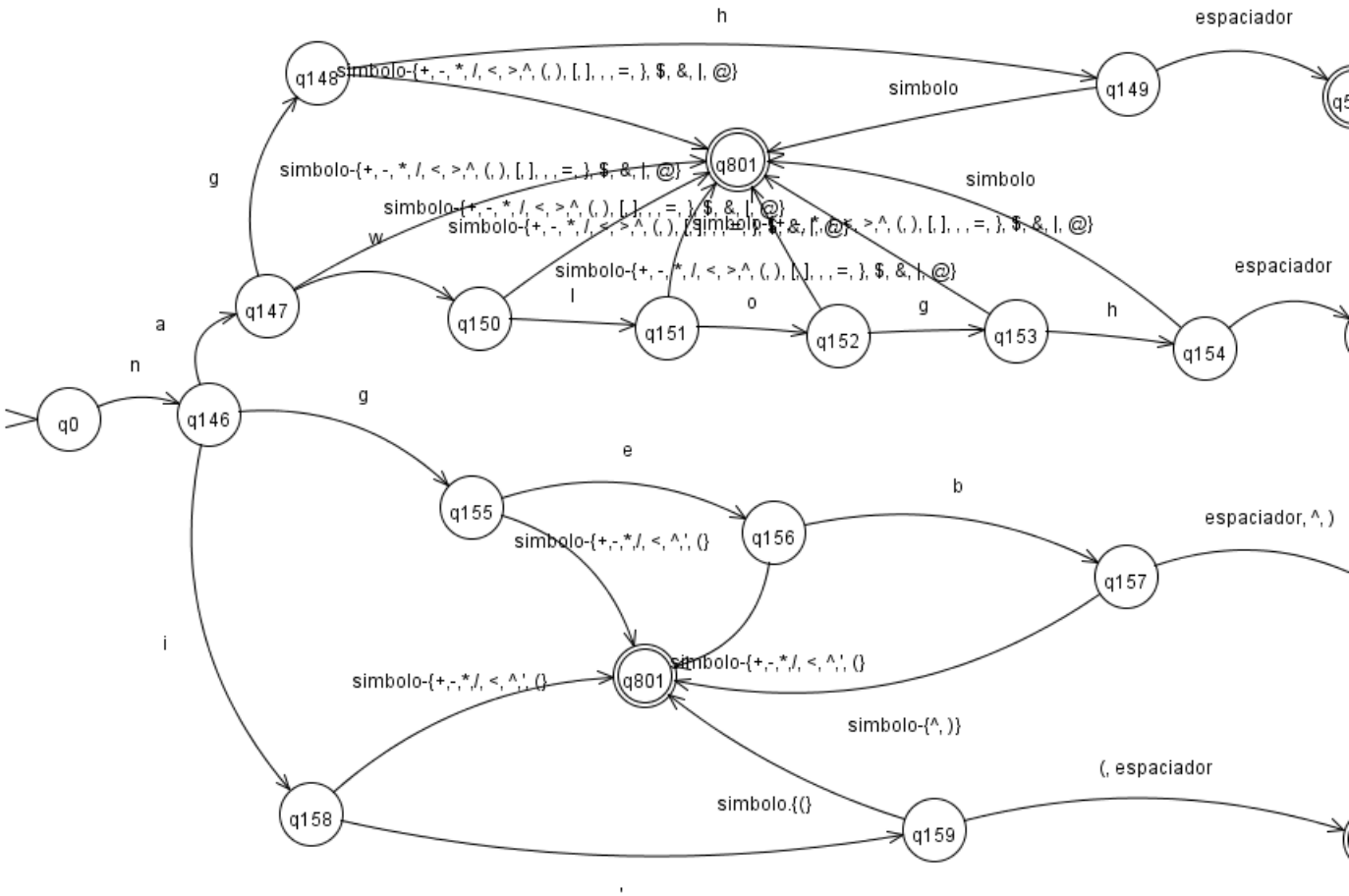


Figure 38: Error 801



5.3.37 Reconocedor de p, estado q160-q166

Figure 39: Estado 1

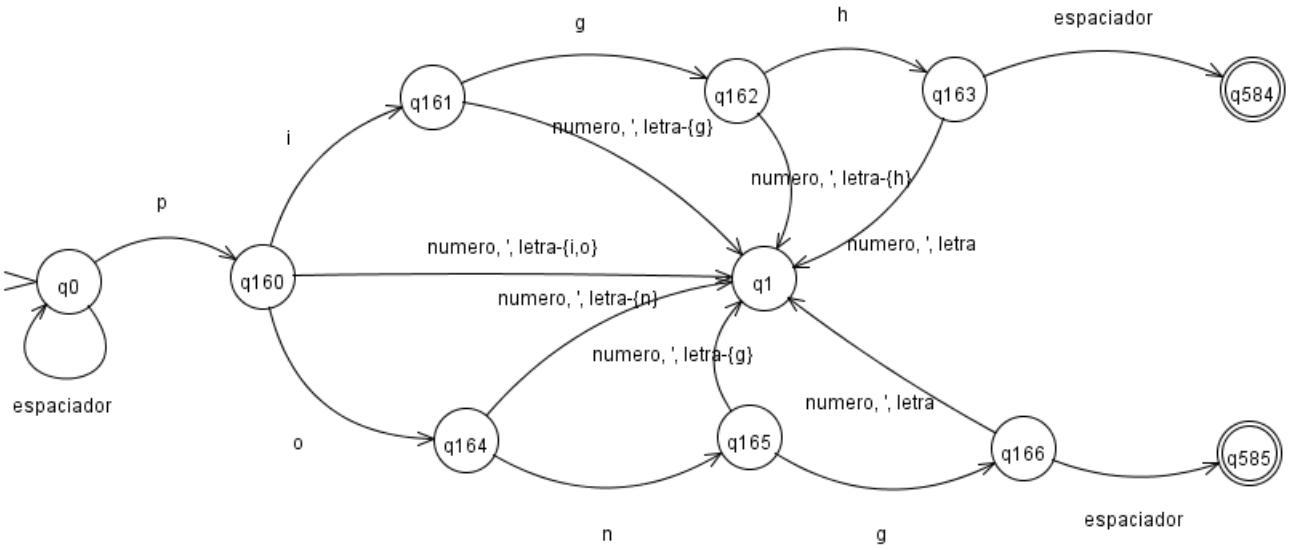


Figure 40: Estado 500

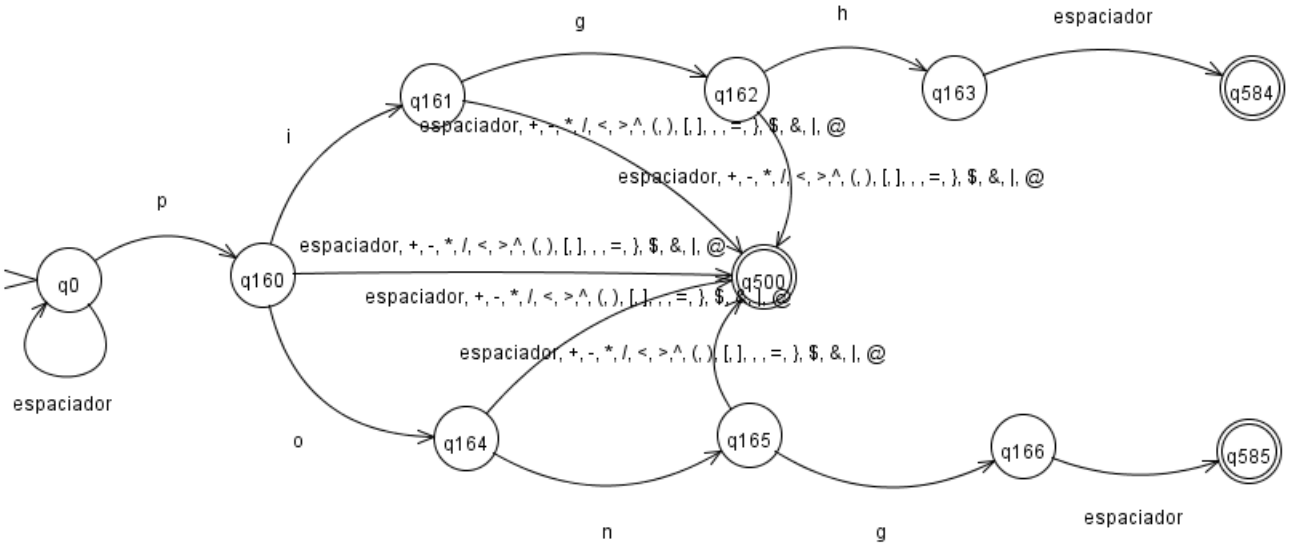


Figure 41: Error 800

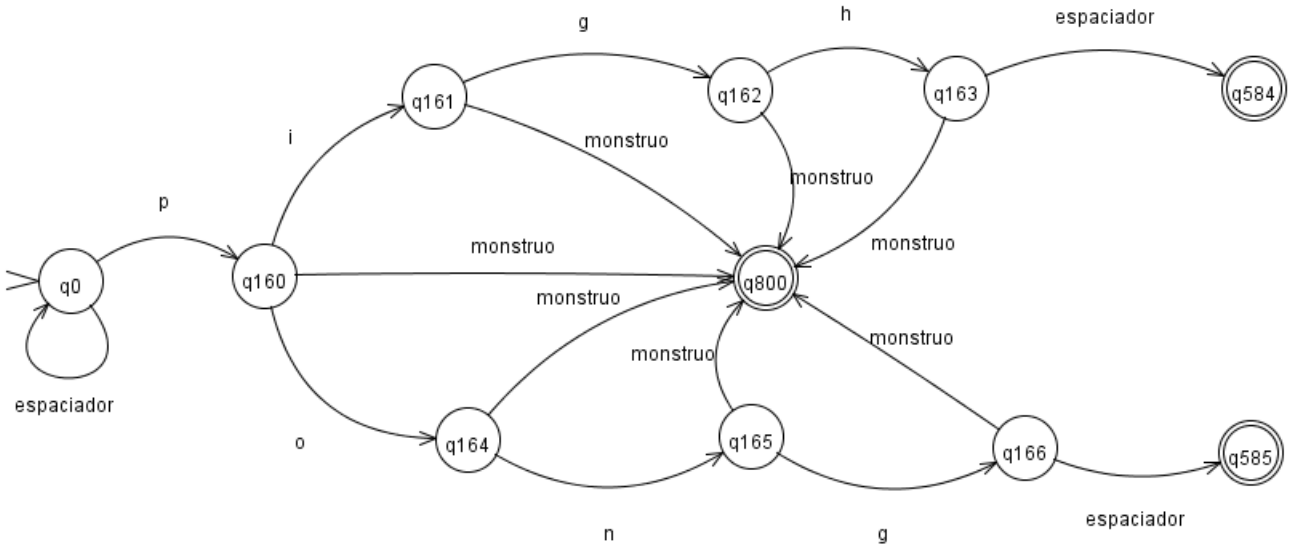
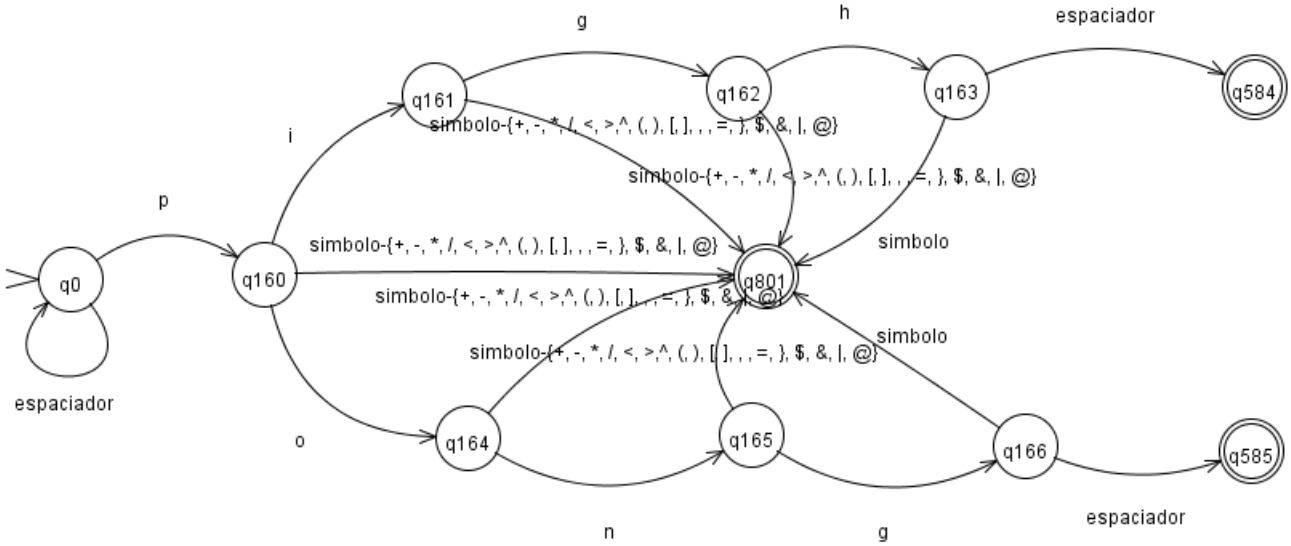


Figure 42: Error 801



5.3.38 Reconocedor de q, estado q167-q176

Figure 43: Estado 1

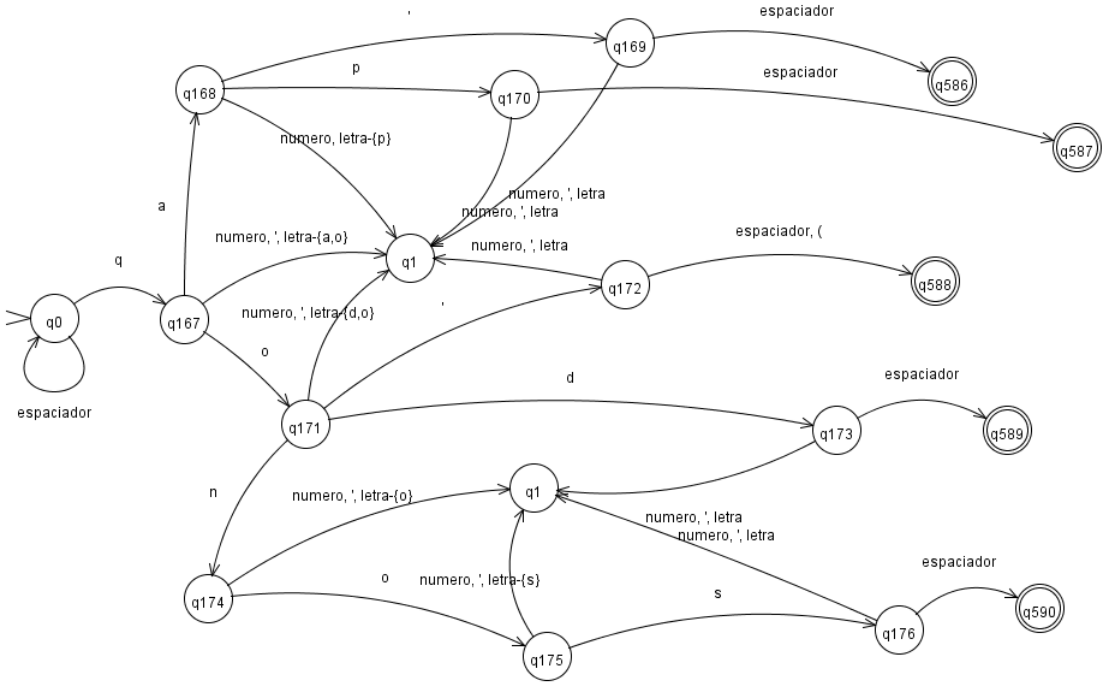


Figure 44: Estado 500

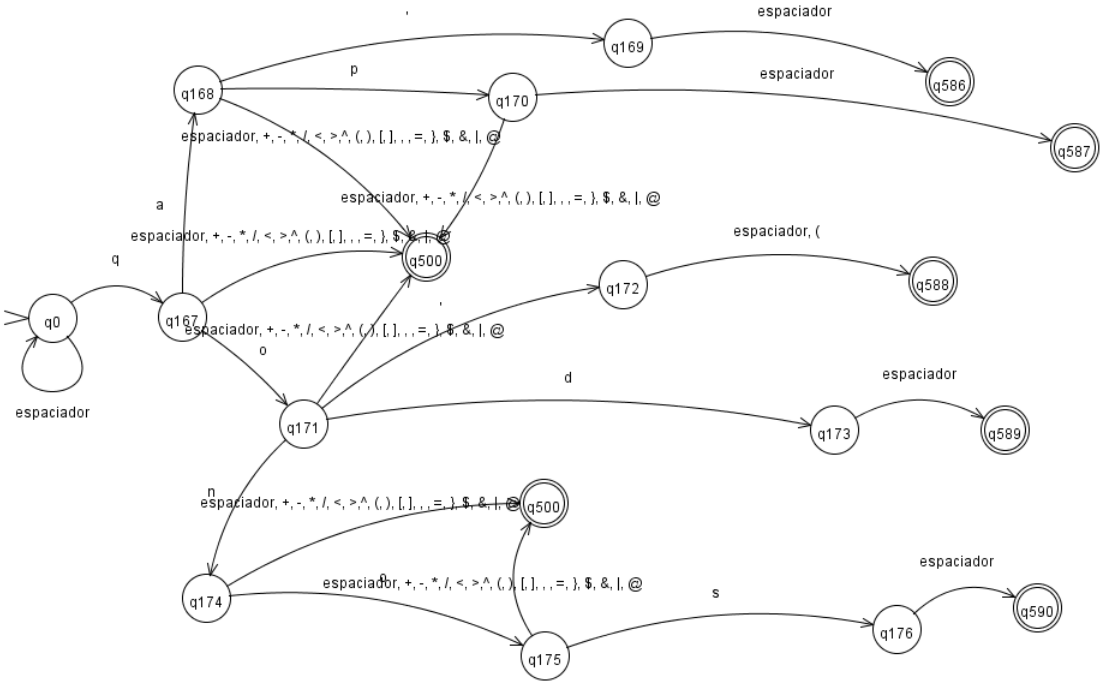


Figure 45: Error 800

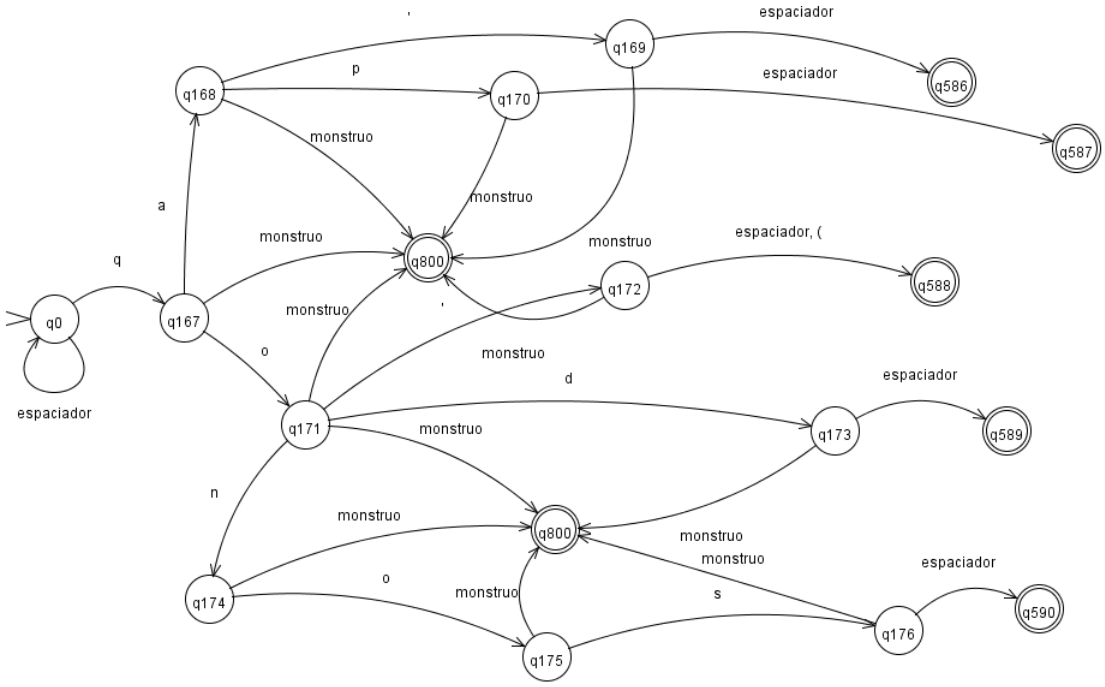
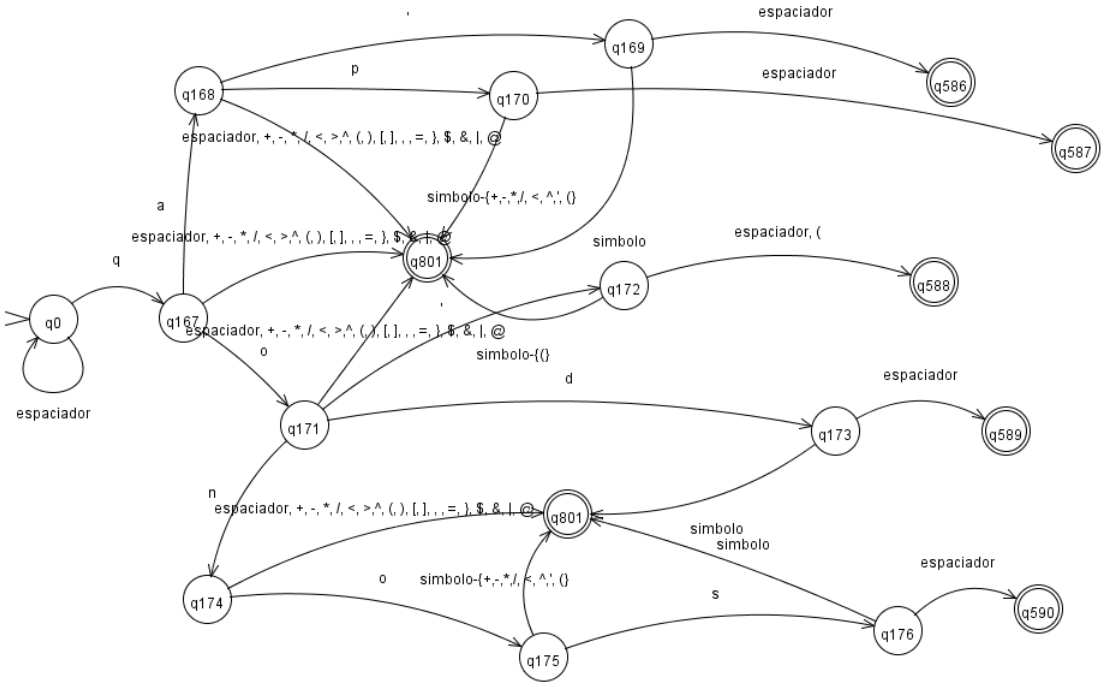


Figure 46: Error 801



5.3.39 Reconocedor de s, estado q177-q184

Figure 47: Estado 1

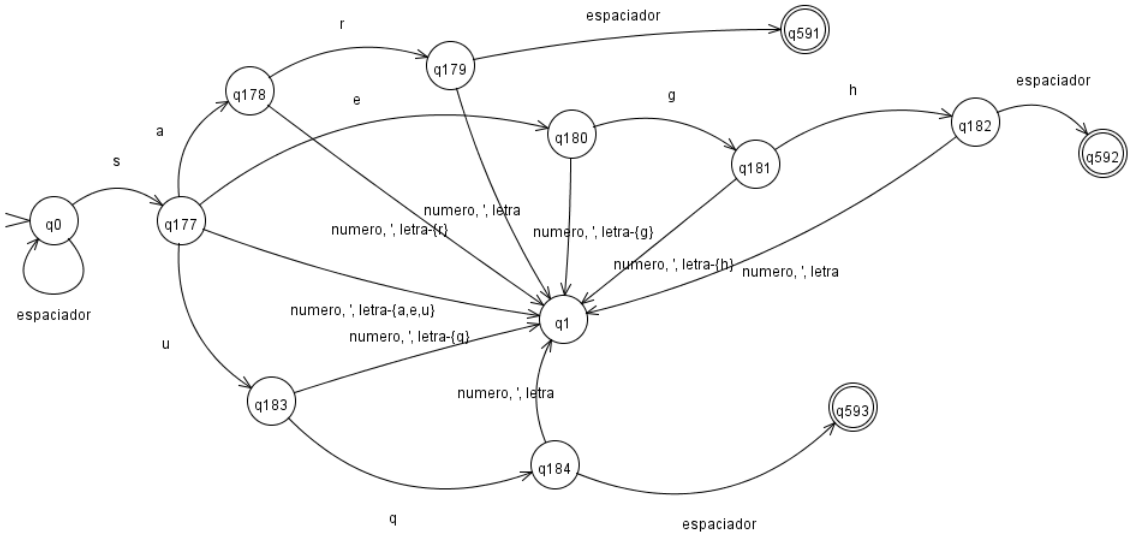


Figure 48: Estado 500

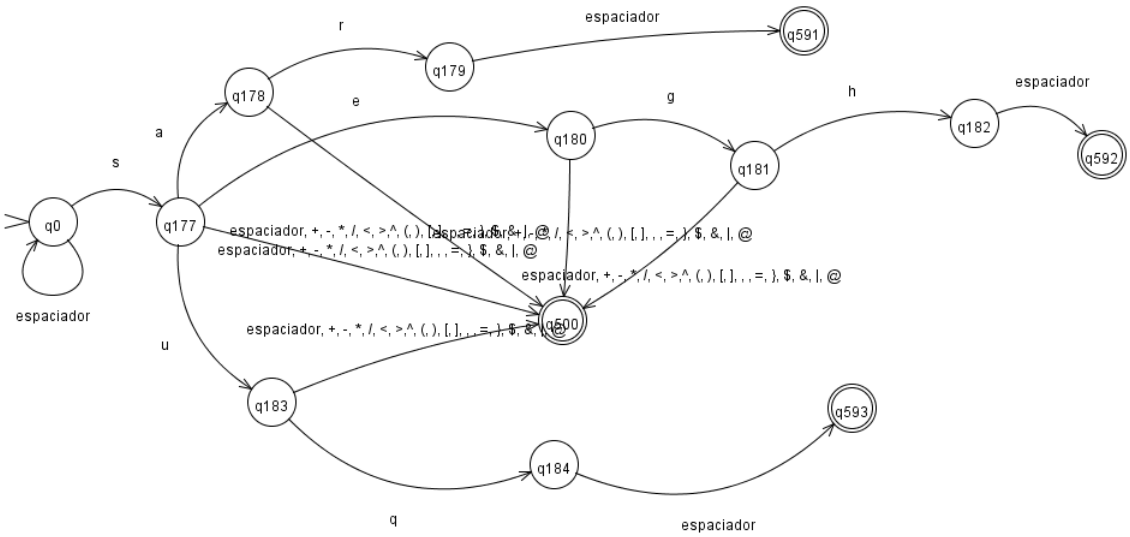


Figure 49: Error 800

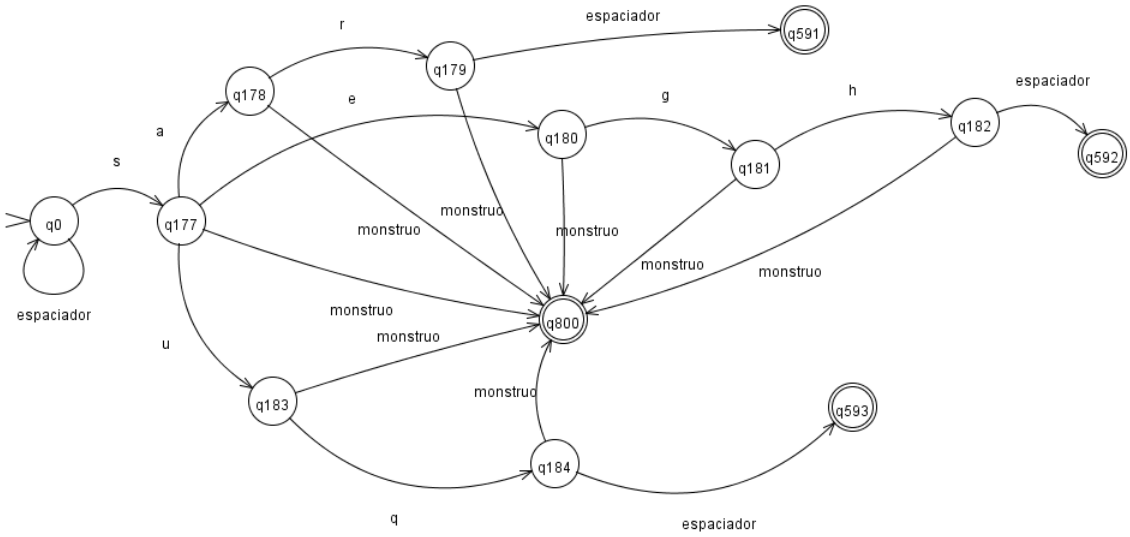
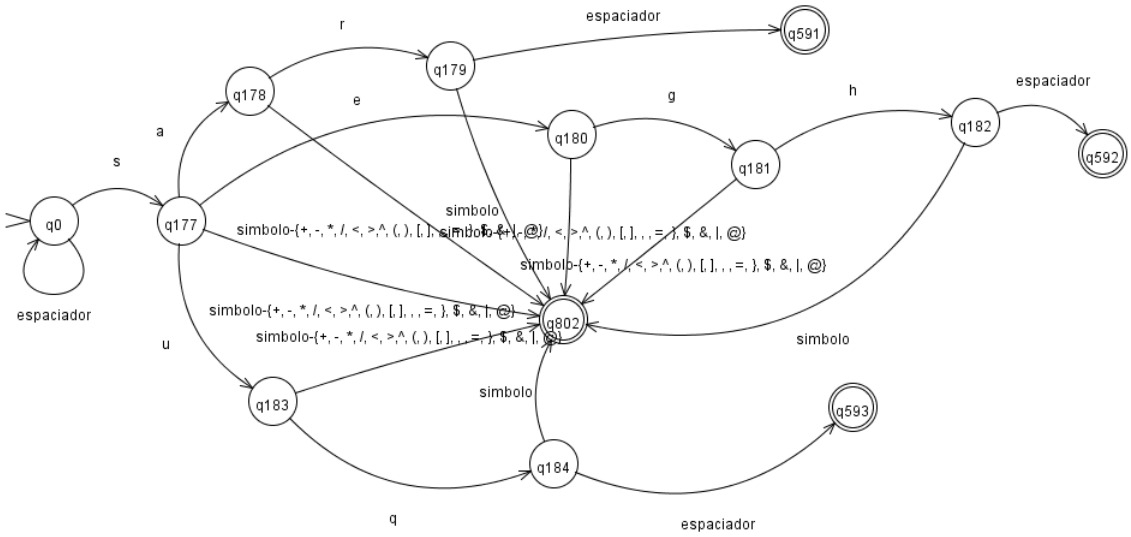


Figure 50: Error 801



5.3.40 Reconocedor de t, estado q185-q195

Figure 51: Estado 1

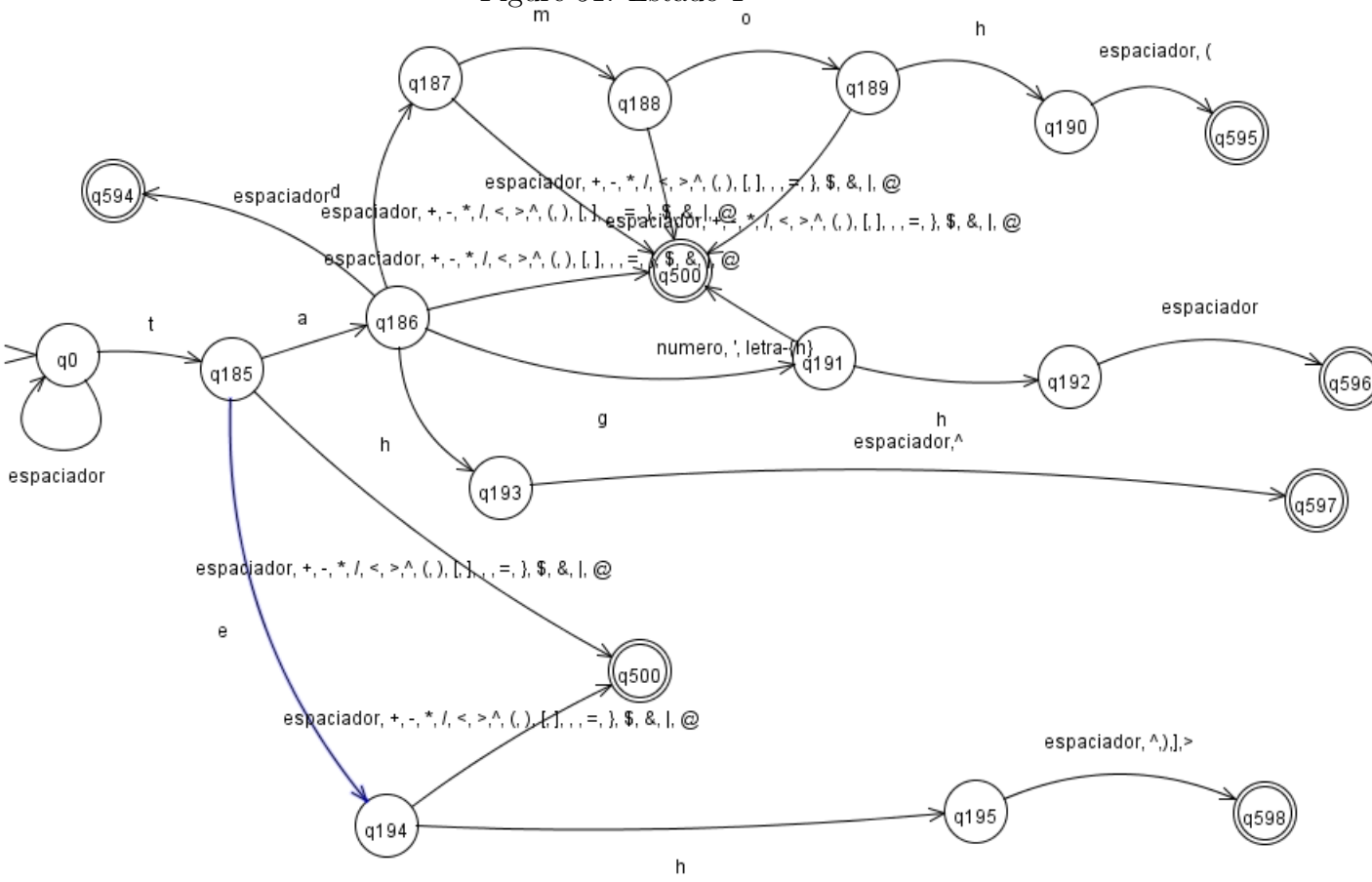


Figure 52: Estado 500

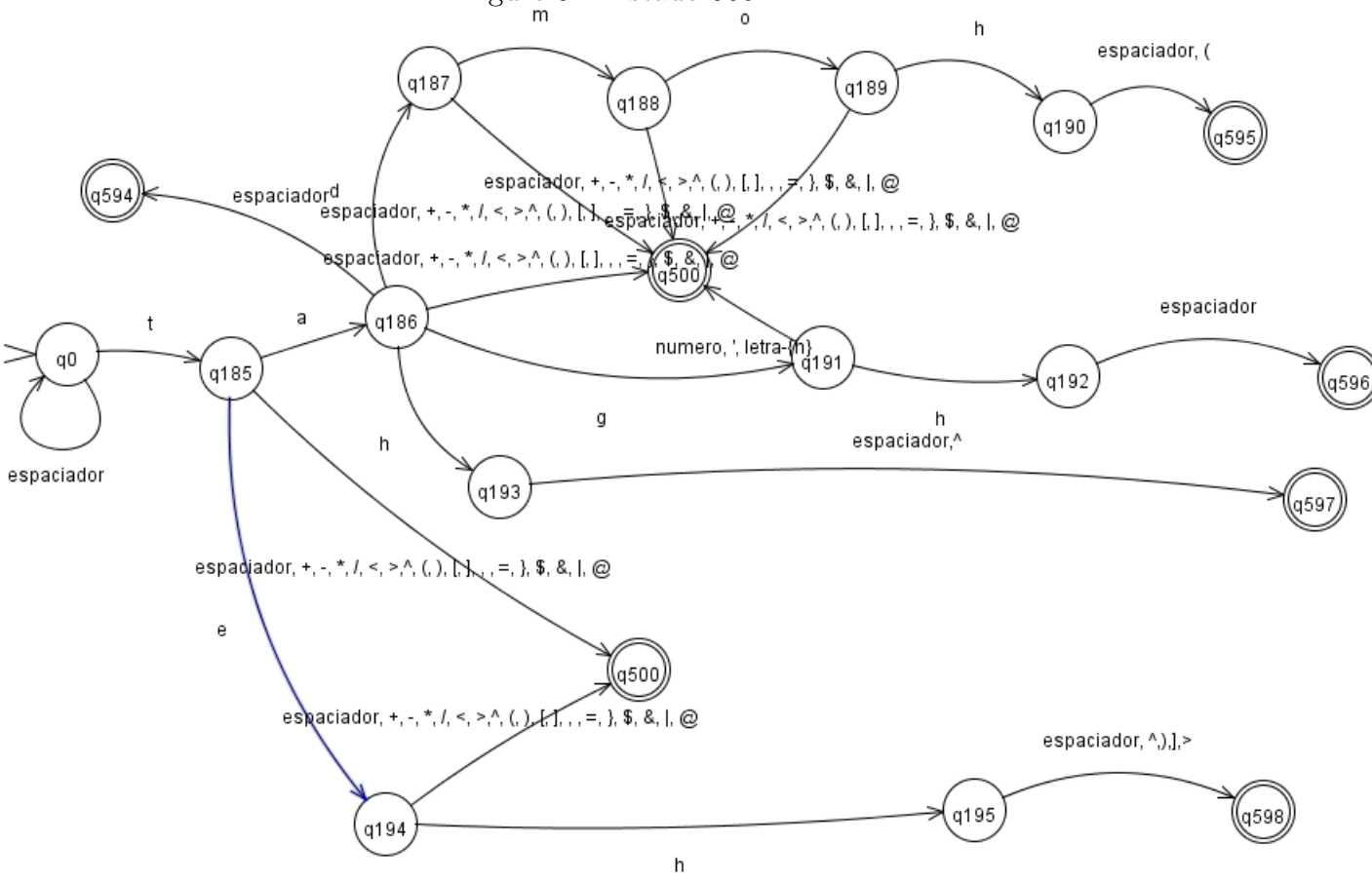


Figure 53: Error 800

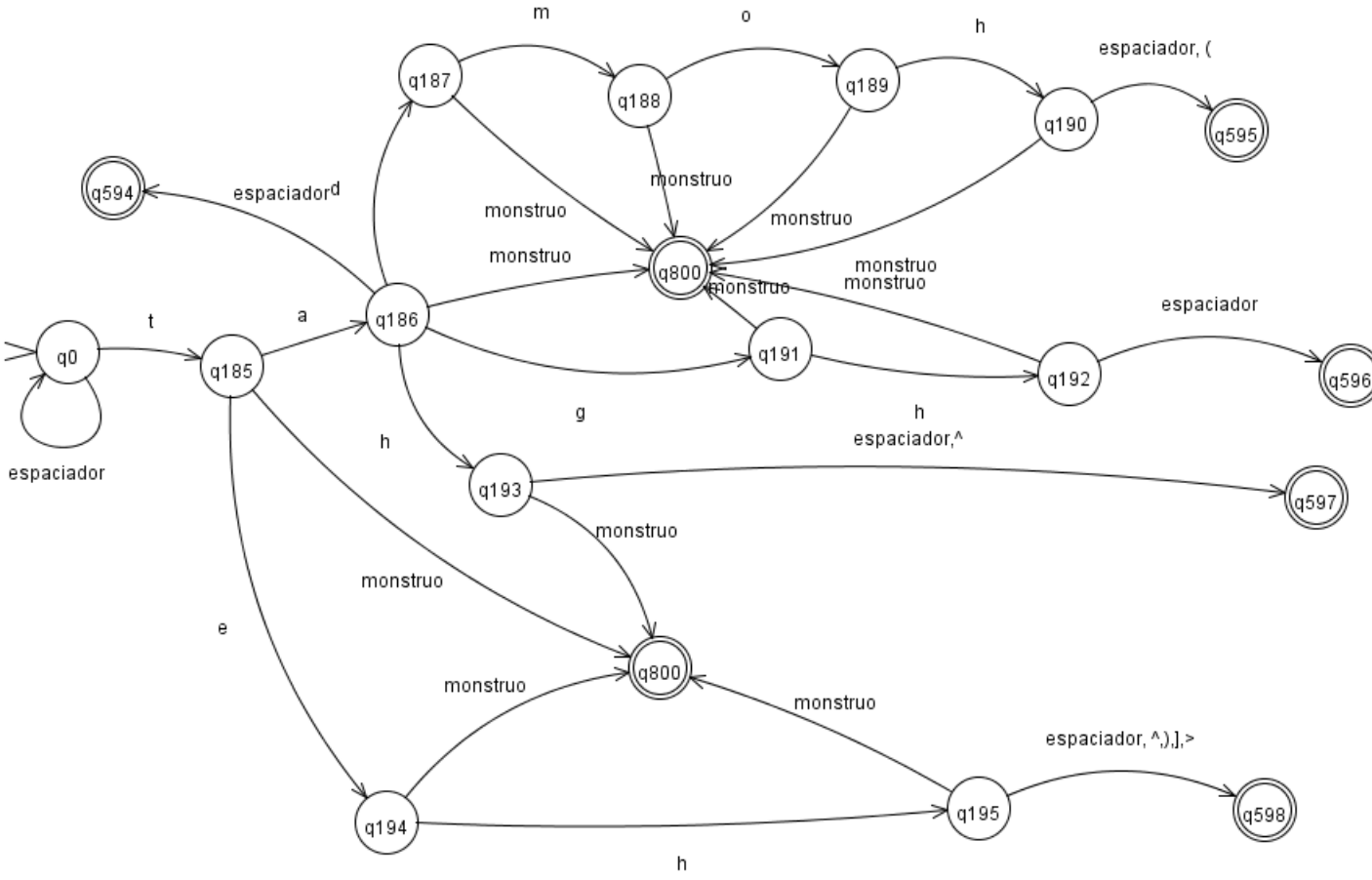
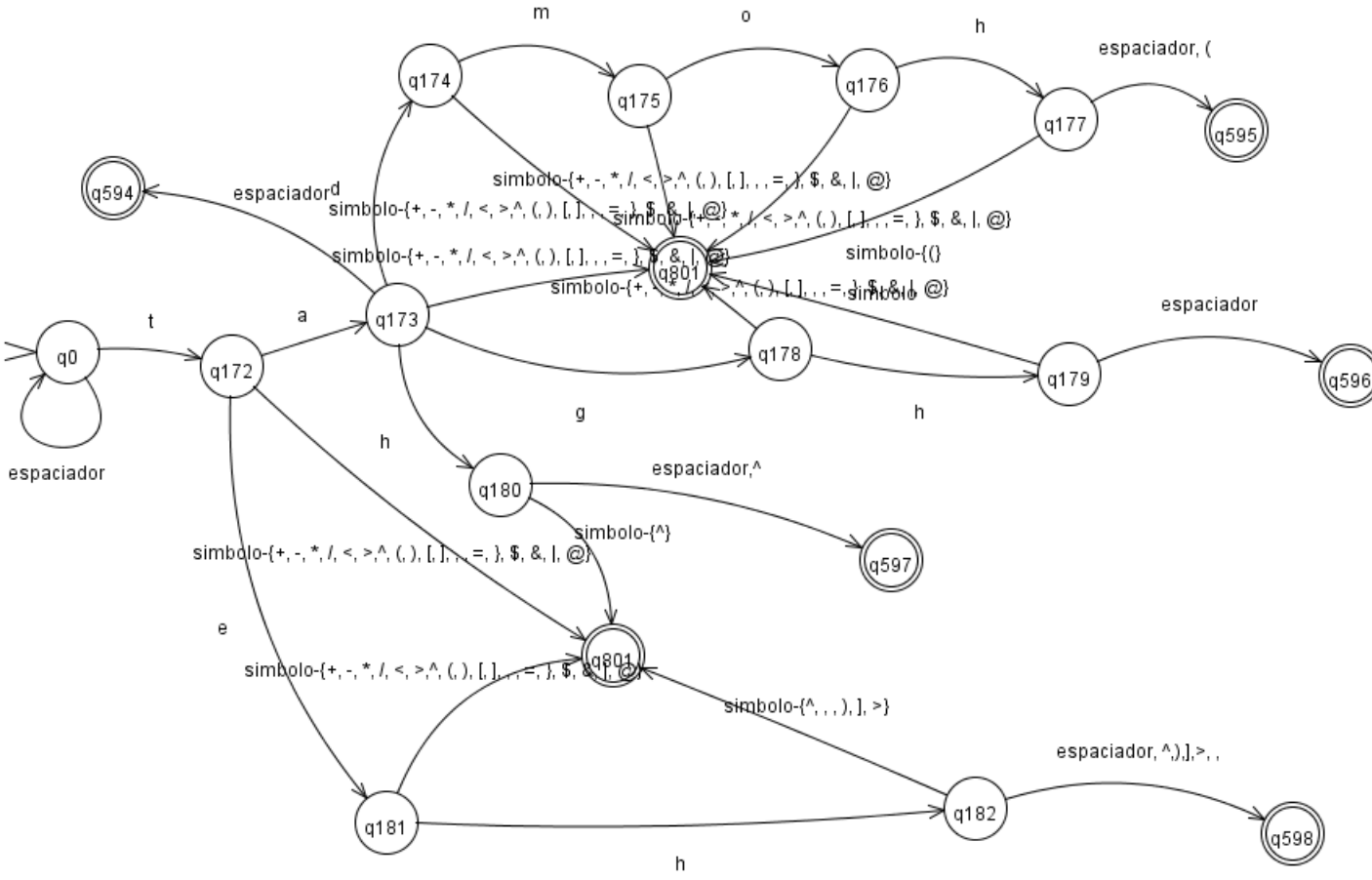


Figure 54: Error 801



5.3.41 Reconocedor de v, estado q196-q203

Figure 55: Estado 1

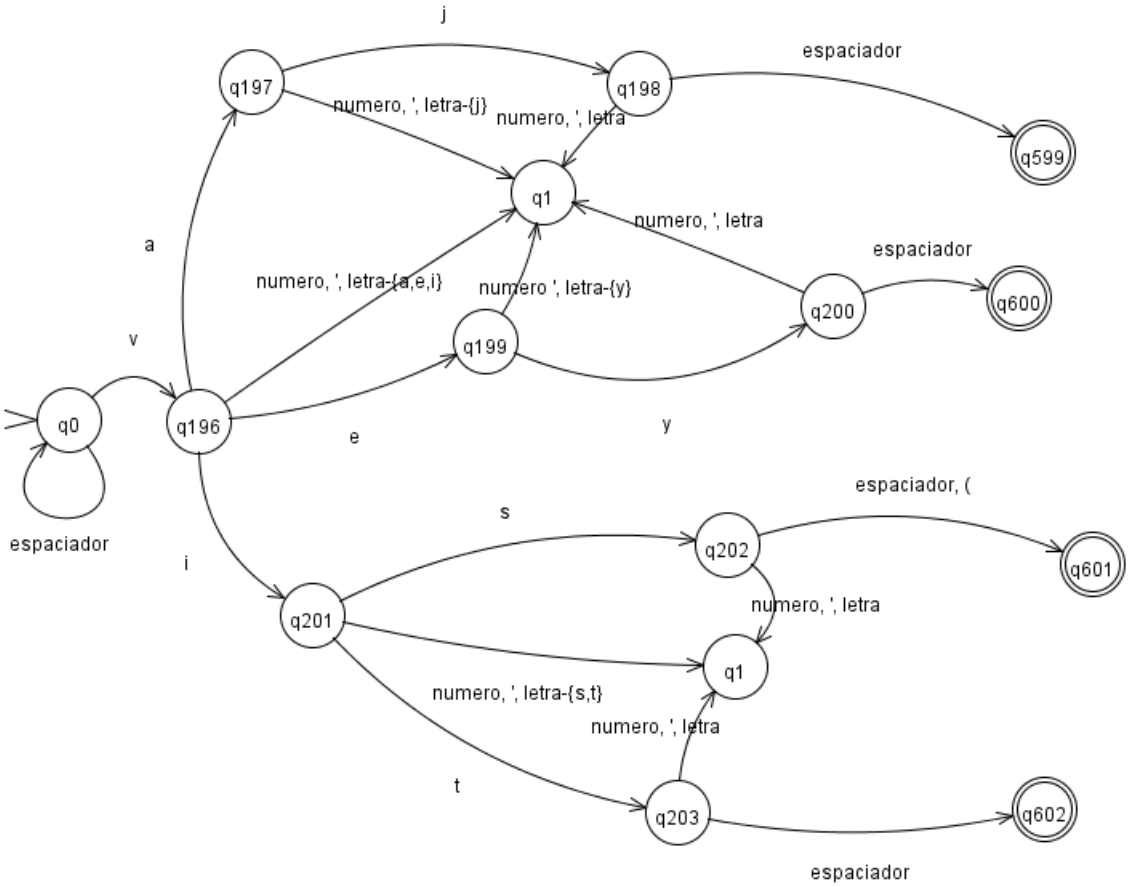


Figure 56: Estado 500

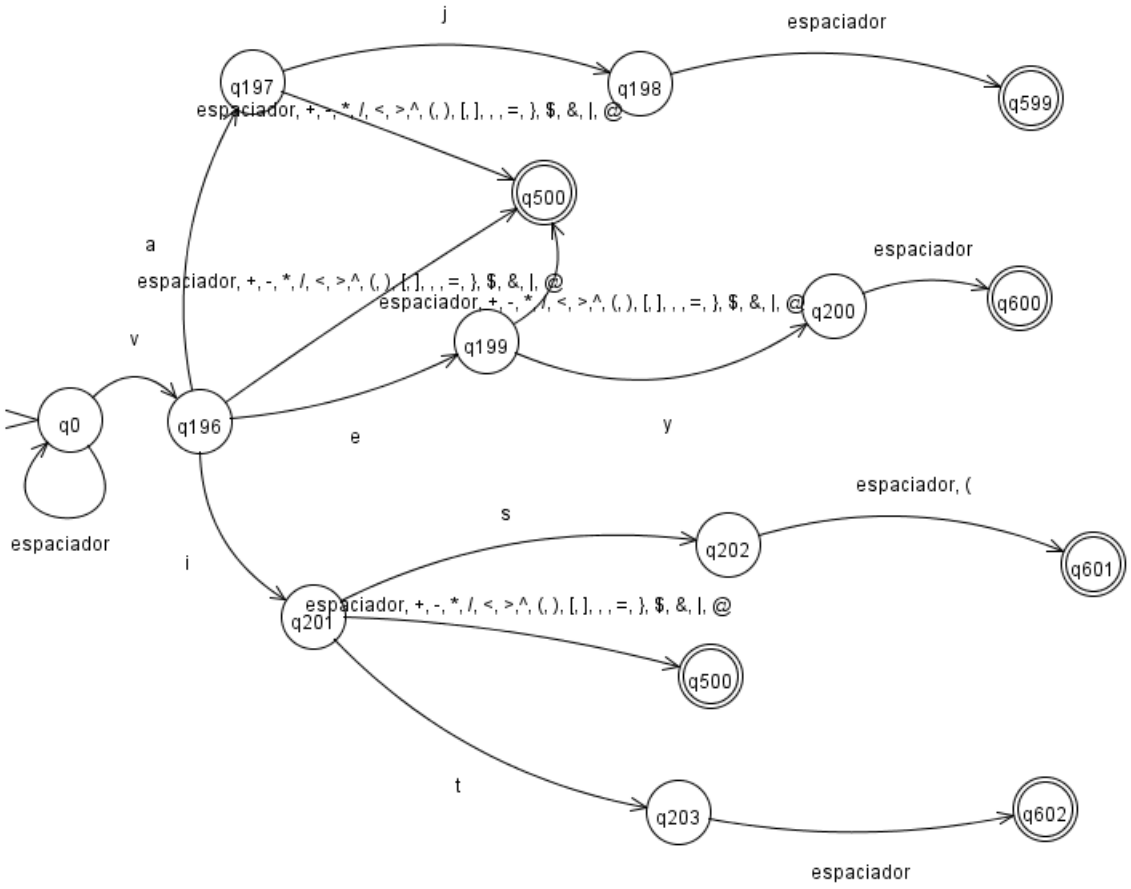


Figure 57: Error 800

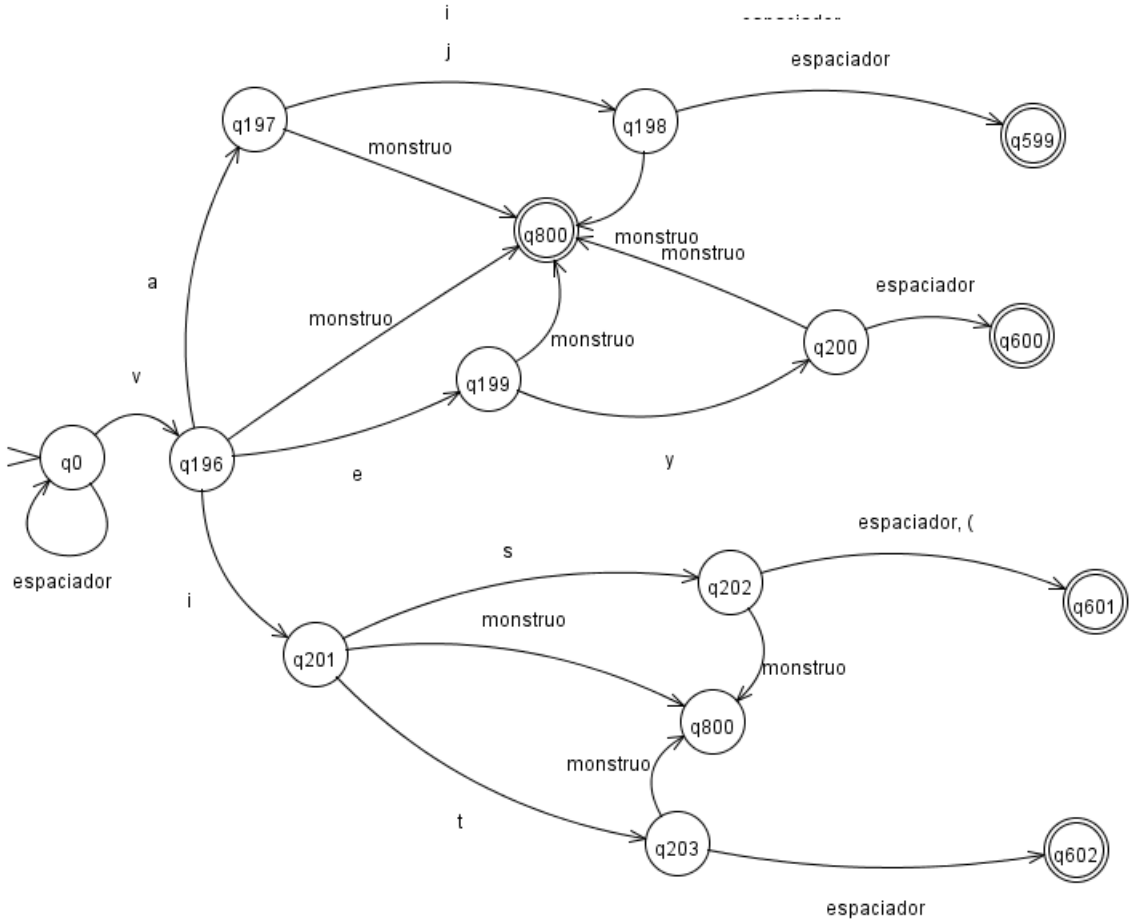
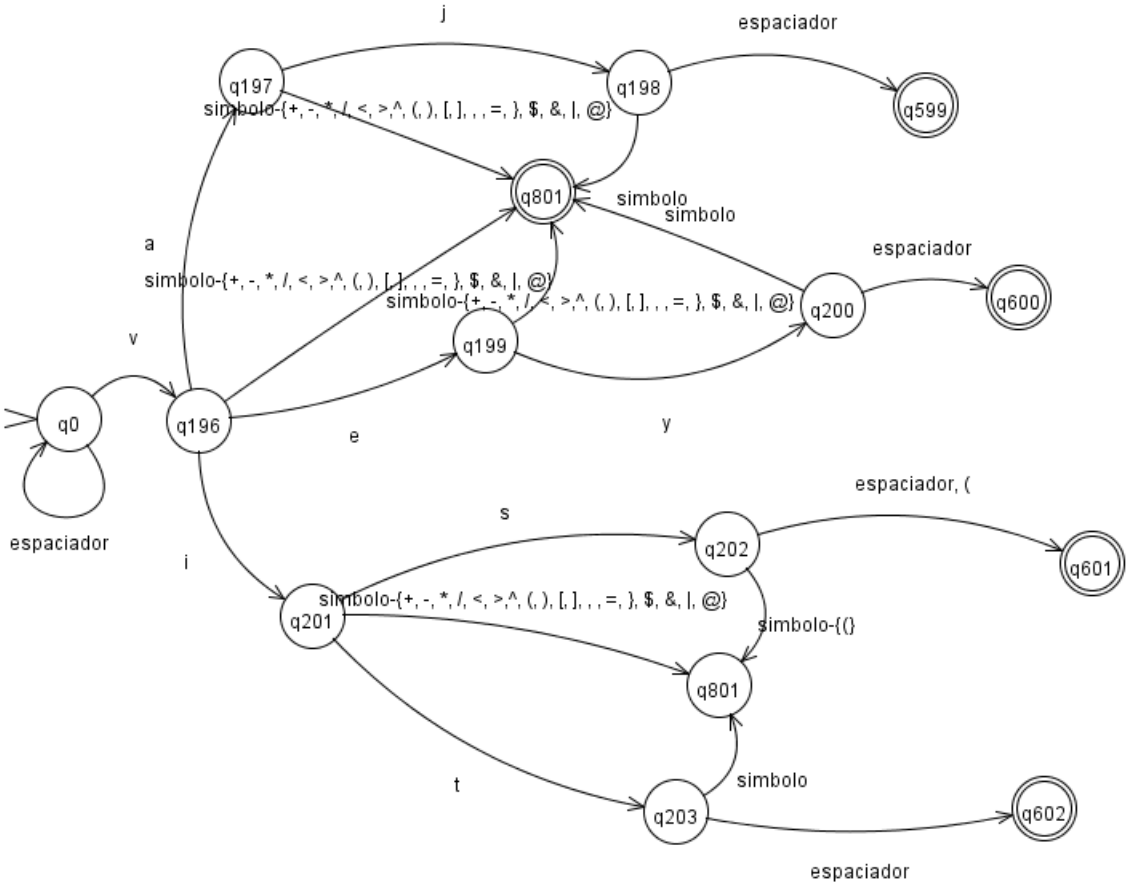
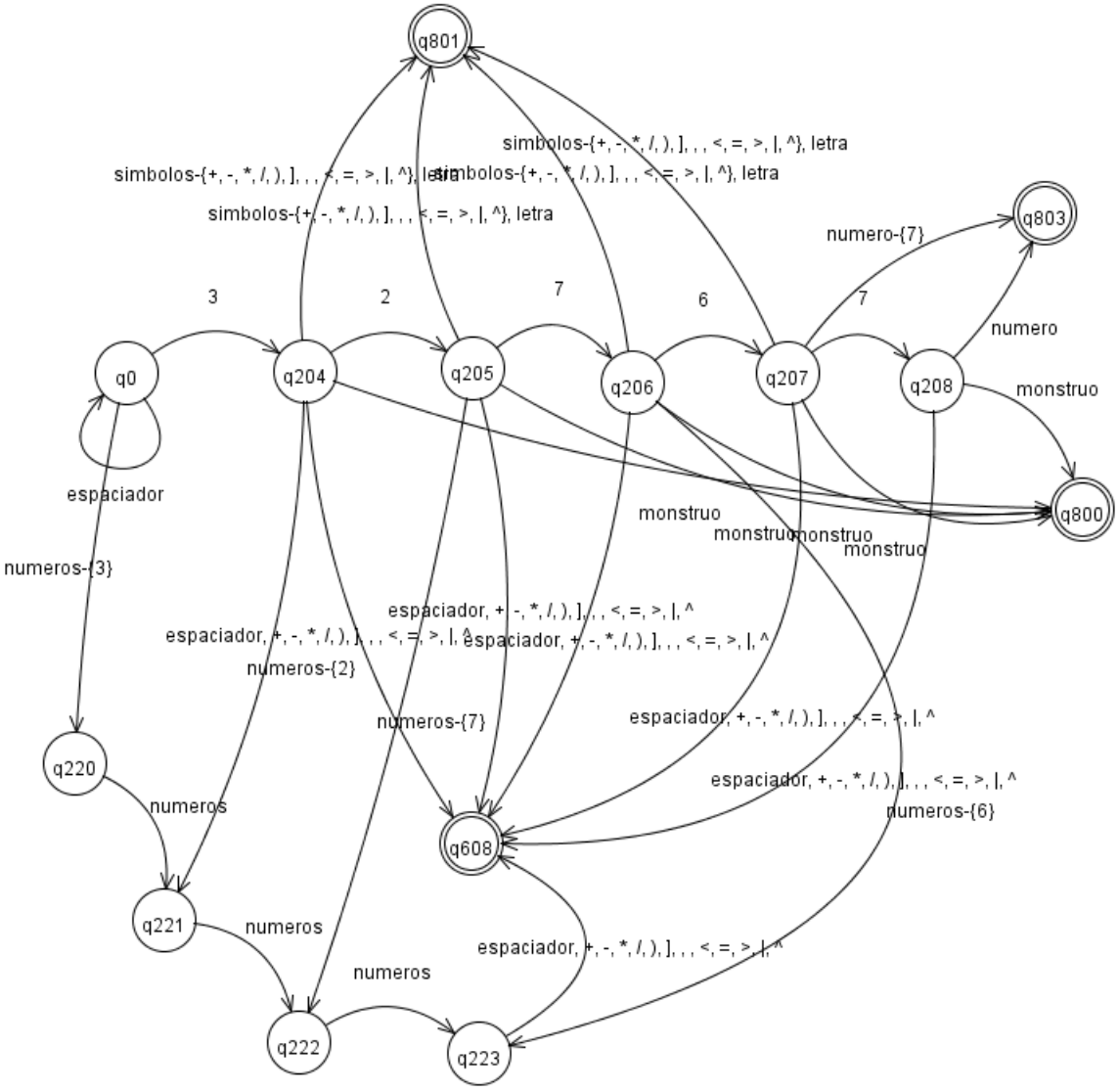


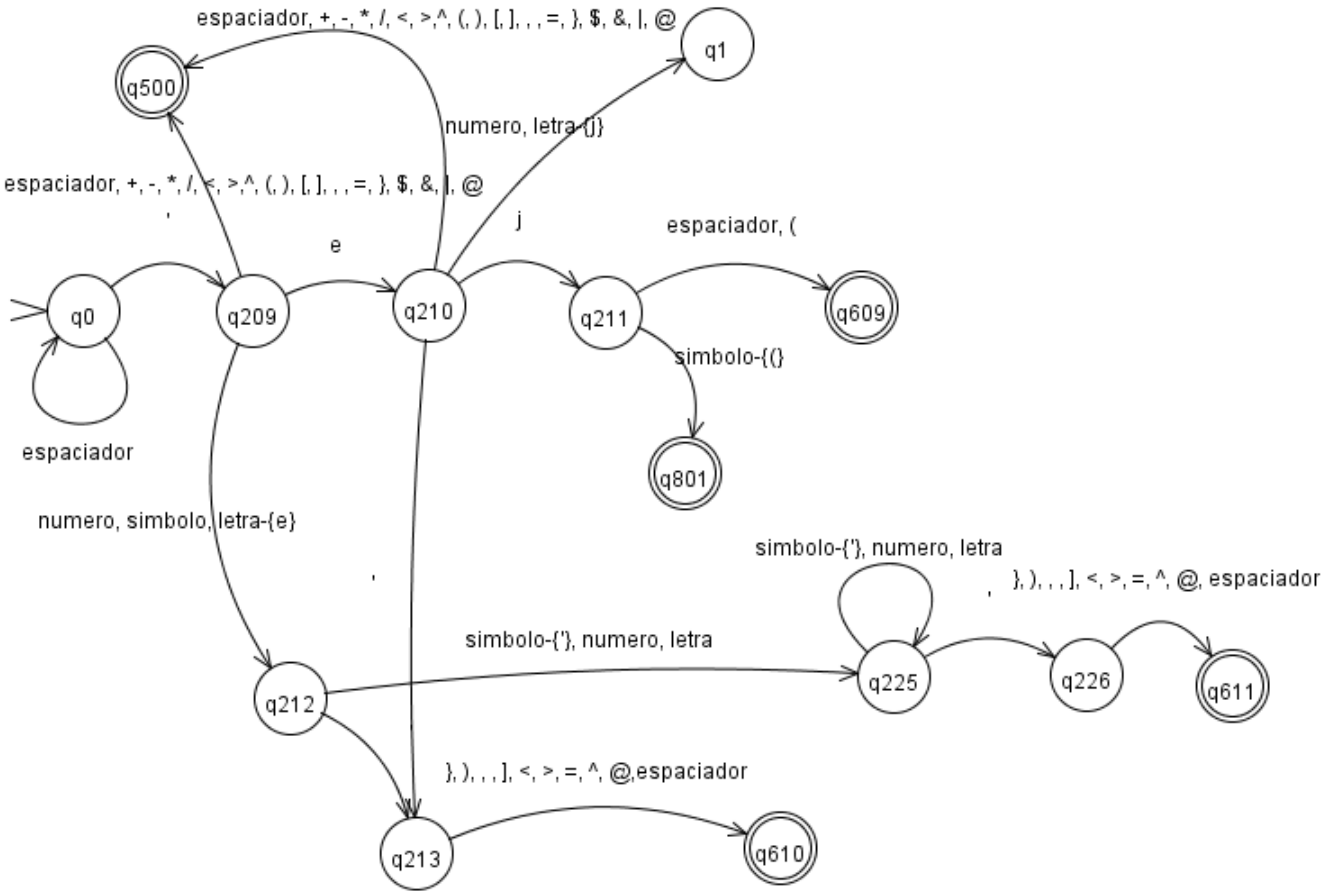
Figure 58: Error 801



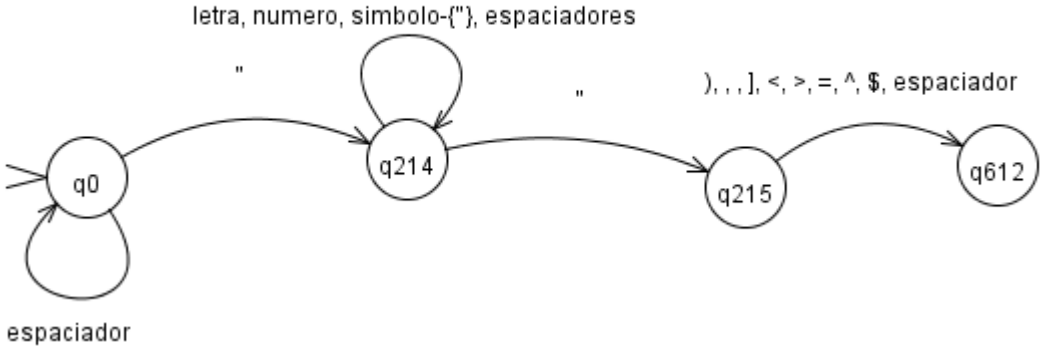
5.3.42 Reconocedor de numero, estado q204-q208



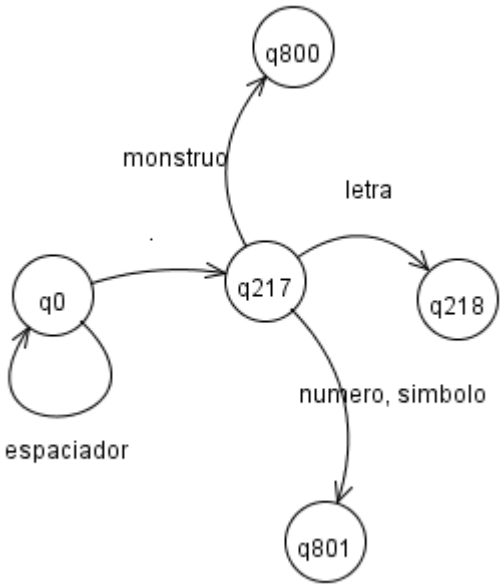
5.3.43 Reconocedor de comilla('), estado q209-q213



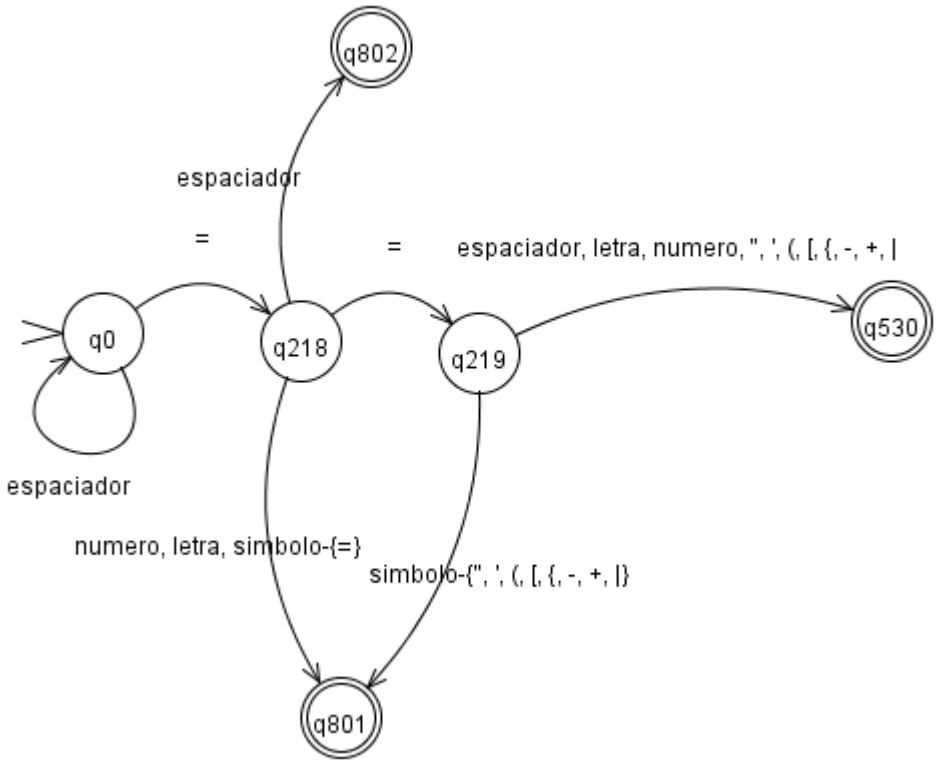
5.3.44 Reconocedor de comillas("), estado q214-q215



5.3.45 Reconocedor de coma(,), estado q217



5.3.46 Reconocedor de coma(,), estado q218-q219



6 Recuperación de estados

Por falta de tiempo y otros factores, el compilador funciona por pánico.