GLL – a highly ambiguous grammar

A brief overview of the GLL approach and the functions used in the following example is given in GLL Algorithm Sketch and Terminology on our GLL parsing webpage `www.rhul.ac.uk/computerscience/research/csle/researchareas/gllparsers.aspx` . Below is a GLL parser for the grammar

$$S \quad ::= \quad b \mid S\ S \mid S\ S\ S$$

---

read the input into I and set $I[m] := \$$
create GSS node $u_o := (L_0, 0)$
set $c_I := 0$; $c_U := u_0$; $c_N := \$$
set $\mathcal{R} := \emptyset$; $\mathcal{P} := \emptyset$;
for $0 \leq j \leq m$ { set $\mathcal{U}_j := \emptyset$ }
goto L_S

L_0: **if** $(\mathcal{R} \neq \emptyset)$ { remove (L, u, i, w) from $\mathcal{R}$
$c_U := u$; $c_I := i$; $c_N := w$; **goto** L }
else if (there is an SPPF node $(S, 0, m)$) report success
else report failure

L_S: **if** $(test(I[c_I], S, b))$ $add(L_{S_1}, c_U, c_I, \$)$
if $(test(I[c_I], S, SS))$ $add(L_{S_2}, c_U, c_I, \$)$
if $(test(I[c_I], S, SSS))$ $add(L_{S_3}, c_U, c_I, \$)$
goto L_0

L_{S_1}: $c_N := getNodeT(b, c_I)$; $c_I := c_I + 1$
$pop(c_U, c_I, c_N)$; **goto** L_0

L_{S_2}: $c_U := create(R_{S_1}, c_U, c_I, c_N)$; **goto** L_S
R_{S_1}: **if** $(test(I[c_I], S, S))$ { $c_U := create(R_{S_2}, c_U, c_I, c_N)$; **goto** L_S }
else goto L_0
R_{S_2}: $pop(c_U, c_I, c_N)$; **goto** L_0

L_{S_3}: $c_U := create(R_{S_3}, c_U, c_I, c_N)$; **goto** L_S
R_{S_3}: **if** $(test(I[c_I], S, SS))$ { $c_U := create(R_{S_4}, c_U, c_I, c_N)$; **goto** L_S }
else goto L_0
R_{S_4}: **if** $(test(I[c_I], S, S))$ { $c_U := create(R_{S_5}, c_U, c_I, c_N)$; **goto** L_S }
else goto L_0
R_{S_5}: $pop(c_U, c_I, c_N)$; **goto** L_0

Notation usage
m – length of the input string

$ – end-of-string symbol
I – array of length m containing the input string and $
c_I – current input position, an integer between 0 and m
c_U – the current GSS node
c_N – the current SPPF node
c_R – an SPPF node, the right child of the node about to be constructed
$\mathcal{R}$– list of pending descriptors
$\mathcal{U}$ – list of all constructed descriptors
$\mathcal{U}_i$ – all elements $(L.u, w)$ such that $(L, u, i, w) \in \mathcal{U}$
$\mathcal{P}$ – list of GSS node pop records

Centre for Software Language Engineering, Royal Holloway, University of London