New Approaches to Sequential Strong LR(1) Parsing B. Melichar, P. Šaloun

The sequential parsing theory represents the fundament of the Computer Science. Context–free grammars, especially strong LR grammars, seems to be an applicable base for the parallel parsing. For the deterministic parallel parsing of strong LR(1) grammars it is enough to use lookback string of one symbol, see [2]. In [1] is used a table containing for each admissible pair of lookback and lookahead symbol these items: the initial contents of the pushdown store, the final contents of the pushdown store, substring of right parse. Let us to mark the triplet (α, ω, rp). The analysed string is embedded with the starting and closing marker. The table implies that a transformed grammar is used. This is because in many cases a sequence of expansions is done.

The following algorithm of sequential parsing of strong LR(1) grammars is the feedback from the parallel one. The time complexity is $O(n)$ always, where n is the length of string being parsed. One step of the algorithm processes one terminal symbol and produces a substring of the right parse. It is much more in comparison with one step of classical sequential approach. The modified parsing table contains the same information, as the table mentioned above.

Algorithm: Efficient sequential parser for the strong LR(1) grammar.
Input: Modified parsing table $\bar{f}$ for the augmented strong LR(1) grammar $\bar{G} = (N, T, P, \bar{S})$ and the input string $\vdash w \dashv,\ w \in T^*$.
Output: Right parse in case when the string $w \in L(G)$, otherwise the error signalization appears.
Method: Algorithm reads symbols from the input string w, uses the pushdown store and creates the right parse. The initial symbol of the pushdown store is $\vdash$. S is the start symbol of G.
Repeat steps (1) and (2), until the *ACCEPT* or *ERROR* appears. The top symbol of the pushdown store is denoted by X.

1. Establish the lookahead string of the length 1 and designate it as u.
2. (a) If $f(X, u) = (\alpha, \omega, rp)$ read the input symbol u. Pop the string α from the pushdown store. If such a string is not at the top of the pushdown store, the parsing ends with an error signalization. Otherwise, push the string ω to the pushdown store and add the right parse substring rp to the output string.

   (b) If $(X = \dashv) \wedge (u = \varepsilon)$ then *ACCEPT*, parsing ends and the output string is the right parse of the input string only in the case when the string $\vdash S \dashv$ is the sole content of the pushdown store. Otherwise, parsing ends with an error signalization.

   (c) If $f(X, u) = ERROR$, parsing ends with an error signalization.

Example

The augmented strong $LR(1)$ grammar has the form:
$\bar{G} = (\{\bar{S}, E, E', T, T', F\}, \{a, +, *, (,), \vdash, \dashv\}, P, \bar{S})$, the rules P are in Table 1. The efficient sequential strong $LR(1)$ parsing of input string $\vdash a + a * a \dashv$ is given in Table 2.

(0)	$\bar{S} \to \vdash E \dashv$
(1)	$E \to E'T$
(2)	$E' \to E+$
(3)	$E' \to \varepsilon$
(4)	$T \to T'F$
(5)	$T' \to T*$
(6)	$T' \to \varepsilon$
(7)	$F \to (E)$
(8)	$F \to a$

Table 1: Rules of the grammar $\bar{G}$.

$(\vdash$	$, a + a * a \dashv,$	$\varepsilon)$
$\vdash (\vdash E'T'a,$	$+a * a \dashv,$	$36)$
$\vdash (\vdash E+\ ,$	$a * a \dashv,$	$36841)$
$\vdash (\vdash E'T'a,$	$*a \dashv,$	$3684126)$
$\vdash (\vdash E'T'*,$	$a \dashv,$	$368412684)$
$\vdash (\vdash E'T'a,$	$\dashv,$	$3684126845)$
$\vdash (\vdash E \dashv\ ,$	$\varepsilon,$	$3684126845841)$
$\vdash$ *accept*		

Table 2: Parsing of $a + a * a$.

	a	$+$	$*$	$($	$)$	$\dashv$
$\vdash$	$\vdash$ $\vdash E'T'a$ 36			$\vdash$ $\vdash E'T'($ 36		
a		$E'T'a$ $E+$ 841	$T'a$ $T*$ 84		$E'T'a$ $E)$ 841	$E'T'a$ $E \dashv$ 841
$+$	$E+$ $E'T'a$ 26			$E+$ $E'T'($ 26		
$*$	$T*$ $T'a$ 5			$T*$ $E)T'($ 5		
$($	$($ $(E'T'a$ 36			$($ $(E'T'($ 36		
$)$		$E'T'(E)$ $E+$ 741	$T'(E)$ $T*$ 74		$E'T'(E)$ $E)$ 741	$E'T'(E)$ $E \dashv$ 741

Table 3: Parsing table $\bar{F}$ for augmented strong $LR(1)$ grammar $\bar{G}$

References

[1] Melichar, B.: *Parallel Parsing.* The chapter from the textbook, draft. 1996

[2] Šaloun, P.: *Parallel Parsing of Strong LR Grammars.* Internal report, DCS FEE, Czech Technical University, September 1996.