Building Context-Sensitive Parsers from CF Grammars with Regular Control Language

César Bravo* and João José Neto

Laboratório de Linguagens e Tecnologias Adaptativas
Escola Politécnica da Universidade de São Paulo
Av. Prof. Luciano Gualberto, 158, tv. 3 – 05508-900 São Paulo, SP, Brasil
joao.jose@poli.usp.br
lupus@usp.br

Abstract. In this work we propose a method to derive Stack Automata [1] from context-free grammars with regular control languages [3]; by slightly restricting the operation of the machine we obtain a bottom-up parser which operates by reducing instances of the right-hand side of productions in a sentencial form to the corresponding left-hand side; since context-free grammars with regular control languages are Turing powerful and Stack Automata accept, at least, context-sensitive languages [2], the resulting devices are indeed context-sensitive parsers.

In this work, we propose a practical way to build parsers for context-sensitive languages following three steps:

1. Design a context-free grammar G with regular control set R for the target language,
2. Build the finite automaton corresponding to the regular expression R,
3. Obtain the corresponding one-way stack automaton introduced in this paper.

One-way stack automata, which are able to write anywhere onto the stack but only read once the input, are defined in [1], page 391, but in this paper their operation will be restricted by the following constraints:

1. **Pre-processing:** Push the whole input string onto the stack,
2. **Processing:** Nondeterministically scan the stack from top to bottom in order to determine some transition to be applied; if a match is found, then perform the transition. Afterwards, make the stack pointer indicate the top of the stack,
3. **Halting Condition:** Stop accepting the input just when the stack becomes empty.

Let $G = (N, T, P, S, R, F)$ be a context-free grammar with regular control language [3]. Note that the stack alphabet will be formed by the sentencial forms generated by G.

* Supported by a CAPES PhD grant.

O.H. Ibarra and Z. Dang (Eds.): CIAA 2003, LNCS 2759, pp. 306–308, 2003.

1. Obtain a completely specified deterministic finite automaton corresponding to the regular expression R. Note that the input alphabet for this automaton is $\{lhs/rhs : lhs \rightarrow rhs \in P\}$;
2. Invert the orientation of all its transitions and swap the roles of its initial and final states. Convert the resulting machine into a deterministic finite-state automaton;
3. For each transition $p \in \delta(q, lhs/rhs)$ in the automaton, add a new one $(p, 0, lhs)$ to the set defining $\delta(q, \lambda, (rhs)^R)$ in the stack automaton;
4. For all final states of the deterministic finite automaton (item 2 above), convert them into final states of the stack automaton;
5. Add a new a state to the stack automaton set of states and create a λ transition from this state to the initial state of the deterministic finite automaton.

The resulting machine is a context-sensitive parser. Next we show an illustrative example.

Example. Parser for the Triple-Balanced Language, Parsing the Sentence $a^3b^3c^3$. The parser in this example has been constructed according to the strategy described above, based on the following context-free grammar with regular control set and without appearance checking $G = (\{S, A, B\}, \{a, b, c\}, \{p_1, p_2, p_3, p_4, p_5\}, S, p_1(p_2p_3)^*p_4p_5, \emptyset)$ where: $p_1 = S \rightarrow AB, p_2 = A \rightarrow aAb, p_3 = B \rightarrow cB, p_4 = A \rightarrow ab, p_5 = B \rightarrow c$.

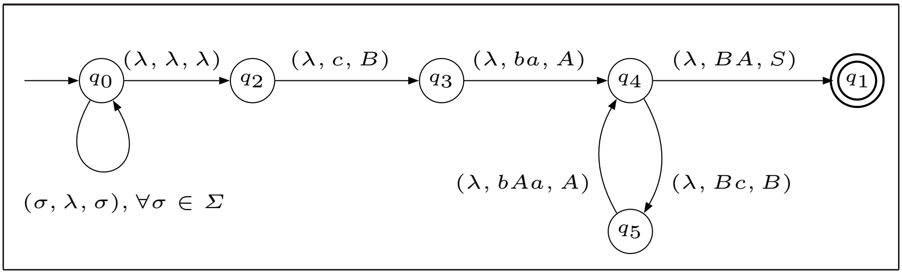

Fig. 1. A Stack Automaton that accepts $a^nb^nc^n$

The table on the right traces the parsing process after pushing the input string onto the stack. The header marks the *parsing time*, counted from the moment the automaton reaches state q_2 on.

time	0	1	2	3	4	5	6	7
	c	B						
	c	c	B					
	c	c	c	B				
	b	b	c	c				
	b	b	b	b	B			
	b	b	b	b	c	B		
	a	a	A	A	b	b		
	a	a	a	a	A	A	B	
	a	a	a	a	a	a	A	S
previous state	q_0	q_2	q_3	q_4	q_5	q_4	q_5	q_4
applied production	$-$	p_5	p_4	p_3	p_2	p_3	p_2	p_1
current state	q_2	q_3	q_4	q_5	q_4	q_5	q_4	q_1

References

[1] S. Ginsburg, S. Greibach and M. A. Harrison. One-Way Stack Automata. Journal of the ACM Vol.14, No. 2, January, p.389-418, 1967. 306

[2] J. E. Hopcroft and J. D. Ullman. Sets Accepted by One-Way Stack Automata are Context Sensitive. Information and Control 13, 2, August. p 113-133, 1968. 306

[3] J. Dassow, G. Păun and A. Salomaa. Grammars with controlled derivations. Handbook of Formal Languages. Vol 2. Srpinger-Verlag. 1997. 306