University of Nevada, Las Vegas Computer Science 456/656 Spring 2021

Answers to Assignment 5: Due Thursday April 1, 2021

1. Consider G, the following context-free gammar with start symbol E. Stack states are indicated.

   1. $E \to E_{1,11} +_2 E_3$
   2. $E \to E_{1,11} -_4 E_5$
   3. $E \to E_{1,3,5,11} *_6 E_7$
   4. $E \to -_8 E_9$
   5. $E \to (_{10} E_{11})_{12}$
   6. $E \to x_{13}$

   What follows is an ACTION table followed by a GOTO table for an LALR parser for G. Which entry guarantees that negation has higher priority than multiplication?

   The entry in column "$*$" of row 9. The precedence is indicated by the fact that we reduce before we shift the multiplication sign.

	x	$+$	$-$	$*$	(	)	\$	E
0	s13		s8		s10			1
1		s2	s4	s6			**halt**	
2	s13		s8		s10			3
3		r1	r1	s6		r1	r1	
4	s13		s8		s10			5
5		r2	r2	s6		r2	r2	
6	s13		s8		s10			7
7		r3	r3	r3		r3	r3	
8	s13		s8		s10			9
9		r4	r4	r4		r4	r4	
10	s13		s8		s10			11
11		s2	s4	s6		s12		
12		r5	r5	r5		r5	r5	
13		r6	r6	r6		r6	r6	

2. State the pumping lemma for context-free languages. For any context-free language L, there is an integer $p > 0$, the *pumping length* of L, such that for any $w \in L$ of length at least p, there exist string u, v, x, y, z such that the following four conditions hold.

   1. $w = uvxyz$.
   2. $|vxy| \leq p$.
   3. v and y are not both empty.
   4. For any integer $i \geq 0$, $uv^ixy^iz \in L$.

3. Use the pumping lemma to prove that $L = \left\{a^j b^k c^\ell \,:\, 0 \le j \le k \le \ell\right\}$ is not context-free.

Proof by contradiction. Suppose L is context-free. Let p be the pumping length of L. Let $w = a^p b^p c^p$ which is in L, and $|w| = 3p > p$. Then, by the pumping lemma, there are strings u, v, x, y, z such that the four conditions hold:

   1. $w = uvxyz$.
   2. $|vxy| \le p$.
   3. v and y are not both empty.
   4. For any integer $i \ge 0$, $uv^i xy^i z \in L$.

We observe that any substring of w that contains both an a and a c must have length at least $p + 2$. Therefore vxy either contains no a or no c.

Case 1. vxy contains no a. Then neither v nor y contains a. By the pumping lemma, $w' = uv^2xy^2z \in L$. Since neither v nor y contains a, there are exactly p a's in w'. Since $w' \in L$, that means that w' must contain p b's and p c's. However, w' is longer than w by $|v| + |y|$, which is greater than zero. it follows that $|w'| > 3p$, contradiction.

Case 2. vxy contains no c. The proof is similar to that of Case 1.

4. Consider the following problem. Given binary numerals $\langle u\rangle$ and $\langle v\rangle$ of length n, decide whether $u < v$. Give an $\mathcal{NC}$ algorithm for solving this problem.

The algorithm uses divide-and-conquer, in a manner reminiscent of mergesort. Let $T(n)$ be the time complexity of the problem, and let $W(n)$ be the *work* complexity, meaning the total number of steps executed for an instance of size n. Without loss of generality, we assume that n is a power of 2. (We can always pad each numeral with leading zeros to achieve this.)

The numeral $\langle u\rangle$ is the concatenation of two numerals of length $n/2$. Let $u_L = \lfloor \frac{u}{n/2} \rfloor$ and $u_R = u \,\mathbf{mod}\,(n/2)$. (Recall that the C++ operator % is an implementation of **mod**.) Then $\langle u_L\rangle$ and $\langle u_R\rangle$ are the left and right halves of the string $\langle u\rangle$. (For example, if $n = 8$ and $\langle u\rangle = 10011110$, then $\langle u_L\rangle = 1001$ and $\langle u_R\rangle = 1110$.)

Our $\mathcal{NC}$ algorithm is as follows:
If $n = 1$, that is u and v each have one digit, we need only $O(1)$ steps.

If $n = 2^k$ for $k > 0$, answer the following two sub-questions in parallel:

   1. Is u_L less than, equal to, or greater than v_L?
   2. Is u_R less than, equal to, or greater than v_R?

If $u_{malyL} < v_L$, then $u < v$. If $u_{malyL} > v_L$, then $u < v$. If $u_{malyL} = v_L$, the answer is given by sub-question 2.

We now do the time and work analysis. Our two recurrences are

$W(n) = 2W(n/2) + O(1)$

$T(n) = T(n/2) + O(1)$ since the two subproblems are done simultaneously.

Solving these recurrences, we have that the time complexity is $O(\log n)$, while the work complexity is $O(n)$. Thus, the algorithm is $\mathcal{NC}$.

5. Prove that a language is enumerable in canonical order by some machine if and only if it is decidable.

   Suppose a language L over an alphabet Σ is accepted by a machine M_1. Let M_2 be a machine which:

   1. Generates the canonical enumeration of Σ^*, $w_1, w_2 \ldots$.
   2. For each w_i, emulates M_1 and decides whether $w_i \in L$.
   3. Writes w_i if and only if $w_i \in L$.

   Conversely, suppose there is a machine M_1 which enumerates L in canonical order. There are two cases.

   If L is finite, then L clearly decidable.

   If L is infinite, M_2 decides L by the following method:

   1. Read a string w.
   2. Emulate M_1, until a string w_i is found where $w_i \geq w$ in the canonical order.
   3. If $w_i = w$, then accept w. Otherwise, reject w.