Comprehensive Examination
Compiler Construction
Spring 1997

This is a closed-notes, closed-book exam administered under the Rice Honor Code. You have two hours to complete the exam.

We have tried to give you all the information required to complete each question. If you reach a point where you are missing a critical piece of information, make a reasonable assumption, record it in your answer book, and continue.

---

1. *General Knowledge:* Each of the following items might be built during compiler construction, during compilation, or during execution. For each item, state (*i*) what it is, (*ii*) when it is built, (*iii*) when it is used, and (*iv*) what role it plays in translation or execution.
   (a) free list
   (b) access link
   (c) ACTION table
   (d) interference graph
   (e) dope vector
2. *Storage allocation:*
   (a) Briefly define "activation record." Be sure to include examples of things that are stored in one.
   (b) Describe the various mechanisms that the compiler can use to allocate and free activation records. What language features would influence your choice of allocation discipline for activation records.
   (c) When, if ever, is it legal to allocate an activation record statically?
3. *LR Parsing:*
   (a) Write down the algorithm for *parsing* a sentence using an LR parse table (the "skeleton parser").
   (b) The simple expression grammar on the left hand side was fed to an LR parser generator that produced the *action* and *goto* tables on the right. Using the tables to drive your parsing algorithm from part (a) above, parse the string "`x * (2 ++ y)`". Show the state of the stack and the lookahead symbol when the syntax error is discovered. What is the last action taken prior to discovering the error?

(1) $E \rightarrow E + T$
(2) $E \rightarrow T$
(3) $T \rightarrow T * F$
(4) $T \rightarrow F$
(5) $F \rightarrow (E)$
(6) $F \rightarrow \texttt{id}$

	action						*goto*		
State	`id`	+	*	(	)	$	*E*	*T*	*F*
0	s5			s4			1	2	3
1		s6				acc			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

4. *Data-flow analysis:* In solving global data-flow analysis problems, we are often interested in determining the "meet-over-all-paths" (mop) solution to a problem. However, the mop solution is sometimes expensive or impossible to calculate, so we settle for an estimate of the solution that is a "conservative" solution. When estimating each of the following sets, tell whether too-large or too-small estimates are conservative.

   (a) available expressions

   (b) live variables

   (c) interprocedural changed variables

   (d) reaching definitions

   Explain your answers in terms of the intended use of the information.