Wright State University
Department of Computer Science and Engineering

Finals (35 pts)

1 Cool Project (10 pts)

Explain clearly **all** the changes to be made to the lexer (that is, `cool.flex` file) and the parser (that is, `cool.y` file) you wrote, to incorporate the following expression into the COOL-language.

expr	::=	...
	|	repeat *expr* until *expr* taeper
	|	...

Assume that the semantics of this construct resembles the Pascal language `repeat-until` statement. That is, the `repeat`-body between the keywords `repeat` and `until` is executed as long as the expression between the keywords `until` and `taeper` remains *false*. Note also that the `repeat`-body is executed at least once.

You are not permitted to make any changes to the "backend", including the definitions for the abstract syntax tree. Indirectly, you are simulating the new construct in terms of the existing ones.

2 Grammar and Parsing (15 pts)

Consider the following context-free grammar G.

S	::=	P$
P	::=	(P)
	|	) P (
	|	ϵ

Is it an SLR(1) grammar? An LR(1) grammar? An LL(1) grammar? Justify your answer in each case, such as by constructing a parse table.

3 Language and Parsing (10 pts)

Design an LL(1) grammar for the language $\mathcal{L}$ over $\{0, 1\}$ that contains strings in which each 0 is sandwitched between 1s. That is, $10 \notin \mathcal{L}$, $10101 \in \mathcal{L}$, $\lambda \in \mathcal{L}$, $111110111101 \in \mathcal{L}$, $010 \notin \mathcal{L}$, $1111 \in \mathcal{L}$, etc.