PRIFYSGOL CYMRU; UNIVERSITY OF WALES

DEGREE EXAMINATIONS MAY/JUNE 2002

SWANSEA

Computer Science

CS 218 Compilers

Attempt 2 questions out of 3

Time allowed: 2 hours

Students are permitted to use the dictionaries provided by the University

Students are NOT permitted to use calculators

CS_218 COMPILERS

(Attempt 2 questions out of 3)

Question 1.

(a) Draw and label a diagram showing the structure of the *analysis-synthesis* model of a compiler.

[3 marks]

Describe the important aspects of the following phases of compilation :

- *lexical analysis*
- *semantic analysis*
- *symbol table manager;* and
- *code optimization*

[12 marks]

(b) Consider the following LL(1) grammar for simple arithmetic expressions :

$E \rightarrow TG$
$G \rightarrow +TG \mid \varepsilon$
$T \rightarrow FR$
$R \rightarrow *FR \mid \varepsilon$
$F \rightarrow (E) \mid id$

and its LL(1) parsing table :

	id	$+$	$*$	$($	$)$	$\$$
E	$E \rightarrow TG$			$E \rightarrow TG$		
G		$G \rightarrow +TG$			$G \rightarrow \varepsilon$	$G \rightarrow \varepsilon$
T	$T \rightarrow FR$			$T \rightarrow FR$		
R		$R \rightarrow \varepsilon$	$R \rightarrow *FR$		$R \rightarrow \varepsilon$	$R \rightarrow \varepsilon$
F	$F \rightarrow id$			$F \rightarrow (E)$		

where '$' is used to mean 'end-of-input'. Write down a trace of an LL(1) parse of the string *(id + id) * id* using the parse table.

[6 marks]

(c) Remove the ε - *productions* from the grammar :

$S \rightarrow Ba \mid b$
$B \rightarrow c \mid CdE$
$C \rightarrow \varepsilon \mid eSE$
$E \rightarrow \varepsilon \mid f$

[4 marks]

Question 2.

(a) The following two languages consist of strings over the alphabet $\Sigma = \{a, b\}$. Write down a *regular expression* and draw a DFA *transition diagram* for recognising strings in the two languages.

- all strings which include the substring *ab*.
- all strings which do not include the substring *ba*

[6 marks]

(b) Consider a programming language which includes the symbol '−' (minus) and identifiers formed from an alphabetic character followed by a sequence of 0 or more alphanumeric characters. Draw and label a DFA transition diagram which could form a basis for a *lexical analyser* for this language. You may assume the existence of the type definition:

$$token = (ident, minus, error, endOfInput).$$

Label your diagram to show when each token should be returned by the lexical analyser.

[4 marks]

(c) Write a lexical analyser for the language as a **getNextToken** procedure. You may assume the existence of the global variables

currentToken : token

rereadChar : boolean

and you can also assume the existence of the functions

isAlpha(c : char):boolean

isNumber(c : char):boolean

isWhitespace(c : char):boolean.

Your procedure should correctly set the value of **currentToken** and **rereadChar**. You will need to define an enumerated type **state** for representing the states of the DFA, but your program need not interact with a symbol table. You should write your code in Pascal but you will not be penalised for minor errors of syntax.

[9 marks]

(d) Explain, using examples, what is meant by each of the following :

- *left factoring* of grammar rules.
- removal of *immediate left recursion* from a grammar.

[6 marks]

Question 3.

(a) Let $G = (T, N, S, P)$ be a *context-free* grammar. Explain each of the four components of G.

[4 marks]

(b) Consider the grammar

$$S \rightarrow ab$$
$$S \rightarrow aSb$$

Define the set *Follow*(S) of terminal symbols, including, if appropriate, the end of input marker \$.

[2 marks]

(c) Consider the following LR(0) grammar G.

$$S \rightarrow bSe$$
$$S \rightarrow a$$

Construct the LR(0) state set for the augmented grammar G'.

[6 marks]

(d) Using your state set construct the LR(0) parsing table for G' and write down a parse of the string

$$bbbaeee$$

[7 marks]

(e) Explain, using examples, the meaning of the following terms:

- *rightmost derivation*
- *ambiguous grammar*
- *production rule*

[6 marks]