F29CC2

HERIOT-WATT UNIVERSITY

SCHOOL OF MATHEMATICAL
AND COMPUTER SCIENCES

COMPUTER
SCIENCE

Compilers and Concurrency

Friday 15th May 2009
9:30 to 11:30

Answer **THREE** questions

Please use a **SEPARATE** script book for **EACH** answer.

Candidates may only use a University approved Calculator

1. **(a)** State what the start symbol, terminals, and non-terminals are, for the grammar:

$$(\mathbf{G1}) \qquad A ::= a \mid b \mid cA$$

(2)

(b) In English, describe the action on an input stream of the recogniser procedure associated with A in the grammar $(\mathbf{G1})$. Your answer may use the terms 'eats', 'input stream', 'succeed', and 'fail', without explanation.

(3)

(c) Illustrate your answer to part (b) by stating whether a parser based on your procedure succeeds or fails on input streams $a\mathrm{EOF}$, $ab\mathrm{EOF}$, and $ca\mathrm{EOF}$ (here, EOF is the end-of-file character). Your answer must make clear whether the parser succeeds or fails in each case.

(3)

(d) **i)** In English, describe the recogniser procedure associated with A in the grammar:

$$\begin{aligned} A ::= &\ (A) \mid BAB \mid z \\ B ::= &\ Bx \mid y \end{aligned}$$

(3)

ii) Explain in detail why the grammar above is unsuitable for a recursive descent implementation.

(1)

iii) State, with full working, the lead and follow sets of A.

(3)

(e) Left-factor the grammar

$$A ::= ab \mid ac$$

(2)

(f) Convert the following grammar to a non-ambiguous form which gives $*$ precedence over $+$ and $+$ precedence over $-$, such that $*$, $+$, and $-$ associate to the right.

$$A ::= A + A \mid A - A \mid A * A \mid (A) \mid 0$$

(3)

2. **(a)** Explain what a compiler is, and what an interpreter is. Take care to make clear the difference between these two concepts.

(2)

(b) State one example of a compiled language, and one example of an interpreted language. In your answer, make sure to be clear which is which.

(2)

(c) Compilers are usually divided into a 'front end' and a 'back end'. Explain the meanings of these words.

(2)

(d) State two advantages of the 'front end'/'back end' architecture.

(2)

(e) State one realistic situation where you would prefer to use an interpreter, and one realistic situation where you would prefer to use a compiler. For each situation, explain your reasons.

(4)

(f) Explain in detail the role of the lexer, the parser, and the compiler. In doing so, describe by what sequence of steps a stream of input characters is transformed into tokens, parsed code, semantically annotated parsed code, intermediate code, and finally into a compiled program.

(4)

(g) Explain what a symbol table is and give an example of the kind of information it holds.

(2)

(h) Describe what a *collision* is, what a *hash function* is, and indicate how hash functions can be used to minimise collision problems.

(2)

3. **(a)** When does the parallel composition of a process with *n* states and a process with *m* states results in a process with *nm* (the product of *n* and *m*) states?

Give an example of such two process and their composition.
You are asked to indicate the Labelled Transitions Systems and the Finite State Processes.

(10)

(b) A variable stores values in the range $0...N$ and supports the actions *read* and *write*. Model the variable as a process, `VARIABLE`, using Finite State Processes.

For $N = 2$, check that it can perform the actions given by the trace:

`write.2->read.2->read.2->write.1->write.0->read.0`.

(6)

(c) What is the difference between concurrency and parallelism?

(4)

4. **(a)** Give a simple example of how non-deterministic choice can be expressed in Finite State Processes and draw the Labelled Transition System associated to your example. Provide a description of your example and 3 difference traces of it.

(5)

(b) Draw the Labelled Transition System associated to the following Finite State Process:

```
COUNTDOWN (N=3) = (start->COUNTDOWN[N]),
COUNTDOWN[i:0..N] =
    (when(i>0) tick->COUNTDOWN[i-1]
    |when(i==0)beep->STOP
    |stop->STOP
    ).
```

(5)

(c) A roller-coaster control system only permits its car to depart when it is full. Passengers arriving at the departure platform are registered with the roller-coaster controlled by a turnstile. The controller signals the car to depart when there are enough passengers on the platform to fill the car to its maximum capacity of M passengers. The car goes around the roller-coaster track and then waits for another M passengers. A maximum of M passengers may occupy the platform. Ignore the synchronisation detail of passengers embarking from the platform and car departure. The roller-coaster consists of three processes: `TURNSTILE`, `CONTROL` and `CAR`.

`TURNSTILE` and `CONTROL` interact by the shared action passenger indicating an arrival and `CONTROL` and `CAR` interact by the shared action depart signalling car departure. Draw the structure diagram for the system and provide the Finite State Processes descriptions for each process and the overall composition.

(10)

END OF PAPER