CST1
COMPUTER SCIENCE TRIPOS Part IB

Monday 6 June 2022 11:00 to 14:00 BST

COMPUTER SCIENCE Paper 4

*Answer **five** questions: **up to four** questions from Section A, and **at least one** question from Section B.*

*Submit each question answer in a **separate** PDF. As the file name, use your candidate number, paper and question number (e.g., `1234A-p4-q6.pdf`). Also write your candidate number, paper and question number at the start of each PDF.*

You must follow the official form and conduct instructions for this online examination

SECTION A

1 Compiler Construction

This question concerns constructing $LL(1)$ parsers from context-free grammars.

(a) Suppose we have a grammar that contains these productions.

$$\begin{array}{lcl} S & \rightarrow & \text{if } E \text{ then } S \text{ else } S \text{ end} \\ S & \rightarrow & \text{if } E \text{ then } S \text{ end} \end{array}$$

Note that these productions have a shared initial segment. Explain how this prevents us from automatically generating an $LL(1)$ parser directly from this grammar. [2 marks]

(b) Rewrite the productions from Part (a) so that they could be suitable as input to an $LL(1)$ parser generator. [4 marks]

(c) Eliminate the shared initial segments from these grammar productions.

$$\begin{array}{lcl} A & \rightarrow & aAbCeDg \\ A & \rightarrow & aAbCd \\ B & \rightarrow & eDgEf \end{array}$$

[5 marks]

(d) Give a general method for eliminating shared initial segments from a grammar. [5 marks]

(e) Argue carefully that the grammar produced by your method generates the same language as the original grammar. [4 marks]

2 Compiler Construction

This question involves the derivation of "stack machines" using the CPS transformation.

(a) Consider the following OCaml code of type

```
add_right : int list -> int
```

that returns the sum of the integers in its argument list.

```
let rec add_right l =
  match l with
  | [] -> 0
  | h::tl -> h + (add_right tl);;
```

Explain why this code, as presented, is not tail recursive. [2 marks]

(b) Use the CPS transformation to rewrite `add_right` to a function that could be given the type

```
add_right_cps : int list -> (int -> int) -> int
```

[6 marks]

(c) Apply defunctionalisation to your code for `add_right_cps`. That is, define a (non-functional) data type `cnt` and a transformed function `add_right_dfc` of type

```
add_right_dfc : int list -> cnt -> int
```

[6 marks]

(d) The function `add_right` from Part (a) could be generalised to the following function.

```
let rec fold_right f l accu =
  match l with
  | [] -> accu
  | a::l -> f a (fold_right f l accu);;
```

For simplicity, we will treat this code as if it had the type

```
fold_right : (int -> int -> int) -> int list -> int -> int
```

and not worry about polymorphism. Rewrite this program using the CPS transformation. Justify your treatment of the variable `f`. What problems might you encounter in attempting to defunctionalise your CPS version? [6 marks]

3 Further Java

A novice Java developer designs a client-server file download service for the Internet. On start-up the server accepts a directory name on the command line and then operates in a loop, retrieving files from this directory when requested to do so by clients. Clients request the contents of a file by connecting to the server on port 1991 and sending the text string of the filename followed by a newline character. The server responds to a client request for a file by sending the contents of the file as bytes to the client before closing the connection and waiting for the request from the next client. If the client requests a file which does not exist, the server closes the connection.

(a) Sketch Java code for a single-threaded version of the server with blocking I/O as specified above. You do not need to handle failure due to `java.io.Exception`. You may make use of the Java standard library and you might find the following classes helpful: `java.io.BufferedReader`, `java.io.File`, `java.io.FileInputStream` and `java.io.InputStreamReader`. [8 marks]

(b) Describe a significant disadvantage of implementing the server as required for Part (a). How might this be mitigated? [3 marks]

(c) Another developer suggests the server is re-written to send or receive serialized Java objects in all communications between the client and the server. Briefly outline how this approach would work and provide suitable class definitions for any required serialised objects. Describe one advantage and one disadvantage of this approach compared to the method used in Part (a). [4 marks]

(d) Outline in words how you might improve this service so that clients can securely download, upload, modify and delete files stored by the server. Consideration should also be given to communicating server errors to the client. [5 marks]

4 Programming in C and C++

(*a*) BCPL was a precursor to C. It stored character strings using a byte initialised to l, the length of the string, with the characters of the string held in the following bytes. What differences arise from the C method for storing character strings? [3 marks]

(*b*) A dictionary data structure in C stores key/value pairs where both elements are character strings. Entries are created or updated using

```
void insert(const char *key, const char *value)
```

Discuss whether the caller or callee should be responsible for allocating store for the strings. What consequences could arise if a caller mutates its local copy of a key? [5 marks]

(*c*) Students have been told that, in many situations, C array declarations are interchangeable with C pointer declarations. Rather than using a shared header file, a student writes '`extern char *mydata;`' at top-level directly in a `.c` file. Their program does not work. A friend changes it to '`extern char mydata[];`'. Might this change make any difference to compiler errors/warnings or run-time operation? Would a shared header file ('`.h`') help? [6 marks]

(*d*) Stating any assumptions made, and considering endianness, describe what happens inside the compiler and at run time for each of the following type casts.

```
extern double f1;
int i1 = (int)f1;
char *p2 = (char *)(&i1);
int i3 = (int)(*p2);
```

[6 marks]

5 Programming in C and C++

(a) A FIFO is implemented in C using a singly-linked list that maintains global head and tail pointers. These are initialised to represent the empty FIFO as follows:

```
struct fifo_entry *head_ptr = 0;
struct fifo_entry **tail_ptr = &head_ptr;
```

The entries in the FIFO use a union to store either an integer or a double-precision floating point number. A further field records which is stored. Give syntactically-accurate C code that defines `fifo_entry` and either a function `enqueue_int` or a function `enqueue_double` enqueuing a new value into the FIFO. [5 marks]

(b) To avoid repetitive reallocation of memory, suppose now that FIFO entries that are no longer in use are to be saved in an auxiliary linked list. Give syntactically-accurate code that implements this approach and then discuss two other approaches for store management. [5 marks]

(c) The C code in part (a) suffers from a lack of encapsulation – variables like `head_ptr` are visible to the the rest of the program. Write C++ defining a `class FIFO` which maintains a single FIFO implemented using elements `fifo_entry` as defined in your answer above, but which only exports member functions `enqueue_int`, `enqueue_double`, `isempty` and

```
void dequeue(void do_I(int), void do_D(double));
```

It should not be possible to create an instance of `class FIFO` and storage allocation/deallocation should use C++ mechanisms rather than those of C. Your C++ is not required to be syntactically accurate, but should capture the main concepts. It is not necessary to give full code for the above four member functions – focusing on allocation and deallocation of `fifo_entry` elements suffices. [*Note:* `do_I` and `do_D` are user-provided processing functions to be applied to the dequeued value.] [6 marks]

(d) The following lines approximate (e.g. omitting access qualifiers) analogous generic/templated class definitions in Java and C++. Explain which types X are valid for use in `Gen<`X`>` and `Tem<`X`>`.

```
[Java]: class       Gen<T>              { T v; Gen() { v = 1; } };
[C++]: template<typename T> class Tem { T v; Tem() { v = 1; } };
```

[4 marks]

6 Security

As an Elite Pentester, you have been engaged to attack a C program that Megacorp runs on an obsolete OS on a 32-bit x86 box, in the mistaken belief that their intrusion detection system offers sufficient protection. The target program accepts textual input from unauthenticated remote users. From prior intelligence you know that the input routine uses a word-aligned 64-byte buffer, that it is vulnerable to buffer overflow, and that modern buffer overflow countermeasures are not enabled. You can replicate the hardware and OS of the target in your lab, but you do not have a copy of the software. So as not to trigger the intrusion detection system, you may only send *one* attack input to the remote server. You have estimated that the buffer will appear at 0xfffe8200 plus or minus 4096 bytes, both endpoints included, and that the return address will be between 64 and 1024 bytes from the start of the buffer, both endpoints included. You wish to craft an attack input (as a sequence of bytes) that will result in the execution of a specific machine code payload (supplied, of length 113 bytes) on the remote machine. Your attack input consists of n_{ret} repetitions of your rewritten return address r, starting at offset $o_{ret} = 0$; followed by n_{nop} repetitions of the nop instruction, starting at offset $o_{nop} = 4 \cdot n_{ret}$; followed by your payload, starting at offset $o_{pl} = 4 \cdot n_{ret} + n_{nop}$; for a total file length $l = 4 \cdot n_{ret} + n_{nop} + 113$ bytes.

[*Note:* Correct numerical answers are important for this question. Please highlight your final numerical answers to distinguish them clearly from your scratch work and intermediate results.]

(*a*) Describe precisely all the absolute memory locations (hex values) where the return address of the vulnerable routine might appear and say how many there are (decimal value). [5 marks]

(*b*) Explain in detail how to compute n_{ret}, and do so (decimal value). [5 marks]

(*c*) Explain in detail how to compute n_{nop} and do so (decimal value), clarifying also why we need those nops in the first place. [5 marks]

(*d*) Explain in detail how to compute r, and do so (hex value). [5 marks]

7 Security

As an Elite Pentester, you must demonstrate the flaws of Megacorp's obsolete website, which was developed without any of the modern countermeasures against well-known attacks. The website includes a web bulletin board that allows users to post arbitrary text or HTML content, without any pre-filtering. Technically, a logged-in user posts a message by sending a `GET` request to `http://www.megacorp.com/bb/post.php`, with appropriate values for the parameters `Subject` and `Body`. Let *PAYLOAD* stand for a malicious piece of Javascript source code.

(*a*) Write a message (somehow including *PAYLOAD*) that, if posted on the bulletin board from an unprivileged guest account, would cause *PAYLOAD* to be executed on your victim's browser. Explain clearly how this would happen and who the victim would be. [4 marks]

(*b*) Briefly explain what cookies are and why an attacker might wish to steal them. Then, within *PAYLOAD*, write Javascript code to copy the victim's cookies into the `stolenCookies` variable. [4 marks]

(*c*) Within *PAYLOAD*, write Javascript code to exfiltrate the victim's cookies to you, the attacker, and explain clearly how your technique works. [4 marks]

(*d*) Using the techniques in parts (*a*)–(*c*), you have obtained the cookies of the CEO of Megacorp. As a demonstration of the vulnerability of Megacorp's website, write a raw HTTP request that will post the message "My account has been hacked", purporting to be from the CEO. Write also the command you would type to send the request to the server with `netcat`. [4 marks]

(*e*) You test your attack in part (*d*) and it works. Great. You delete the fake message immediately. You test it again successfully, and again delete the message. Then you tell your boss that you are ready for the big demo. You two have a meeting with the Megacorp CEO. You run your attack again but... it fails miserably. You are supremely embarrassed. Why did your attack work the first and second time but not the third? [4 marks]

SECTION B

8 Semantics of Programming Languages

Consider the following grammar of types and subtyping relation:

$$A, B \;::=\; \mathsf{int} \;\mid\; \{l_1 : A_1, \ldots, l_n : A_n\} \;\mid\; A \to B$$

$$\frac{}{\mathsf{int} \leq \mathsf{int}}\leq\mathsf{int}$$

$$\frac{B_1 \leq A_1 \qquad A_2 \leq B_2}{A_1 \to A_2 \leq B_1 \to B_2}\leq\to$$

$$\frac{n \leq m \qquad A_1 \leq B_1 \qquad \ldots \qquad A_n \leq B_n}{\{l_1 : A_1, \ldots, l_n : A_n\} \leq \{l_1 : B_1, \ldots, l_m : B_m\}}\leq\mathsf{rec}$$

We did not include explicit rules for reflexivity or transitivity in our subtyping relation. However, we can prove that they are *admissible* – that is, the subtyping judgement already has enough power to simulate them.

In the following questions, you may make use of any needed inversion properties without giving explicit proofs for them.

(*a*) Prove that for each type A, a subtyping derivation $A \leq A$ can be constructed. [5 marks]

(*b*) Prove that for all A, B, C, if $A \leq B$ and $B \leq C$ are derivable, then $A \leq C$ is derivable. [15 marks]

9 Semantics of Programming Languages

Consider the following small expression language:

$$
\begin{array}{lrcl}
\text{Expressions} & e & ::= & x \mid \mathsf{let}\, x = e\, \mathsf{in}\, e' \mid \underline{n} \mid e + e' \\
 & & \mid & \mathsf{true} \mid \mathsf{false} \mid \mathsf{if}\, e\, \mathsf{then}\, e'\, \mathsf{else}\, e'' \\
\text{Values} & v & ::= & \underline{n} \mid \mathsf{true} \mid \mathsf{false} \\
\text{Evaluation Contexts} & E & ::= & \mathsf{let}\, x = \Box\, \mathsf{in}\, e' \mid \Box + e \mid v + \Box \\
 & & \mid & \mathsf{if}\, \Box\, \mathsf{then}\, e'\, \mathsf{else}\, e'' \\
\text{Environments} & \rho & ::= & \cdot \mid \rho, x = v
\end{array}
$$

with the following reduction semantics:

$$\frac{\langle \rho; e\rangle \mapsto \langle \rho'; e'\rangle}{\langle \rho; E[e]\rangle \mapsto \langle \rho'; E[e']\rangle} \qquad \frac{x \notin \mathrm{dom}(\rho_1)}{\langle \rho_0, x = v, \rho_1; x\rangle \mapsto \langle \rho_0, x = v, \rho_1; v\rangle}$$

$$\frac{}{\left\langle \rho; \underline{n_1} + \underline{n_2}\right\rangle \mapsto \left\langle \rho; \underline{n_1 + n_2}\right\rangle} \qquad \frac{}{\langle \rho; \mathsf{let}\, x = v\, \mathsf{in}\, e\rangle \mapsto \langle \rho, x = v; e\rangle}$$

$$\frac{}{\langle \rho; \mathsf{if}\, \mathsf{true}\, \mathsf{then}\, e'\, \mathsf{else}\, e''\rangle \mapsto \langle \rho; e'\rangle} \qquad \frac{}{\langle \rho; \mathsf{if}\, \mathsf{false}\, \mathsf{then}\, e'\, \mathsf{else}\, e''\rangle \mapsto \langle \rho; e''\rangle}$$

(*a*) Give a set of typing rules for this language. [4 marks]

(*b*) Find an example expression which exhibits an anomalous execution behaviour, and briefly explain what the problem with the semantics is. [5 marks]

(*c*) Modify the semantics to repair the problem, and briefly explain how your solution works. [5 marks]

(*d*) Formulate progress and type preservation lemmas for the repaired language. (A proof is not necessary.) [6 marks]

END OF PAPER