UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

TYPES AND SEMANTICS FOR PROGRAMMING LANGUAGES

Thursday 1st January 1970

00:00 to 00:00

INSTRUCTIONS TO CANDIDATES

Answer QUESTION 1 and ONE other question.

Question 1 is COMPULSORY. If both QUESTION 2 and QUESTION 3 are answered, only QUESTION 2 will be marked.

All questions carry equal weight.

CALCULATORS MAY NOT BE USED IN THIS EXAMINATION

Year 4 Courses

Convener: ITO-Will-Determine
External Examiners: ITO-Will-Determine

1. THIS QUESTION IS COMPULSORY

Consider an indexed relation $m \neq n$ over natural numbers m and n, defined inductively.

$$\texttt{z≠s}\ \frac{}{\texttt{zero} \neq \texttt{suc}\ n} \qquad \texttt{s≠s}\ \frac{}{\texttt{suc}\ m \neq \texttt{zero}} \qquad \texttt{s≠s}\ \frac{m \neq n}{\texttt{suc}\ m \neq \texttt{suc}\ n}$$

Recall that $m \not\equiv n$ stands for $\neg(m \equiv n)$, where $\equiv$ is equality as defined in the Agda standard library. Note that $\neq$ and $\not\equiv$ are distinct symbols. (The Unicode symbols are typed as `\=n` and `\==n`, respectively.)

(a) Formalise the definition above. [*12 marks*]

(b) Prove $m \neq n$ implies $m \not\equiv n$, and conversely, for all natural numbers m and n. [*13 marks*]

2. ANSWER EITHER THIS QUESTION OR QUESTION 3

You will be provided with a definition of intrinsically-typed lambda calculus in Agda. Consider constructs satisfying the following rules, written in extrinsically-typed style.

The rules replace the given definition of μ (fixpoint). In the given definition terms of the form $\mu\ f \Rightarrow N$ are not values but may have any type. In contrast, in the definition below terms of the form $\mu\ f\ \lambda\ x \Rightarrow N$ are values but must have function type.

Typing:

$$\mu\lambda \frac{\Gamma\ ,\ f ⦂ A \Rightarrow B\ ,\ x ⦂ A \vdash N ⦂ B}{\Gamma \vdash \mu\ f\ \lambda\ x \Rightarrow N ⦂ A \Rightarrow B}$$

Values:

$$\texttt{V-}\mu\lambda \frac{}{\texttt{Value}\ (\mu\ f\ \lambda\ x \Rightarrow N)}$$

Reduction:

$$\beta\texttt{-}\mu\lambda \frac{\texttt{Value}\ V}{(\mu\ f\ \lambda\ x \Rightarrow N) \cdot V \longrightarrow N\ [\ f\ := (\mu\ f\ \lambda\ x \Rightarrow N)\ ]\ [\ x\ := V\ ]}$$

(a) Update the given definition to formalise the above rules, including any other required definitions. [*12 marks*]

(b) Prove progress. You will be provided with a proof of progress for the simply-typed lambda calculus that you may revise. Confirm your work by updating the given example. [*13 marks*]

Please delimit any code you add as follows.

```
-- begin
-- end
```

3. ANSWER EITHER THIS QUESTION OR QUESTION 2

You will be provided with a definition of inference for extrinsically-typed lambda calculus in Agda. Consider constructs satisfying the following rules, written in extrinsically-typed style that support bidirectional inference.

See Question 2 to motivate replacing terms of the form $\mu\ f\ \Rightarrow\ N$ by terms of the form $\mu\ f\ \lambda\!\!\!\!\lambda\ x \Rightarrow N$.

Typing:

$$\vdash\mu\lambda\!\!\!\!\lambda\ \frac{\Gamma\ ,\ f \mathbin{⦂} A \Rightarrow B\ ,\ x \mathbin{⦂} A \vdash N \downarrow B}{\Gamma \vdash \mu\ f\ \lambda\!\!\!\!\lambda\ x \Rightarrow N \downarrow A \Rightarrow B}$$

(a) Update the given definition to formalise the above typing rule. [*10 marks*]

(b) Update the code to support type inference for the above typing rule. Confirm your work by updating the given example. [*15 marks*]

Please delimit any code you add as follows.

```
-- begin
-- end
```