Logic and Verification
Informatics
Technical University of Munich

Eexam
Place student sticker here

Note:

- During the attendance check a sticker containing a unique code will be put on this exam.
- This code contains a unique number that associates this exam with your registration number.
- This number is printed both next to the code and to the signature field in the attendance check list.

Lambda Calculus

Exam:	IN2358 / Endterm	**Date:**	Friday 16^{th} February, 2024
Examiner:	Prof. Tobias Nipkow	**Time:**	11:00 – 12:30

	P 1	P 2	P 3	P 4	P 5
I					

Working instructions

- This exam consists of **12 pages** with a total of **5 problems**.
  Please make sure now that you received a complete copy of the exam.
- The total amount of achievable credits in this exam is 40 credits.
- Detaching pages from the exam is prohibited.
- Allowed resources:
  - one **DIN A4 sheet** with **hand-written** notes on both sides
  - one **analog dictionary** English ↔ native language
- Subproblems marked by * can be solved without results of previous subproblems.
- **Answers are only accepted if the solution approach is documented.** Give a reason for each answer unless explicitly stated otherwise in the respective subproblem.
- Do not write with red or green colors nor use pencils.
- Physically turn off all electronic devices, put them into your bag and close the bag.

Left room from __________ to __________ / Early submission at __________

Problem 1 Programming with λ-terms (8 credits)

0 1 2 3 4

a)* We want to define optionals, i.e. we want to define λ-terms none, some, mapopt, and fromopt with the following behavior:

- fromopt d none $=_\beta d$
- fromopt d (some a) $=_\beta a$
- mapopt f none $=_\beta$ none
- mapopt f (some a) $=_\beta$ (some $(f\ a)$)

Complete the set of definitions below such that the above conditions are fulfilled.

- fromopt $:= (\lambda d\ o.\ o\ d\ (\lambda x.\ x))$
- none $:= (\lambda d\ f.\ d)$
- some $:=$
- mapopt $:=$

0 1 2 3 4

b)* Using the fixed point combinator fix, define a λ-term these such that it holds that

- these (cons none t) $=_\beta$ these t
- these (cons (some a) t) $=_\beta$ cons a (these t)
- these nil $=_\beta$ nil

Your implementation must not rely on the underlying encoding of lists but you may use the functions nil, cons, null, head, tail, and append with the usual semantics. Remember that null returns a boolean, i.e. null nil $x\ y \to_\beta^* x$ and null (cons $h\ t$) $x\ y \to_\beta^* y$.

Hint: Define a function that converts a single optional into a list first.

- these $:=$

Problem 2 Rewriting (8 credits)

Let $\to_1, \to_2 \subseteq A \times A$ be two relations such that $\to_1^* \subseteq \to_2^*$. Properties C1 and C2 are defined as follows:

C1: $\forall a\ b.\ a \to_2^* b \Longrightarrow \exists c.\ a \to_1^* c \wedge b \to_1^* c$

C2: $\forall a\ b\ c.\ a \to_2 b \to_1^* c \Longrightarrow \exists d.\ a \to_1^* d \wedge c \to_1^* d$

Note: implicitly all variables are assumed to be elements of A.

Prove that C1 if and only if C2.

The proof must be given in the standard verbal style. However, it may be helpful to draw a diagram, in particular as a starting point.

0
1
2
3
4
5
6
7
8

Problem 3 **Quiz (9 credits)**

0 1 2

a)* Is there any λ-term F in System F such that $F\ g \rightarrow_\beta^* g\ (F\ g)$?

0 1 2 3

b)* Give a proposition P and two simply typed λ-terms s and t such that

- $s \neq_{\alpha\beta\eta} t$ where $=_{\alpha\beta\eta}$ is defined as $(=_\alpha \cup =_\beta \cup =_\eta)^*$, and
- both s and t prove P in intuitionistic logic.

0 1 2 3 4

c)* We consider intuitionistic logic with implication, conjunction, and disjunction in this exercise. Additionally, we consider $\perp$E, i.e. we have the rule

$$\frac{\Gamma \vdash t : \perp}{\Gamma \vdash \varepsilon\ t : A} \perp\mathrm{E}$$

where ε is some fixed constant. Give a lambda term that proves the proposition $(A \vee B) \rightarrow \neg A \rightarrow B$ where $\neg A$ is an abbreviation for $A \rightarrow \perp$.

Problem 4 Typing (9 credits)

In this exercise, you are tasked to solve different typing problems by drawing a type derivation tree. For each node in such a tree, **annotate** it with the typing rule that you used.

a)* Find most general types ?1, ?2 and ?3 that solve the typing problem

0 1 2 3

$$x : ?1, y : ?2 \vdash (x\ (y\ x)) : ?3$$

in simply typed lambda calculus and draw the type derivation tree (with the right ?1, ?2 and ?3).

Use only the introduction and elimination rules for $\to$ and $\forall$, the rule for `let`, and the assumption rule $\Gamma \vdash x : \Gamma(x)$ for the following exercises.

b)* Find the most general *type schema* $?\sigma$ such that $x : A \vdash \lambda y.\ x : ?\sigma$.

0 1 2

c) Now, find the most general *type* $?\tau$ and complete the following derivation tree by filling in the blanks that are underlined with dots. Each leaf of the tree must be closed by the assumption rule or part b).

$$\dfrac{\dfrac{\dfrac{\text{Part b)}}{x : A \vdash \lambda y.\ x : ?\sigma} \qquad \ldots\ldots\ldots\ldots\ldots\ldots}{\ldots\ldots\ldots\ldots \vdash \texttt{let}\ y = \lambda y.\ x\ \texttt{in}\ (y\ x) : \ldots\ldots\ldots\ldots}\ \ldots\ldots\ldots}{\vdash \lambda x.\ \texttt{let}\ y = \lambda y.\ x\ \texttt{in}\ (y\ x) : \ldots\ldots\ldots\ldots = ?\tau}\ \ldots\ldots\ldots$$

Problem 5 Intuitionistic Logic (6 credits)

In this exercise, we consider intuitionistic logic $\vdash_I$ with negation, conjunction, and disjunction extended with the exclusive or operator $\oplus$ to obtain the logic $\vdash_\oplus$. We can translate formulae in this logic to intuitionistic logic with the following function:

$$\begin{aligned} A^* &= A \qquad\qquad \text{for atomic } A \\ (A \oplus B)^* &= (A^* \wedge \neg B^*) \vee (\neg A^* \wedge B^*) \\ (\neg A)^* &= \neg A^* \\ (A \vee B)^* &= A^* \vee B^* \\ (A \wedge B)^* &= A^* \wedge B^* \end{aligned}$$

We use the following rules for $\oplus$ in $\vdash_\oplus$:

$$\frac{\Gamma \vdash_\oplus A \qquad \Gamma \vdash_\oplus \neg B}{\Gamma \vdash_\oplus A \oplus B}\,\oplus\mathrm{I}_1 \qquad\qquad \frac{\Gamma \vdash_\oplus A \oplus B \qquad \Gamma, A \wedge \neg B \vdash_\oplus C \qquad \Gamma, \neg A \wedge B \vdash_\oplus C}{\Gamma \vdash_\oplus C}\,\oplus\mathrm{E}$$

a)* Write down the missing introduction rule $\oplus\mathrm{I}_2$.

0
1

b)* Prove that $\Gamma \vdash_\oplus A \implies \Gamma^* \vdash_I A^*$ by induction on the derivation of $\Gamma \vdash_\oplus A$ where Γ^* means that we apply the function $-^*$ pointwise. You only need to consider the cases where $\Gamma \vdash_\oplus A$ was proved by $\oplus\mathrm{I}_1$or $\oplus\mathrm{E}$.
Explicitly **annotate** each logical inference rule or induction hypothesis that you use.
Note: You don't need to use any rules for negation.

0
1
2
3
4
5

Additional space for solutions–clearly mark the (sub)problem your answers are related to and strike out invalid solutions.