Automatic Program Analysis

Final Exam

Mon, June 27, 2022, 17:00–20.00

- Please make sure that you write your name and student number on each sheet of paper that you hand in. On the first sheet of paper, indicate the total number of sheets handed in.
- This is a closed-book exam: it is not allowed to consult lecture notes, books, etc. Whenever reference material is required it is given as part of the exam.
- Always explain the reasoning behind your answers, even when not explicitly asked for.
- Each question includes the amount of points which can be obtained when done correctly, adding up to a total of 100.

---

1 Monotone Frameworks

Question 1. (9 points) An instance of a monotone framework is denoted as $(L, \mathcal{F}, F, E, \iota, \lambda l.\ f_l)$. What do all the symbols mean? One name or a few words per element of this tuple is sufficient.

Question 2. (6 points) For distributive instances, the MFP algorithm gives the exact same solution as the theoretical Meet Over all Paths. Why is such an analysis still an approximation? Where do we lose precision?

Question 3. (7 points) Defend or attack this statement: a monotone framework is forward if and only if it is about properties of values, and backward if and only if it is about properties of computations.

2 Fixed-points

Consider the lattice $L = \{[x, y] \mid x, y \in \mathbb{Z} \cup \{-\infty, \infty\},\ \ x \leqslant y\} \ \cup\ \{[\infty, -\infty]\}$, of intervals with integrals, ∞ and $-\infty$ boundaries. We add a special interval $\bot = [\infty, -\infty]$ to denote the empty interval. We define that $k \sqsubseteq l$ denotes that interval k is contained in interval l:

$$[i_1, j_1] \sqsubseteq [i_2, j_2] \ \text{ if } \ i_1 \geqslant i_2 \ \text{ and } \ j_1 \leqslant j_2$$

Question 4. (8 points) Answer the following questions with 'yes' or 'no'. Explanation of the answer is not needed.

(a) Does the ascending chain condition hold on this lattice?

(b) Does the descending chain condition hold on this lattice?

(c) Is fixed-point iteration for a monotone function, starting with $\bot$, guaranteed to terminate?

(d) Is fixed-point iteration for a monotone function, starting with $\top$, guaranteed to terminate?

We define $B = \{2^n \mid n \in \mathbb{N}\} \cup \{-2^n \mid n \in \mathbb{N}\}$ as the set of powers of two, and the negative numbers of those. Now consider this attempt at creating a widening operator:

$$[i_1, j_1] \nabla [i_2, j_2] = \begin{cases} [i_2, j_2] & \text{if } [i_1, j_1] = \bot \\ [LB(i_1, i_2), UB(j_1, j_2)] & \text{otherwise} \end{cases}$$

where

$$LB(x, y) = \begin{cases} x & \text{if } x \leqslant y \\ \max\{b \mid b \in B, b \leqslant y\} & \text{otherwise} \end{cases}$$

and

$$UB(x, y) = \begin{cases} x & \text{if } x \geqslant y \\ \min\{b \mid b \in B, b \geqslant y\} & \text{otherwise} \end{cases}$$

I.e., we compute the smallest interval containing both intervals only using the boundaries of the left interval and values from B. We have a special case when the left interval is bottom.

Question 5. (7 point) Is this operator a widening operator? Motivate your answer.

3 Control Flow Analysis

For control flow analysis we made an annotated type system with types $\hat{\tau} := \hat{\tau}_1 \to^{\phi} \hat{\tau}_2 \mid ....$ We incorporated subeffecting as follows:

$$\frac{\hat{\Gamma} \vdash_{CFA} t : \hat{\tau}_1 \to^{\phi} \hat{\tau}_2}{\hat{\Gamma} \vdash_{CFA} t : \hat{\tau}_1 \to^{\phi \cup \phi'} \hat{\tau}_2} \text{ [CFA-SUB]}$$

Question 6. (4 points) This rule is not syntax-directed. Give the names of two other typing rules that are also not syntax-directed. No explanation is needed.

Consider we want to extend this to subtyping, where we can modify annotations deeper in the type. We can express that in terms of a subtyping relation $\leqslant$ as follows:

$$\frac{\hat{\Gamma} \vdash_{CFA} t : \hat{\tau} \quad \hat{\tau} \leqslant \hat{\tau'}}{\hat{\Gamma} \vdash_{CFA} t : \hat{\tau'}} \text{ [CFA-SUB]}$$

Question 7. (8 points) Define the subtyping relation $\leqslant$ relation for a function type $\hat{\tau}_1 \to^{\phi} \hat{\tau}_2$.

4 Side Effect Analysis

Question 8. (10 points) Give the typing rules for lambda abstraction and function applications in side effect analysis. No explanation is needed.

Question 9. (6 points) Explain how the effects of a lambda abstraction are propagated to a call to that function, by referring to those typing rules.

5 Program Transformation

Question 10. (10 points) Algorithm W can be extended to perform program transformation. Why does this require a multi-pass (multiple passes over the program)? I.e., why is it not possible to perform this in a single pass over the program?

6 Guest lectures

Question 11. (5 points) In constant time cryptography, certain operations are not allowed on secret, highly confidential, values. Name two different operations that are not allowed, for instance the two that we encoded in the type system during the lecture.

Question 12. (5 points) Give and explain a short example (one or two lines of code) where the greedy approach of Algorithm W results in more type errors than needed.

7 Galois Connections

Consider we have Galois connections $(L, \alpha_1, \gamma_1, \texttt{Bool})$ and $(L, \alpha_2, \gamma_2, M)$. We now want to construct a Galois connection $(L, \alpha, \gamma, \texttt{Maybe}\ M)$, where `Bool` is the set $\{\text{True}, \text{False}\}$ and `Maybe` is the data type Maybe from Haskell.

A concrete value $c \in L$ is mapped to Just if $\alpha_1(c)$ is True, as follows:

$$\alpha(c) = \begin{cases} \text{Just } (\alpha_2(c)) & \text{if } \alpha_1(c) \\ \text{Nothing} & \text{otherwise} \end{cases}$$

Question 13. (8 points) Define γ in terms of γ_1 and γ_2. Multiple definitions of γ are possible, give the most precise one.

Question 14. (7 points) If $(L, \alpha_1, \gamma_1, \texttt{Bool})$ and $(L, \alpha_2, \gamma_2, M)$ are Galois insertions, is $(L, \alpha, \gamma, \texttt{Maybe}\ M)$ then also a Galois insertion? Prove this statement or give a counter example.