Automatic Program Analysis

Final Exam Answers

Mon, June 27, 2022, 17:00–20.00

1 Monotone Frameworks

Question 1. (9 points)

- L: Lattice
- $\mathcal{F}$: Function space, set of (monotone) transfer functions or subset of $L \to L$ of monotone functions closed under composition with identity.
- F: Flow of the program.
- E: Extremal labels.
- ι: Extremal value of the analysis.
- f_l: Transfer function (at label l).

1.5 point per item, rounded down.

Question 2. (6 points) MFP now gives an exact solution given the transfer functions. We do not loose precision when joining values, but we can still loose precision in the transfer functions and/or the choice of the lattice.

Question 3. (7 points) There is no strict relation between the direction of the analysis and whether the analysis is about values or computations. Some examples:

- Available expressions: property of computations, forward
- Constant propagation: property of values, forward
- Live variables: property of computations, backward

- 2 points for choosing rejection.
- 5 points for reasoning.

2 Fixed-points

Question 4. (8 points)

(a) No

(b) No. Initial grading scheme had 'yes' as answer, hence yes and no are both marked as correct.

(c) No

(d) Yes and no both marked as correct. The question was ambiguous whether termination was guaranteed in general, or for this specific lattice.

2 points per subquestion.

Question 5. (7 point) No. Consider the infinite sequence $[0,0], [0,1], [0,2], \ldots$. A widening operator would project this to a sequence that stabilizes at some point, but this widening operator will use arbitrary large powers of two: $[0,0], [0,1], [0,2], [0,4], [0,8], [0,16], \ldots$ (duplicate entries in the sequence omitted). Hence it doesn't satisfy the second axiom of a widening operator.

- 2 points for choosing no.
- 5 points for reasoning.
- When answering yes, and showing that axiom 1 of a widening operator holds, then 3 points.

3 Control Flow Analysis

Question 6. (4 points) generalise, instantiate

2 points per answer.

Question 7. (8 points)

$$\frac{\hat{\tau}_1' \leqslant \hat{\tau}_1 \quad \hat{\tau}_2 \leqslant \hat{\tau}_2'}{\hat{\tau}_1 \to^{\phi} \hat{\tau}_2 \leqslant \hat{\tau}_1' \to^{\phi \cup \phi'} \hat{\tau}_2} \text{ [CFA-SUB]}$$

or

$$\frac{\hat{\tau}_1' \leqslant \hat{\tau}_1 \quad \hat{\tau}_2 \leqslant \hat{\tau}_2' \quad \phi \subset \phi'}{\hat{\tau}_1 \to^{\phi} \hat{\tau}_2 \leqslant \hat{\tau}_1' \to^{\phi'} \hat{\tau}_2} \text{ [CFA-SUB]}$$

- 2 points for general shape.
- 2 points for subset relation on annotation (ϕ).
- 2 points for covariant subtyping on $\hat{\tau}_2$.
- 2 points for contravariant subtyping on $\hat{\tau}_1$.

4 Side Effect Analysis

Question 8. (10 points) Type rules for app and abstractions for side effect analysis, see the slides.

- 2 points: part corresponding to underlying type rule of abstractions.
- 2 points: part corresponding to underlying type rule of applications.
- 1 point: abstraction has the empty effect

- 2 points: effect of body of abstraction is stored on arrow
- 1 points: effect of applied function is given
- 2 points: effects of subterms of an application

Question 9. (6 points) A function abstraction stores its effects in the type. When the function is applied, then those effects are returned.

5 Program Transformation

Question 10. (10 points) When Algorithm W visits a node, we don't know its final type yet. More substitutions may follow later on, hence we cannot perform the transformation directly. Hence we do another pass over the program when we know the complete substitution.

6 Guest lectures

Question 11. (5 points) Two out of:

- Array indexing
- Branching
- Variable time operations as integer division

Question 12. (5 points) `[True, 1, 2, 3, 4]`. Here we could get 4 errors saying that an integer isn't a boolean. Algorithm W is greedy and directly assumes that the list is a list of booleans after seeing the first element (assuming it goes left-to-right).

- 2 points for the example
- 3 points for explanation

7 Galois Connections

Question 13. (8 points)

$$\alpha(a) = \begin{cases} \gamma_1(True) \sqcap \gamma_2(a') & \text{if } a = \text{Just } a' \\ \gamma_1(False) & \text{if } a = \text{Nothing} \end{cases}$$

- 5 points if you forgot γ_1 in both cases.
- 7 points if you forgot $\gamma_1(True)$.
- 8 points if entirely correct.

Question 14. (7 points) It is not guaranteed to be an insertion. For instance if $M = \text{Bool}$, $\alpha_1 = \alpha_2$, $\gamma_1 = \gamma_2$, and α_1 non-constant, then Just False won't be mapped to itself.

- 2 points for ‘no’
- 2 points for counter example
- 3 points for explanation of the example