

Exercise Sheet 10: Correctness Properties for Execution Time

Due date: July 8th. You can hand in your solutions at the start of the exercise class.

Exercise 1 (Timed correctness in non-deterministic programs)

10%

Extend the proof system of Hoare logic for timed correctness to incorporate a demonic model of the non-deterministic operator $c_1 \sqcap c_2$. (In the demonic model, all possible program executions must establish the postcondition and satisfy the time bound).

Solution:

$$\frac{\{A\} c_1 \{e \Downarrow B\} \quad \{A\} c_2 \{e \Downarrow B\}}{\{A\} c_1 \sqcap c_2 \{e \Downarrow B\}}$$

Exercise 2 (Alternative while Rule)

20%

Suggest a rule for $\text{while } (b) \text{ do } \{c\}$ that expresses that its execution time, neglecting constant factors, is the product of the number of times the loop is executed and the maximal execution time for the body of the loop.

Solution:

$$\frac{\{i \geq 0 \wedge A(i+1)\} c \{j \Downarrow A(i)\}}{\{\exists i \exists j : A(i) \wedge e \leq i \cdot j\} \text{ while } (b) \text{ do } \{c\} \{e \Downarrow A(0)\}}$$

Exercise 3 (Completeness of Hoare logic for timed correctness)

25%

Prove or disprove: There exists a valid total correctness property $\{A\} c \{\Downarrow B\}$ such that for every $e \in \text{AExp}$, the timed correctness property $\{A\} c \{e \Downarrow B\}$ is *not* valid.

Solution: We prove that the statement is correct by providing a suitable total correctness property.

Let $c \equiv y := 1; \text{while } (x \neq 0) \text{ do } \{x := x - 1; y := y \cdot x\}; \text{while } (y \neq 0) \text{ do } \{y := y - 1\}$. If x is larger than 0, the first loop of c computes the factorial of x and stores it in y . The second loop then decrements variable y exactly $y = x!$ times. Hence, the running time of c is in $\mathcal{O}(x!)$.

Then $\{x > 0\} c \{\Downarrow y = 0\}$ is a valid total correctness property. Now, e is a (finite) arithmetic expression that may make use of variables x and y which denote their *initial* valuation. However, $x!$ cannot be expressed by any finite arithmetic expression (as defined in the lecture). Hence, there exists no $e \in \text{AExp}$ such that $\{x > 0\} c \{e \Downarrow y = 0\}$ is valid.

Exercise 4 (Correctness Properties for Lower Execution Time Bounds)

45%

In the lecture, we considered a calculus to prove upper bounds on the execution time of programs.

- (a) [25%] Modify the Hoare logic for timed correctness from the lecture to prove *lower* execution time bounds instead of upper bounds. To be more precise, a lower bound

correctness property $\{A\}c\{e \uparrow B\}$ is valid if there exists $k > 0$ such that for each $I \in Int$, $\sigma, \sigma' \in \Sigma$ and $\tau \in \mathbb{N}$, $\langle c, \sigma \rangle \xrightarrow{\tau} \sigma'$ implies $\tau \geq k \cdot \mathfrak{A}[e]$ and $\sigma' \models^I B$.

Solution: The rules for **skip**, assignments and **if** are as presented in the lecture. Thus, it suffices to present lower bound execution time rules for sequential composition, **while** and the rule of consequence.

$$\begin{aligned}
 (seq) & \frac{\{A \wedge e'_2 = u\} c_1 \{e_1 \uparrow C \wedge e_2 \geq u\} \quad \{C\} c_1 \{e_2 \uparrow B\}}{\{A\} c_1; c_2 \{e_1 + e'_2 \uparrow B\}} \\
 (while) & \frac{\{A \wedge b \wedge e' = u\} c \{e_0 \uparrow A \wedge e \geq u\}}{\{A\} \text{while } (b) \text{ do } \{c\} \{e \uparrow A \wedge \neg b\}} \quad \text{where } e \leq e_0 + e' \\
 (cons) & \frac{\models (A \Rightarrow A' \wedge \exists k \in \mathbb{N} : e' \geq k \cdot e) \quad \{A'\} c \{e' \uparrow B'\} \quad \models (B' \Rightarrow B)}{\{A\} c \{e \uparrow B\}}
 \end{aligned}$$

- (b) [5%] For upper execution time bounds, we considered total correctness properties only. Why are we considering partial correctness properties instead for lower bound execution time bounds?

Solution: If we can prove a (global) upper execution time bound for a given program then termination is guaranteed. Hence, partial correctness properties make no sense in the presence of upper execution time bounds. In contrast, arbitrary lower bounds can be proven for non-terminating programs.

- (c) [5%] Is there a postcondition $e \uparrow B$ such that $\{A\}c\{e \uparrow B\}$ universally holds regardless of the choice of A and c ?

Solution: Choose $B \equiv \text{true}$ and e as the trivial lower bound 1. Since every program requires at least constant execution time, we have $\models \{A\}c\{1 \uparrow \text{true}\}$.

- (d) [10%] Using your Hoare calculus for lower execution time bounds, prove that

$$\{\text{true}\} \text{while } (\text{true}) \text{ do } \{\text{skip}\} \{2 \uparrow N \uparrow \text{true}\}$$

is valid for each $N \in \mathbb{N}$ ¹. *Note:* You may assume $\uparrow$ to be a given functional symbol, i.e. you do not have to define it in Hoare logic first.

Solution: Let $N \in \mathbb{N}$ be fixed.

$$\begin{aligned}
 & \frac{\models (\text{true} \wedge \text{true} \wedge e' = (2 \uparrow N) - 1 \Rightarrow \text{true} \wedge \exists k : 1 \geq k \cdot 1) \quad \{\text{true}\} \text{skip} \{1 \uparrow \text{true}\}}{\{\text{true} \wedge \text{true} \wedge e' = (2 \uparrow N) - 1\} \text{skip} \{1 \uparrow \text{true}\}} \\
 & \frac{\{\text{true}\} \text{while } (\text{true}) \text{ do } \{\text{skip}\} \{2 \uparrow N \uparrow \text{true} \wedge \neg \text{true}\} \quad \models (\text{true} \wedge \neg \text{true} \Rightarrow \text{true})}{\{\text{true}\} \text{while } (\text{true}) \text{ do } \{\text{skip}\} \{2 \uparrow N \uparrow \text{true}\}}
 \end{aligned}$$

¹Knuth's arrow notation is defined recursively as $a \uparrow 1 := a$ and $a \uparrow (b + 1) := a^{a \uparrow b}$ where $a, b \in \mathbb{N}$.