UNIVERSITETET I OSLO
Institutt for Informatikk
PMA
Einar Broch Johnsen, Martin Steffen

INF 4140: Models of Concurrency

Høst 2014 **Series 10** 12. 11. 2013

Topic: Local reasoning

Issued: 12. 11. 2013

Exercise 1 (The Coin Machine Example Revisited) Consider the coin machine example from the slides to lecture 11.

(a) Verification conditions for the Hoare analysis
Convince yourself that the verification conditions on slide 17 and 18 are satisfied.

(b) Local history invariant
Consider the history invariant $I_C(h)$ for the coin machine (slide 23). Ensure that this invariant is satisfied after the two receive statements in the coin machine implementation (on slide 14).

A weakness with the history invariant $I_C(h)$ is that it does not express that once the balance b gets greater than 10, the coin machine will immediately send a *ten* message. Thus, even though this property is satisfied by the implementation on page 14, it is not reflected by the history invariant. (The invariant $I_C(h)$ will be satisfied by an implementation that accepts *one* messages as long as the balance is less than 15.) In the next exercise we therefore consider a stronger history invariant.

(c) Strengthened history invariant
Consider the local history invariant $I_C(h)$ defined by:

$$\begin{array}{lcl} I_C(\varepsilon) & = & \mathit{true} \\ I_C(h; (U{\downarrow}C : \mathit{five})) & = & 0 \leq \mathit{diff}(h) < 10 \\ I_C(h; (U{\downarrow}C : \mathit{one})) & = & 0 \leq \mathit{diff}(h) < 10 \\ I_C(h; (C{\uparrow}U : \mathit{ten})) & = & 10 \leq \mathit{diff}(h) < 15 \end{array}$$

where $\mathit{diff}(h)$ is defined by $\mathit{sum}(h/\downarrow) - \mathit{sum}(h/\uparrow)$. The task here is to prove this local history invariant, reusing the loop invariants from the slides: you must prove that the local history invariant is satisfied after each send and receive statement.

Hint: If the local condition is not strong enough to prove the local history invariant after a send/receive statement, you may instead make a new Hoare proof, proving that the precondition you have in the loop gives you the local invariant at the place where it should hold. For instance, for the case of receiving one, you may prove that the inner loop invariant together with the negation of the loop test before the await one-statement gives the local invariant after.

Exercise 2 (The Mini Bank Example (as far as time allows)) Here we consider the Mini bank (ATM) example from Lecture 11. To simplify the mini bank, we omit all messages and information about pin codes. Thus, pin messages between M and C are ignored. (The bank assumes the the clients are honest.) For instance, $Cycle_M$is simplified to:

$[\ C{\downarrow}M : card\_in(n), M{\uparrow}C : amount, C{\downarrow}M : amount(y),$
$\quad$ **if** $y \leq 0$ **then** ε **else**
$\quad M{\uparrow}B : request(n, y), [B{\downarrow}M : deny \mid B{\downarrow}M : grant, M{\uparrow}C : cash(y)\ ]$ **fi** ,
$\quad M{\uparrow}C : card\_out$ **some** $C, n, y\ ]^*$

(a) Mini bank implementation
Make an implementation of the simplified mini bank M by means of a ”while true do ... od” loop. You may use the choice operator [] as explained in Lecture 10 and 11. Program such that h **is** $Cycle_M$ is the loop invariant.

Hint: In order to express that M is waiting for either a deny message or a grant message from B, you may use the programming construct $(S[]S')$ where S and S' are statement lists, and [] denotes choice. In particular, the construct (**await** $msg; S$ [] **await** $msg'; S'$) will await the first message matching msg or msg' and select that branch, while skipping the other branch. For instance, the statement

(**await** $B : grant; \ldots$ [] **await** $B : deny; \ldots$)

will let you wait for either an incoming grant message or a deny message.

(b) Entry condition
Show that the loop invariant holds upon entry of the loop.

(c) Invariance
Verify by the Hoare Logic from Lecture 11, that the loop invariant is maintained by each iteration of the loop.

(d) Proof of local history invariant
Prove that the local mini bank history invariant $h \leq Cycle_M$ is satisfied after each send and receive statement, by proving

$$Q \Rightarrow (h \leq Cycle_M)$$

for each postcondition Q of a send/receive statement.
Remark that $(h; x \leq Cycle_M)$ implies $(h \leq Cycle_M)$ which may simplify the proof.

(e) Improved mini bank
The slides from Lecture 11 presented local history invariants for three kinds of agents: the client (C), the mini bank (M), and the central bank (B), and a global invariant was found.

The mini bank could be improved by introducing two new events *cash_taken* from C to M, and *card_taken* from C to M, representing the removal of cash by the client, and the removal of card by the client, respectively. Assume as above that pin codes are ignored.

1. Write a new local invariant for Clients, ensuring that a *card_out* event is followed by a *card_taken* event, and similarly that a *cash_out* event is followed by a *cash_taken* event.

2. Rewrite the invariants for the central bank and the mini bank, adding behavior for the new events and such that all information and messages about pin-codes are removed. (You do not need to modify the above implementation of M or redo the Hoare analysis.)
3. Use the composition rule to find a global invariant for the system with one client, one mini bank and one central bank.
4. Compare the result with the global invariant from the slides.