Universitetet i Oslo
Institutt for Informatikk

PMA

Olaf Owe, Martin Steffen

INF 4140: Models of Concurrency

Høst 2015

Series 9

17. 11. 2015

Topic: Histories

Issued: 17. 11. 2015

Exercise 1 (History functions) Using the techniques on the slides from lecture 10, define the following functions over histories:

1. a Boolean function *endswith*: $\mathit{Hist} \times \mathit{Set} \rightarrow \mathit{Bool}$ such that h *endswith* A is true if h is nonempty and ends with an event in the set A. For instance, *abcd endswith* $\{b, c\}$ is *false*, ϵ *endswith* $\{b, c\}$ is *false*, and $[a, b, c, d]$ *endswith* $\{b, d\}$ is *true.*
2. a Boolean function *beginsswith*: $\mathit{Hist} \times \mathit{Set} \rightarrow \mathit{Bool}$ such that h *beginsswith* A is *true* if h is nonempty and begins with an event in the set s. For instance, $[a, b, c, d]$ *beginsswith* $\{b, c\}$ is *false*, ϵ *beginsswith* $\{b, c\}$ is false, and $[a, b, c, d]$ *beginsswith* $\{b, a\}$ is true.
3. a Boolean function testing if one history is a subsequence of another history, $\_ \sqsubseteq \_ :$ $\mathit{Hist} \times \mathit{Hist} \rightarrow \mathit{Bool}$. For instance, $[b, d, e] \sqsubseteq [a, b, c, d, e]$, but $[b, e, d] \not\sqsubseteq [a, b, c, d, e]$.
4. a function $\_ \backslash \_ : \mathit{Hist} \times \mathit{Set} \rightarrow \mathit{Hist}$ such that $h \backslash A$ is the subsequence of h consisting of all events *not* in the set A. For instance, $[a, b, c, b, d, a] \backslash \{d, c\}$ is $[a, b, b, a]$.
5. a function *pending* : $\mathit{Hist} \rightarrow \mathit{Hist}$ such that $\mathit{pending}(h)$ is the sequence of all send messages that not yet have been received by the partner.

   For instance, $\mathit{pending}([A{\uparrow}B{:}m_1, A{\uparrow}B{:}m_2, A{\uparrow}B{:}m_1, A{\downarrow}B{:}m_1])$ is $[A{\uparrow}B{:}m_1, A{\uparrow}B{:}m_2]$. (In case there are several identical send messages, and some but not all of these have been received, you may choose the order of the remaining ones as you wish. For instance, the example above could give the result $[A{\uparrow}B{:}m_2, A{\uparrow}B{:}m_1]$.)

   Hint: here you need to distinguish between send and receive events in the definition, and you may need to introduce an additional function.

Exercise 2 (Coin machine users) Consider the coin machine from the lecture, where the *history invariant* for the coin machine C is defined over the global history H by:

$$I_C(H/\alpha_C) \triangleq 0 \leq \mathit{sum}(H/\downarrow C) - \mathit{sum}(H/C\uparrow) < 15 \ . \tag{1}$$

In the lecture, a coin machine agent C was composed with a user agent U with exact change. We will here consider the composition of C with two different users, U_1 and U_2.

(a) User U_1 inserts only “5 krone” coins, i.e., U_1 only sends messages “`five`”. The (outside behavior of the) user is specified by the following invariant:

$$I_{U_1}(H/\alpha_{U_1}) \triangleq H/\{U_1\uparrow\colon \texttt{one}\} = \epsilon \wedge sum(H/U_1\uparrow) - sum(H/\downarrow U_1) \in \{0, 5, 10\} \quad (2)$$

Write down the *global* invariant for the system consisting of C and U_1. Is it possible to use the function “legal” to simplify this invariant? For instance, may we say something more precise about the difference $sum(H/\downarrow C) - sum(H/C\uparrow)$ compared to what we know from I_C?

(b) User U_2 sends both five and one messages to the coin machine, but U_2 never cares about collecting the coins returned by the machine. User U_2 is specified by the following invariant:

$$I_{U_2}(H/\alpha_{U_2}) \triangleq 0 \leq sum(H/U_2\uparrow) \wedge sum(H/\downarrow U_2) = 0\ . \quad (3)$$

Write down the global invariant for the system consisting of C and U_2. Is it possible to use the legal function to simplify this global invariant?