Safety = invariants + history variables

Wim H. Hesselink, whh472, September 4, 2012

Dept. of Computing Science, University of Groningen
P.O.Box 407, 9700 AK Groningen, The Netherlands

1 Introduction

Recently, I wrote in some paper that safety properties can be verified by restricting the attenttion to current and next states. A reviewer of the paper found the claim a bit dubious, and would restrict it to safety properties that can be expressed in terms of invariants. Indeed, my gut feeling had been that all safety properties can be expressed in terms of invariants.

In this note, I explain that my gut feeling was wrong: invariants alone are not enough to specify all safety properties. Yet, it becomes true if one allows the introduction of history variables.

2 The definition of safety

Let $M = (X, A, N)$ be a state machine [1]. Here, X is the state space, $A \subseteq X$ is the set of initial states and N is the next state relation, which is supposed to be reflexive. An *execution* of M is an infinite sequence of states $xs \in X^\omega$ with $xs_0 \in A$ and $(xs_n, xs_{n+1}) \in N$ for all $n \in \mathbb{N}$, i.e., an element of

$$\mathbf{Exec}.M = [\![A]\!] \cap \Box [\![N]\!]_2 \ .$$

A subset $D \subseteq X$ is called an *invariant* iff D contains all elements of all executions of M, in other words, iff $\mathbf{Exec}.M \subseteq \Box [\![D]\!]$.

Recall that $L \subseteq X^\omega$ is called a *property* [1] iff $xs \in L$ implies $ys \in L$ for every stutter equivalent pair of sequences xs, ys. Unions and intersections of properties are properties. If U is a subset of X, then $[\![U]\!]$ is a property. If R is a reflexive relation on X, then $\Box [\![R]\!]_2$ is a property. As the next state relation N is required to be reflexive, it follows that the set $\mathbf{Exec}.M$ is a property.

For $xs \in X^\omega$ and $n \in \mathbb{N}$, we write $(xs|n)$ for the finite sequence of the first n elements of xs. For $xs \in X^\omega$ and a subset L of X^ω, we say that xs is a limit point of L iff for every $n \in \mathbb{N}$ there is $ys \in L$ with $(xs|n) = (ys|n)$. The set L is defined to be *closed* iff it contains all its limit points. It is well-known, and easy to prove, that every intersection of closed sets is closed. If U is a subset of X, then $[\![U]\!]$ and $\Box [\![U]\!]$ are closed. If R is a relation on X, then $[\![R]\!]_2$ and $\Box [\![R]\!]_2$ are closed.

The set L is called a *safety property* [1] iff it is a closed property. It follows that $\mathbf{exec}.M$ is a safety property, and that $\Box [\![U]\!]$ is is a safety property for every subset U of X.

Example. Consider a state machine $M = (X, A, N)$ for which the state space contains an integer variable k. For a state $x \in X$, we write $x.k$ for the value of k in state x. Let L be the subset of X^ω of the state sequences in which the values of k never differ by precisely 1. So we have

$$xs \in L \quad \equiv \quad (\forall\, i, j : xs.i.k \neq 1 + xs.j.k) \ .$$

The set L is a property because adding or removing stuttering does not change membership of L. It is also easy to verify that the set L is closed. It is therefore a safety property.

At this point, we know nothing about the state machine M that we can use to contruct an invariant D such that L is expressed by $\Box[\![\, D \,]\!]$. This shows that invariants alone are not enough to specify safety properties. Of course, one can postulate the invariant D that k is odd. Then $\Box[\![\, D \,]\!]$ implies L, but this invariant is clearly overspecific. End of example.

Safety properties are determined by their "treatment" of finite prefixes in the following way. If hs is a nonempty finite list, we write hs^+ for the infinite list obtained by concatenating hs with the infinite repetition of its last element.

Let L be a safety property, and $xs \in L$, and $n \in \mathbb{N}$. The infinite list $ys = (xs|n+1)^+$ is a limit point of L, because we can approximate ys arbitrarily close by letting xs stutter at time n sufficiently often. This stuttering remains in L because L is a property. It follows that $ys \in L$ because L is closed. We define L^- to be the set of the nonempty finite sequences hs such that $hs^+ \in L$. Using that L is closed, we obtain that $L = \{xs \mid \forall\, n : (xs|n+1) \in L^-\}$.

3 History variables

History variables [1] are auxiliary variables that have no role in the algorithm, but that serve in the specification or the proof of the algorithm. They are also called auxiliary variables [2] or ghost variables.

Let $M = (X, A, N)$ be a state machine and let H be the type of the history variable h we want to introduce in M. For this purpose, we choose an initialization function $\varphi_0 : A \to H$ and an update function $\varphi : H \times X \times X \to H$. We then form the extended state machine $M' = (X \times H, A', N')$ where $A' = \{(a, \varphi_0(a)) \mid a \in A\}$ and the extended next state relation N' is given by

$$((x,h),(x',h')) \in N' \quad \equiv \quad (x,x') \in N \ \wedge \ h' = (x \neq x' \ ?\ \varphi(h,x,x') : h) \ .$$

Note that we keep $h' = h$ in the case that $x' = x$, because of the convention of [1] that the next state relation must be reflexive. Machine M' is called the history extension of M.

Let $\pi : X \times H \to X$ be the projection on the first component which forgets h. For every execution xs of M, there is a unique execution ys of M' with $xs.i = \pi(ys.i)$ for all $i \in \mathbb{N}$. On the other hand, every execution ys of M' corresponds to an execution of M in this way. The history variable h records the history of the execution of M, as specified by means of functions φ_0 and φ.

We can now specify an arbitrary safety property L of M in the following way. We take H to be the set X^+ of nonempty finite sequences of states. We take φ_0 and φ to be given by $\varphi_0(a) = \langle a \rangle$, and $\varphi(h,x,x') = (h; \langle x' \rangle)$ where $\langle\ \rangle$ is the singleton constructor and ";" is list concatenation. Let xs be an execution of M. Let ys be the corresponding execution of the history extension M' constructed. At any time n, the value of the history variable h of $ys.n$ equals the stutterfree list obtained by unstuttering the prefix $(xs|n+1)$. The safety property L is therefore expressed by the invariant on machine M' given by $h \in L^-$.

References

1. M. Abadi and L. Lamport. The existence of refinement mappings. *Theor. Comput. Sci.*, 82:253–284, 1991.
2. S. Owicki and D. Gries. An axiomatic proof technique for parallel programs. *Acta Inf.*, 6:319–340, 1976.