A method for avoiding total deadlock, courtesy Diethard Michaelis

In [FvG99], the guarded skip

if $B \rightarrow$ skip fi

is the one and only construct that can cause a component to halt during execution. The special case in which all components are halted in a guarded skip (because all guards B are false) is referred to as a state of total deadlock.

In [FvG99], the canonical way to prove the absence of (the danger of) total deadlock is as follows:

> Select a guarded skip in each component. Let it be
> if $B.i \rightarrow$ skip fi in component i,
> and let $R.i$ be a correct pre-condition of this guarded skip.
> Then, this selection of guarded skips does not give rise to total deadlock whenever
>
> $$\langle \forall i :: R.i \rangle \Rightarrow \langle \exists i :: B.i \rangle \ .$$

* * *

After having studied [FvG99], Diethard Michaelis observed that in many cases absence of total deadlock could be concluded on account of much simpler, more syntactic grounds, viz. as follows:

> Select a guarded skip in each component. Let it be
> if $B.i \to$ skip fi in component i,
> and let $S.i$ be the immediately preceding atomic statement.
> Then, this selection of guarded skips does not give rise to total deadlock whenever
>
> $$\langle \forall i :: \{true\}\ S.i\ \{\langle \exists j :: B.j \rangle\} \rangle .$$

The proof of Michaelis' observation is straightforward. Introduce auxiliary booleans, one per component, to specify that a component is "at" its guarded skip, thus:

Pre: $\langle \forall j :: \neg x.j \rangle$

Comp.i: ... $\langle S.i\ ;\ x.i := true \rangle$
; if $B.i \to x.i := false$ fi
; ...

Then $\langle \forall i :: x.i \rangle \Rightarrow \langle \exists i :: B.i \rangle$

is a system invariant: it holds initially, "x.i := false" falsifies the antecedent, and "⟨ S.i ; x.i := true ⟩" truthifies the consequent thanks to the assumption on S.i . And from this invariant, absence of total deadlock immediately follows.

* * *

In [FvG99], the number of examples where Michaelis' criterion applies is not negligable at all. One convincing example is the Initialization Protocol – cf. [FvG99], pages 85 and 86 – :

A:
```
y := false
; x := true
; if y → skip fi
; x := true
```

B:
```
x := false
; y := true
; if x → skip fi
; y := true
```

No total deadlock ... by Diethard Michaelis!

And there are many others.

* * *

But there is more to it than just proving the absence of total deadlock, since the criterion can be bent into a constructive tool: viz., if a problem allows the programmer to fulfil the criterion's premise, absence of

total deadlock comes for free. And, if in addition, the programmer can manage to create a multibound for the multiprogram, individual progress comes for free as well.

Diethard Michaelis' contribution is precisely of the kind that we are hoping for: it is very simple and, at the same time, extremely effective. Therefore he deserves our well-meant compliments.

W.H.J. Feijen
22 June 2005

[FvG99] On a method of multiprogramming, W.H.J. Feijen and A.J.M. van Gasteren, Springer 1999