ERRATUM

Volume **163**, No. 1 (2000), in the article "Verification by Augmented Finitary Abstraction," by Yonit Kesten and Amir Pnueli, pages 203–243, doi:10.1006/inco.2000.3000): On page 213, line 24, replace the formula $x_{\Box p} \Leftrightarrow \chi(p) \vee x'_{\Box p}$ with the formula $x_{\Box p} \Leftrightarrow \chi(p) \wedge x'_{\Box p}$.

On page 214, line 3, replace $\Theta_\varphi : f_1 \wedge \neg f_3$ with $\Theta_\varphi : u = 0 \wedge f_1 \wedge \neg f_3$.

On page 220, beginning of Section 6.2, replace "In the various..." with "In the previous...."

On page 223, lines 7, replace the formula

$$\sim(\exists V : V_A = \mathcal{E}^\alpha(V) \wedge p(V)) \wedge (\exists V : V_A = \mathcal{E}^\alpha(V) \wedge q(V)), \text{ with the formula}$$
$$\sim\exists V : V_A = \mathcal{E}^\alpha(V) \wedge p(V) \vee \exists V : V_A = \mathcal{E}^\alpha(V) \wedge q(V).$$

On page 224, line 4 replace

$$\alpha^-(p \wedge q) \text{ is equivalent to } \alpha^-(p) \wedge \alpha^-(q) \quad \text{with}$$
$$\alpha^-(p \vee q) \text{ is equivalent to } \alpha^-(p) \vee \alpha^-(q).$$

On page 225, line 32, replace the formula

$$\forall V : V_A = \mathcal{E}^\alpha(V) \rightarrow p(V) \wedge \forall V : V_A = \mathcal{E}^\alpha(V) \rightarrow \forall V : V_A = \mathcal{E}^\alpha(V) \wedge p(V)$$

with the formula

$$\forall V : V_A = \mathcal{E}^\alpha(V) \rightarrow p(V) \wedge \exists V : V_A = \mathcal{E}^\alpha(V) \rightarrow \exists V : V_A = \mathcal{E}^\alpha(V) \wedge p(V).$$

On page 230, third paragraph of Section 7.2, replace "a *ranking monitor* or a ranking function ..." with "a *ranking monitor* for a ranking function...."

On page 241, beginning of Section 10, replace "We have presented a method or verification..." with "We have presented a method for verification...."
Replace Section 8.2 by the Section that follows:

8.2. A Characteristic Example

The whole construction will be illustrated by a single example. Consider the program COND-TERM, presented in Fig. 13.

$$\begin{array}{l}
\quad\quad\quad y\text{: } \textbf{natural} \\
\quad\quad\quad x\text{: } \{-1, 1\} \\
\ell_0 : \ \textbf{while } y > 0 \textbf{ do} \\
\quad\quad\quad \left[\begin{array}{ll} \ell_1 : & x := \pm 1 \\ \ell_2 : & y := y + x \end{array}\right] \\
\ell_3 :
\end{array}$$

FIG. 13. Program COND-TERM.

Statement ℓ_1 of this program nondeterministically assigns to variable x one of the values -1, 1. Program COND-TERM does not always terminate. In particular, it will not terminate if statement ℓ_1 always assigns to x the value 1. Consequently, the best we can claim for this program is the property of conditional termination which can be specified by

$$\psi\colon\ \Diamond\Box\,(x<0) \rightarrow \Diamond\, at\_\ell_3.$$

This property states that if, from a certain point on, x remains negative, then the program will terminate. It is not difficult to see that this property is valid for program COND-TERM.

Since program COND-TERM is a sequential program, it is associated with no fairness requirement. Therefore, step 2 which shifts the fairness requirements from the system to the property is vacuous, and we have that $\mathcal{D}^- = \mathcal{D}$ and $\Psi = \psi$.

Step 3 of the proof scheme constructs a temporal tester $T_{\neg\Psi}$, which characterizes all the sequences violating ψ.

Following the construction described in Section 4, we obtain the BDS $T_{\neg\psi}$, given by

$$V:\quad \pi:\textbf{natural};\quad x:\{-1,1\};\quad f_1, g_2,\ f_3:\textbf{boolean};\quad u:[0..3]$$

$$\Theta_{\neg\psi}:\quad u=0 \wedge f_1 \wedge \neg f_3$$

$$\rho_{\neg\psi}:\ \left(\begin{array}{ccccc} f_1 & \leftrightarrow & g_2 & \vee & f_1' \quad \wedge \\ g_2 & \leftrightarrow & x<0 & \wedge & g_2' \quad \wedge \\ f_3 & \leftrightarrow & at\_\ell_3 & \vee & f_3' \quad \wedge \\ & & u' = & \left[\begin{array}{ll} \textbf{case} & \\ \ \ u=0 & :1; \\ \ \ u=1\wedge(g_2\vee\neg f_1) & :2; \\ \ \ u=2\wedge(x\geq 0\vee g_2) & :3; \\ \ \ u=3\wedge(at\_\ell_3\vee\neg f_3) & :0; \\ \ \ true & :u; \\ \textbf{esac} & \end{array}\right] & \end{array}\right)$$

$$J:\quad u=0$$

Step 4 of the construction forms the parallel composition of $\mathcal{D}=\mathcal{D}^-$ and $T_{\neg\Psi}$ to obtain the combined BDS $\mathcal{B}_{(\mathcal{D},\neg\Psi)} = \mathcal{D} \parallel\!| T_{\neg\Psi}$. We claim that the system $\mathcal{B}_{(\mathcal{D},\neg\Psi)}$ has no computations. Assume to the contrary, that σ is a computation of $\mathcal{B}_{(\mathcal{D},\neg\Psi)}$. To be a computation, σ must contain infinitely many states in which $u=0$. According to the initial condition, f_1 is initially true, while f_3 is initially false. By the transition relation for f_1 and the condition for getting out of $u=1$, there must exist a position $j\geq 0$ such that $g_2=1$ at j. By the transition relation for g_2, it follows that $x<0$ for all positions $k\geq j$. This means that, from j on, all executions of statement ℓ_2 cause y to decrease. Since a natural number cannot decrease infinitely many times, the while loop of the program must terminate, and the execution must reach location ℓ_3, which by $f_3=0$, is impossible.

According to step 5, we should be able to identify an assertion Φ which is an invariant of $\mathcal{B}_{(\mathcal{D}^-,\neg\Psi)}$, and a progress measure Δ. Indeed, for our example, an appropriate invariant assertion is

$$\Phi\colon\ (f_1\vee g_2)\wedge\neg f_3\wedge(u>1\rightarrow g_2)\wedge(\pi\in\{1,2\}\rightarrow y>0),$$

while a progress measure can be given by

$$\Delta:\ \left[\begin{array}{ll} \textbf{case} & \\ \ \ g_2: & (0,\,3y+2at\_\ell_0+at\_\ell_1); \\ \ \ \ 1\ : & (1,0); \\ \textbf{esac} & \end{array}\right].$$

$$
\begin{array}{ll}
B_{y>0}: & \textbf{boolean} \\
x: & \{-1,1\}
\end{array}
$$

$$
\left[\begin{array}{l}
\ell_0: \ \textbf{while } B_{y>0} \textbf{ do} \\
\quad \left[\begin{array}{l}
\ell_1: \ x := \pm 1 \\
\ell_2: \ B_{y>0} := \left[\begin{array}{lr}
\textbf{case} & \\
\quad x > 0 : & 1\,; \\
\quad \neg B_{y>0} : & 0\,; \\
\quad 1 \quad : & \{0,1\}\,; \\
\textbf{esac} &
\end{array}\right]
\end{array}\right] \\
\ell_3:
\end{array}\right]
\;|||\;
\left[\begin{array}{l}
\begin{array}{ll}
F_1, G_2, F_3: & \textbf{boolean} \\
u: & [0..3] \ \textbf{where } u = 0 \\
inc: & \{-1,0,1\} \\
\textbf{justice} & \quad u = 0 \\
\textbf{compassion} & (inc < 0,\ inc > 0)
\end{array} \\
\textbf{always do} \\
\left[\begin{array}{l}
\alpha^{++}(\rho_{\neg\psi}) \ \wedge \\
inc' = \left[\begin{array}{lr}
\textbf{case} & \\
\neg G_2 \ \wedge \ G_2' & : -1\,; \\
\neg G_2 \ \vee \ at\_\ell_3 & : \ 0\,; \\
at\_\ell_{0,1} & : -1\,; \\
B_{y>0} & : -1\,; \\
1 & : \ 1\,; \\
\textbf{esac} &
\end{array}\right]
\end{array}\right]
\end{array}\right]
$$

FIG. 14. Program ABS-COND-TERM, the augmented abstracted version of program COND-TERM.

It is not difficult to see that any transition taken from a Φ-state is guaranteed not to increase Δ. If such a transition leads to a state in which $u = 0$ then Δ must decrease.

In step 6, we use the tester $T_{true}^{\neg\Psi}$ and the progress measure Δ to construct the progress monitor $M_{T,\Delta}$ given by

$$
M_{T,\Delta}: \left\langle \begin{array}{l}
V_M : \{\pi : \textbf{natural},\ x : \{-1, 1\},\ f_1,\ g_2,\ f_3 : \textbf{boolean},\ u : [0..3],\ inc : \{-1, 0, 1\}\} \\
\Theta_M : u = 0 \quad \rho_M : \rho_{\neg\psi} \wedge inc' = diff(\Delta, \Delta') \\
\mathcal{J} : u = 0 \quad \mathcal{C} : \{(inc < 0, inc > 0)\}
\end{array} \right\rangle .
$$

Next, we form the composition $\mathcal{D} \,|||\, M_{T,\Delta}$, and then compute the abstraction mapping α. To obtain a finitary mapping, we introduce a fresh Boolean variable $B_{y>0}$ with the definition $B_{y>0} = (y > 0)$. Applying the abstraction α to $\mathcal{D} \,|||\, M_{T,\Delta}$, we obtain an abstracted finite-state system equivalent to the program presented in Fig. 14.

The variables F_1, G_2, F_3 are the abstract versions of f_1, g_2, and f_3, respectively. Note that, like $\mathcal{D} \,|||\, M_{T,\Delta}$, system ABS-COND-TERM is a parallel composition of three components, the abstraction of program COND-TERM, the abstraction of the tester T_{true}^{Ψ}, and the abstraction of the monitor, taking into account its joint behavior with the other two components.

Clearly, the system ABS-COND-TERM is a finite-state system and satisfies the property

$$
\psi\colon \Diamond\Box\,(x < 0) \rightarrow \Diamond\, at\_\ell_3 .
$$

To see that ABS-COND-TERM satisfies the property ψ, assume, to the contrary, that there exists a computation σ of ABS-COND-TERM which satisfies $\Diamond\Box\,(x < 0)$ but never reaches location ℓ_3. In this case, the initial values of f_1 and f_3 must be 1 and 0, respectively. The justice requirement with respect to u cannot be satisfied in such a case, unless g_2 eventually assume the value 1. Once this happens, inc is constantly -1 from this point on. This violates the compassion requirement with respect to inc. It follows that σ cannot be a computation.