CORRIGENDUM

Volume 92 Number 2 (1991), in the article "A Domain Equation for Bisimulation," by Samson Abramsky, pages 161–218

Abramsky's seminal article [3] is devoted to a detailed concurrency-theoretic application of the author's "theory of domains in logical form" programme [4]. One of the main results in [3] is a logical characterization of the finitary bisimulation (cf. Theorem 5.8 on p. 191). (See [5] for a behavioural characterization of the finitary bisimulation.) More precisely, Abramsky shows that two processes in any transition system are related by the finitary bisimulation iff they are related by the preorder on processes induced by the finitary version of the domain logic for transition systems synthesized in [3]. This result is really the acid test for the goodness of the domain logic for transition systems, in that it shows that, unlike Hennessy–Milner logic [6, 7], its finitary version captures exactly the finitely observable distinctions made by the notion of bisimulation.

Unfortunately, there is a subtle error in the proof of Theorem 5.8 on p. 191. The interpretation of the domain logic in [3] yields a logic which is sufficiently powerful to distinguish processes that are related by the finitary bisimulation. This is in contrast to the interpretation of the domain logic for transition systems offered by Abramsky in his doctoral dissertation [1]. The original interpretation of the domain logic is subtly different from the one published in the journal article [3], and is the correct one. In particular, the logical characterization theorem for the finitary bisimulation offered in [1, Theorem 5.5.8] does hold. The researchers who wish to apply Abramsky's domain logic for transition systems in their work should therefore use the version presented in [1, Chap. 5] in lieu of that offered in the journal paper.

In the remainder of this communication, we present a simple counterexample to Theorem 5.8 on p. 191 of [3] and show that the version of the finitary domain logic presented in [3] is strictly more expressive than the original version of the logic from [1]. Familiarity with [3] is assumed below, although some basic definitions are repeated briefly for the sake of clarity.

Let $(\mathsf{Proc}, \mathsf{Act}, \rightarrow, \uparrow)$ be a transition system in the sense of [3, Definition 2.1]. The *finitary bisimulation*, denoted by $\sqsubseteq^F$, is defined thus: for every $p, q \in \mathsf{Proc}$,

$$p \mathrel{\sqsubseteq^F} q \text{ iff for every finite synchronization tree } t,\ t \mathrel{\sqsubseteq^B} p \text{ implies } t \mathrel{\sqsubseteq^B} q,$$

where $\sqsubseteq^B$ is the partial bisimulation preorder defined in [3, Definition 2.2]. Intuitively, $p \sqsubseteq^F q$ holds for two processes p and q iff the observations, in the sense of [2], of finite-depth and width that are afforded by p are included in those afforded by q. An instructive example of the role played by finite width in observations, and of the weakness of $\sqsubseteq^F$ over infinite branching processes, is the following:

EXAMPLE 0.1. Assume that $\mathsf{Act} = \{a_i \mid i \geq 0\}$ and that $i \neq j$ implies $a_i \neq a_j$. Consider the synchronization trees $\mathsf{All}_\downarrow$ and $\mathsf{All}_\uparrow$ given by

$$\mathsf{All}_\downarrow \triangleq \sum_{i \geq 0} a_i : \mathbb{O}$$

$$\mathsf{All}_\uparrow \triangleq \mathsf{All}_\downarrow + \Omega.$$

Then $\mathsf{All}_\downarrow \sqsubseteq^F \mathsf{All}_\uparrow$ holds, even though $\mathsf{All}_\downarrow$ is convergent and $\mathsf{All}_\uparrow$ is not. In fact, let t be a finite synchronization tree such that $t \sqsubseteq^B \mathsf{All}_\downarrow$. We shall now argue that $t \sqsubseteq^B \mathsf{All}_\uparrow$ must also hold. First, note that $t \sqsubseteq^B \mathsf{All}_\downarrow$ implies that $t \uparrow$. (This is easy to see because otherwise the finite synchronization tree t would have to have the infinite set Act as its set of initial actions.) Next we remark that if $t \xrightarrow{a} t'$ for some actin a, then $t' \sqsubseteq^B \mathbb{O}$. From these two observations, it follows immediately that $t \sqsubseteq^B \mathsf{All}_\uparrow$.

0890-5401/01 $35.00

The moral of the above example is that observations based on finite synchronization trees cannot, in general, be used to test the convergence of a process that, like $\mathsf{All}_{\downarrow}$ above, can perform infinitely many distinct initial actions.

The processes described by the synchronization trees $\mathsf{All}_{\downarrow}$ and $\mathsf{All}_{\uparrow}$ are at the heart of our counter-example disproving Theorem 5.8 in [3]. Before discussing the example, we review, for the sake of clarity, Abramsky's domain logic for transition systems, as originally presented in [1].

Abramsky's (finitary) domain logic for transition systems $\mathcal{L}_{\omega}$ (over a set of actions Act) is a two-sorted language with sorts π (process) and κ (capability). We write $\mathcal{L}_{\omega\pi}$ (respectively, $\mathcal{L}_{\omega\kappa}$) for the class of formulae of sort π (respectively, κ) which are defined inductively as follows,

$$\frac{\{\phi_i \in \mathcal{L}_{\omega\sigma}\}_{i\in I}}{\bigwedge_{i\in I}\phi_i,\ \bigvee_{i\in I}\phi_i \in \mathcal{L}_{\omega\sigma}} \qquad \frac{a \in \mathsf{Act},\ \phi \in \mathcal{L}_{\omega\pi}}{a(\phi) \in \mathcal{L}_{\omega\kappa}} \qquad \frac{\phi \in \mathcal{L}_{\omega\kappa}}{\phi, \phi \in \mathcal{L}_{\omega\pi}}$$

where I is a finite index set, and $\sigma \in \{\pi, \kappa\}$. As usual, we write true for $\bigwedge_{i\in\emptyset}\phi_i$.

Given a transition system $(\mathsf{Proc}, \mathsf{Act}, \rightarrow, \uparrow)$, we define

$$\mathrm{Cap} \triangleq \{\bot\} \cup (\mathsf{Act} \times \mathsf{Proc}) \quad \text{(the set of } \textit{capabilities}\text{)}$$

$$C(p) \triangleq \{\bot \mid p \uparrow\} \cup \{\langle a, q\rangle \mid p \xrightarrow{a} q\} \quad \text{(the set of capabilities of process } p\text{)}.$$

The satisfaction relations

$$\models_{\pi} \subseteq \mathsf{Proc} \times \mathcal{L}_{\omega\pi}$$

and

$$\models_{\kappa} \subseteq \mathrm{Cap} \times \mathcal{L}_{\omega\kappa}$$

are now defined thus ($\sigma \in \{\pi, \kappa\}$, $w \in \mathsf{Proc} \cup \mathrm{Cap}$):

$$\begin{aligned}
w \models_{\sigma} \bigwedge_{i\in I}\phi_i &\triangleq \forall i \in I.\ w \models_{\sigma} \phi_i \\
w \models_{\sigma} \bigvee_{i\in I}\phi_i &\triangleq \exists i \in I\colon w \models_{\sigma} \phi_i \\
p \models_{\pi} \phi &\triangleq \forall c \in C(p).\ c \models_{\kappa} \phi \\
p \models_{\pi} \phi &\triangleq \exists c \in C(p) \cup \{\bot\}.\ c \models_{\kappa} \phi \\
c \models_{\kappa} a(\phi) &\triangleq \exists q\colon c = \langle a, q\rangle \text{ and } q \models_{\pi} \phi.
\end{aligned}$$

In his doctoral dissertation [1], Abramsky shows that the finitary bisimulation coincides with the preorder over processes induced by the finitary domain logic for transition systems presented (cf. [1, Theorem 5.5.8]), i.e., that $p \sqsubseteq^{F}_{\simeq} q$ holds exactly when the formulae satisfied by p are also satisfied by q. In light of the example we previously discussed, this means, in particular, that no formula of the finitary domain logic from [1] is satisfied only by convergent processes—such a formula would be satisfied by $\mathsf{All}_{\downarrow}$, but not by $\mathsf{All}_{\uparrow}$.

In contrast with the above considerations, a formula satisfied only by convergent processes can be constructed in the version of the finitary domain logic for transition systems presented by Abramsky in [3]. The satisfaction relation for the $\Box$ and $\Diamond$ modalities is given thus in [3]:

$$\begin{aligned}
p \Vdash_{\pi} \phi &\triangleq p \downarrow \text{ and } \forall c \in C(p).\ c \Vdash_{\kappa} \phi \\
p \Vdash_{\pi} \phi &\triangleq \exists c \in C(p) : c \Vdash_{\kappa} \phi
\end{aligned}$$

With this interpretation, the formula true is satisfied exactly by all the convergent processes. In light of the previously discussed example, the presence of such a formula invalidates Theorem 5.8 on p. 191

of [3], which states the logical characterization theorem for the finitary bisimulation in terms of the version of the finitary domain logic offered in [3].

The error in the proof of Theorem 5.8 in [3] can be traced to the falsity of [3, Theorem 4.5]. This result states that every process formula in the finitary domain logic can be effectively transformed into an equivalent one in strong disjunctive normal form (cf. Definition 4.4 on p. 182 of [3]). This result is incorrect because, as the reader can easily verify, the formula true has no equivalent strong disjunctive normal form.

The interpretation of the domain logic presented in [3] yields a logic that is strictly more expressive that the original one from [1]. To see that this is indeed the case, we shall define a semantics preserving translation from the domain logic in [1] to that in [3]. In order to define this translation, it will be convenient to recall the predicate $\uparrow$ on formulae from [1, p. 102]. This is the smallest predicate satisfying

$$\begin{aligned}
\left(\textstyle\bigwedge_{i\in I}\phi_i\right)\uparrow &\triangleq \forall i \in I : \phi_i\uparrow\\
\left(\textstyle\bigvee_{i\in I}\phi_i\right)\uparrow &\triangleq \exists i \in I.\ \phi_i\uparrow\\
(\phi)\uparrow &\triangleq \phi\uparrow\\
(\phi)\uparrow &\triangleq \phi\uparrow.
\end{aligned}$$

Intuitively, $\phi\uparrow$ means that ϕ is logically equivalent to true in the interpretation of the finitary domain logic given in [1]. We write $\phi\downarrow$ iff it is not the case that $\phi\uparrow$.

We now define the translation ϕ^* of a formula ϕ thus:

$$\begin{aligned}
\left(\textstyle\bigwedge_{i\in I}\phi_i\right)^* &\triangleq \bigwedge_{i\in I}\phi_i^*\\
\left(\textstyle\bigvee_{i\in I}\phi_i\right)^* &\triangleq \bigvee_{i\in I}\phi_i^*\\
(\phi)^* &\triangleq \begin{cases}\mathsf{true} & \text{if } \phi\uparrow\\ (\phi^*) & \text{otherwise}\end{cases}\\
(\phi)^* &\triangleq \begin{cases}\mathsf{true} & \text{if } \phi\uparrow\\ (\phi^*) & \text{otherwise}\end{cases}\\
(a(\phi))^* &\triangleq a(\phi^*).
\end{aligned}$$

Then:

PROPOSITION 0.1. *In any transition system,*

$$\begin{aligned}
&(1)\quad p \models_\pi \phi \Leftrightarrow p \Vdash_\pi \phi^*\\
&(2)\quad c \models_\kappa \phi \Leftrightarrow c \Vdash_\kappa \phi^*.
\end{aligned}$$

Proof. The two statements are proven simultaneously by induction on the structure of ϕ. The proof is routine, using Proposition 5.4.2 on p. 103 of [1]. Here we only present a sketch of the proof of the "only if" implication for statement (1) when ϕ takes the form ψ for some $\psi\downarrow$. To this end, assume that $p \models_\pi \psi$. This means that $c \models_\kappa \psi$ for every $c \in C(p)$. As $\psi\downarrow$, Proposition 5.4.2(i) of [1] yields that $\bot$ is not a capability of p, i.e., that p converges. By the inductive hypothesis, we now infer that $c \Vdash_\kappa \psi^*$ for every $c \in C(p)$. It follows that $p \Vdash_\pi \psi^*$, which was to be shown. ■

In view of the above result, every formula in the finitary domain logic from [1] is equivalent to one in the finitary logic from [3]. On the other hand, the formula true in the logic from [3] cannot be expressed in the logic from [1, Chap. 5].

ACKNOWLEDGMENT

We have benefited from illuminating discussions with Samson Abramsky on the two different versions of the domain logic for transition systems presented in [1, 3].

REFERENCES

1. S. Abramsky, "Domain Theory and the Logic of Observable Properties," Ph.D. thesis, University of London, 1987.
2. S. Abramsky, Observation equivalence as a testing equivalence, *Theoret. Comput. Sci.* **53** (1987), 225–241.
3. S. Abramsky, A domain equation for bisimulation, *Inform. and Comput.* **92** (1991), 161–218.
4. S. Abramsky, Domain theory in logical form, *Ann. Pure Appl. Logic* **51** (1991), 1–77.
5. L. Aceto and A. Ingólfsdóttir, A characterization of finitary bisimulation, *Inform. Process. Lett.* **64**(3) (1997), 127–134.
6. M. Hennessy and R. Milner, Algebraic laws for nondeterminism and concurrency, *J. Assoc. Comput. Mach.* **32** (1985), 137–161.
7. R. Milner, A modal characterisation of observable machine behaviour, *in* "Proceedings CAAP 81" (G. Astesiano and C. Bohm, Eds.), Lecture Notes in Computer Science, Vol. 112, pp. 25–34, Springer-Verlag, Berlin/New York, 1981.

Luca Aceto and Anna Ingólfsdóttir

Basic Research in Computer Science
Centre of the Danish National Research Foundation
Department of Computer Science
Aalborg University, Fredrik Bajersvej 7E
9220 Aalborg 0, Denmark
E-mail: luca@cs.quc.dk, annai@cs.auc.dk