Inheritance Preserves Successful Termination Indeed!

Twan Basten[1,2], Jos C.M. Baeten[1,3]

[1]Eindhoven University of Technology
[2]Embedded Systems Institute
[3]CWI

11-11-2011

Abstract

The original development of life-cycle inheritance in a process-algebraic setting [2] does not distinguish between successful and unsuccessful termination. In this note, we reconsider life-cycle inheritance in the algebraic framework of [1]. We show that life-cycle inheritance preserves successful termination options. Given that today witnesses the successful termination of the professional career of Kees van Hee, this result is good news for his PhD graduates.

1 Life-Cycle Inheritance

Let A be a set of observable actions; internal, or silent, actions are denoted τ. We reconsider the notion of life-cycle inheritance from [2] in the equational theory $(\mathrm{TCP}_\tau + \mathrm{FMA})(A, \emptyset)$, the Theory of Communicating Processes with silent steps and the Free-Merge Axiom, as defined in [1]. The essential difference between the framework of [2] and $(\mathrm{TCP}_\tau + \mathrm{FMA})(A, \emptyset)$ is the presence of the empty-process constant, denoted 1, which allows to distinguish successful termination in a process from deadlock, or unsuccessful termination.

Definition 1.1 (Life-cycle inheritance) For closed $(\mathrm{TCP}_\tau + \mathrm{FMA})(A, \emptyset)$-terms p and q, p is a child of q under *life-cycle inheritance*, denoted $p \leq_{lc} q$, if and only if there exist *disjoint* H, $I \subseteq A$ such that $(\mathrm{TCP}_\tau + \mathrm{FMA})(A, \emptyset) \vdash \tau_I \circ \partial_H(p) = q$.

Example 1.2 (Life-cycle inheritance) Consider the parent process $a.1$. Process $a.b.1$ is a child under life-cycle inheritance: $(\mathrm{TCP}_\tau + \mathrm{FMA})(A, \emptyset) \vdash \tau_{\{b\}}(a.b.1) = a.\tau.1 = a.1$. Process $b.0 + a.1$ is also a child under life-cycle inheritance: $(\mathrm{TCP}_\tau + \mathrm{FMA})(A, \emptyset) \vdash \partial_{\{b\}}(b.0 + a.1) = 0 + a.1 = a.1$.

Closed $(\mathrm{TCP}_\tau + \mathrm{FMA})(A, \emptyset)$-terms may contain operators such as sequential composition $(\_ \cdot \_)$ and parallel composition $(\_ \parallel \_)$. They can however be reduced to derivably equal $\mathrm{BSP}_\tau(A)$-terms (with $\mathrm{BSP}_\tau(A)$ the theory of Basic Sequential Processes with silent actions). In other words, closed $(\mathrm{TCP}_\tau + \mathrm{FMA})(A, \emptyset)$-terms can be reduced to terms containing only action-prefix operators ($\alpha.\_$, for $\alpha \in A \cup \{\tau\}$), alternative-composition operators $(\_ + \_)$, and inaction (0) and empty-process (1) constants.

Proposition 1.3 (Elimination, [1]**)** For any closed $(\mathrm{TCP}_\tau + \mathrm{FMA})(A, \emptyset)$-term p, there is a closed $\mathrm{BSP}_\tau(A)$-term q such that $(\mathrm{TCP}_\tau + \mathrm{FMA})(A, \emptyset) \vdash p = q$.

2 Successful Termination Options

The following definition is taken from [1] (Definition 4.4.14, adapted to $(\mathrm{TCP}_\tau + \mathrm{FMA})(A, \emptyset)$).

Definition 2.1 (Deadlock) A closed $(\mathrm{TCP}_\tau + \mathrm{FMA})(A, \emptyset)$-term p is *deadlock free* if and only if there is a closed $\mathrm{BSP}_\tau(A)$-term q without the occurrence of the inaction constant 0 such that $(\mathrm{TCP}_\tau + \mathrm{FMA})(A, \emptyset) \vdash p = q$. Term p *has a deadlock* if and only if it is not deadlock free.

Example 2.2 (Deadlock) The process $b.0 + a.1$ has a deadlock. Process $0 + a.1$ does not have a deadlock though, because $(\mathrm{TCP}_\tau + \mathrm{FMA})(A, \emptyset) \vdash 0 + a.1 = a.1$.

This definition captures processes that may have deadlock and those, the deadlock free ones, that are guaranteed to complete successfully under all circumstances. The processes that may deadlock can be divided into those processes that have a successful termination option, to be defined next, and those that do not, that are bound to deadlock.

Definition 2.3 (Successful termination option) A closed $(\mathrm{TCP}_\tau + \mathrm{FMA})(A, \emptyset)$-term p *has a successful termination option* if and only if there is a closed $\mathrm{BSP}_\tau(A)$-term q containing an occurrence of the empty-process constant 1 such that $(\mathrm{TCP}_\tau + \mathrm{FMA})(A, \emptyset) \vdash p = q$.

Example 2.4 (Successful termination) Processes $a.1$ and $b.0 + a.1$ both have successful termination options. Processes $a.0 \cdot 1$ and $0 \parallel 1$ do not, despite the occurrences of the empty-process constants. $a.0 \cdot 1$ is derivably equal to $a.0$ and $0 \parallel 1$ is derivably equal to 0. (These examples illustrate that in Definition 2.3 it is necessary that term q is restricted to a closed $\mathrm{BSP}_\tau(A)$-term.)

3 Preservation of Successful Termination Options

We show that life-cycle inheritance preserves successful termination options. That is, if a term q has a successful termination option, then any child $p \leq_{lc} q$ has a successful termination option.

Theorem 3.1 (Preservation of successful termination options) Let p, q be closed $(\mathrm{TCP}_\tau + \mathrm{FMA})(A, \emptyset)$-terms with $p \leq_{lc} q$. If q has a successful termination option, then p has a successful termination option.

Proof Based on Proposition 1.3 (Elimination), it is sufficient to consider closed $\mathrm{BSP}_\tau(A)$-terms p and q. Let $H, I \subseteq A$ be such that $(\mathrm{TCP}_\tau + \mathrm{FMA})(A, \emptyset) \vdash \tau_I \circ \partial_H(p) = q$ and assume that q has a termination option (i.e., it contains an occurrence of the empty-process constant 1).

Consider equational theory $(\mathrm{BSP}_\tau + \mathrm{ABS} + \mathrm{DH})(A)$, Basic Sequential Processes with abstraction and encapsulation (see [1]). $(\mathrm{TCP}_\tau + \mathrm{FMA})(A, \emptyset)$ is a conservative ground-extension of $(\mathrm{BSP}_\tau + \mathrm{ABS} + \mathrm{DH})(A)$. Hence, for reasoning about equality between closed terms in the smaller theory $(\mathrm{BSP}_\tau + \mathrm{ABS} + \mathrm{DH})(A)$, it is sufficient to consider derivations within this theory only.

Since none of the axioms of $(\mathrm{BSP}_\tau + \mathrm{ABS} + \mathrm{DH})(A)$ introduces or eliminates empty-process constants, it follows from the assumptions that $\tau_I \circ \partial_H(p)$ has a successful termination option. Again, since none of the abstraction and encapsulation axioms introduces or eliminates empty-process constants, also p has a successful termination option.

Unfortunately, it cannot be shown that any child p of q under life-cycle inheritance has a deadlock only if q has a deadlock. A child process may choose wrongly and end up in a deadlock even though the parent process does not have a deadlock.

Example 3.2 (Child with a deadlock) Consider the parent process $a.1$. It terminates successfully after performing action $a \in A$ and does not have a deadlock. Its child under life-cycle inheritance $b.0 + a.1$ has a deadlock though.

4 Conclusion

A child of a parent that has successful termination options also has successful termination options. Since Kees van Hee has terminated his career successfully, as a consequence, his PhD graduates have successful career options as well. It cannot be excluded though that a child inheriting from a successful parent also has unsuccessful options, in addition to the successful termination options inherited from its parent. Future work will therefore need to focus on guiding children of successful parents to avoid such unsuccessful options.

References

[1] J.C.M. Baeten, T. Basten, and M.A. Reniers. *Process Algebra: Equational Theories of Communicating Processes*, volume 50 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 2010.

[2] T. Basten. *In Terms of Nets: System Design with Petri Nets and Process Algebra*. PhD thesis, Eindhoven University of Technology, Eindhoven, Netherlands, 1998.