氏名（本籍地）	アドリアン ウベール マリー ピエラール Adrien Hubert Marie Pierard
学位の種類	博　士（情報科学）
学位記番号	情　博　第528号
学位授与年月日	平成24年3月27日
学位授与の要件	学位規則第4条第1項該当
研究科、専攻	東北大学大学院情報科学研究科（博士課程）情報基礎科学専攻
学位論文題目	Environmental Bisimulations for Higher-Order Distributed Process Calculi（高階分散プロセス計算のための環境双模倣）
論文審査委員	（主査）東北大学教授　小林　直樹 東北大学教授　大堀　淳　東北大学教授　外山　芳人 東北大学准教授　住井英二郎

論文内容の要旨

Chapter 1: Introduction

With the increasing call for fault-tolerance, on-demand computational power, and better responsiveness, higher-order (the ability to send and receive processes through communication channels) and distribution (the possibility of location-dependent behaviour) are pervasive in today's computing environment. For example, Dell and Hewlett Packard sell products with virtual machine live migration, and Gmail relies on remote execution of JavaScript in the users' browsers. Yet, despite the ubiquity and importance of such higher-order distributed systems, the inherent complexity of these systems makes them difficult to analyse, and thus subject to bugs. Therefore, formal models and methods that help reason about higher-order distribution are sought after.

One way to show correctness of systems is to model them and their specification or reference implementation into *processes* of a *process calculus*, and then to prove equivalence of these processes. We define equivalence as *reduction-closed barbed equivalence* (or *reduction-closed barbed congruence*), which has a simple definition but requires universal quantification over arbitrary processes (or general contexts). Therefore, other relations that imply equivalence but come with a co-inductive proof method, like bisimulations, are desired.

In this thesis, we focus on *environmental bisimulations* [Sumii and Pierce] for two calculi, HOpiP—the Higher-Order Pi-Calculus with Passivation [Lenglet et al.]—and HOpiPc—the Higher-Order Pi-Calculus with Passivation and Name Creation. HOpiP and HOpiPc are foundational process calculi for modelling higher-order distributed systems. As higher-order calculi they provide the following input and output constructs:

a!M.P -a!M-> P (Output),

a?X.P -a?M-> P{M/X} (Input),

where P -α-> Q reads in general “P does transition α and becomes Q”, and distribution is expressed by means of *locations* and of a *passivation* construct:

l[P] -α-> l[P’] if P -α-> P’ (Transparency)

l[P] -l!P-> 0 (Passivation)

so that one can model, for example:

failure: a[P] | a(X).fail -τ-> 0 | fail,

migration: a[P] | a(X).b[X] -τ-> 0 | b[P],

duplication: a[P] | a(X).(b_1[X]|b_2[X]) -τ-> 0 | b_1[P] | b_2[P],

where τ represents a silent action, l[P] the process P at location l, and P | Q the parallel composition of P and Q.

The difference between these calculi lies in their treatment of names, with HOpiP using common *name restriction* semantics, i.e., the hiding of names, and HOpiPc using *name creation*, i.e., the generation of a fresh name.

Our environmental bisimulations are roughly defined as sets of triplets (E, P, Q) where E is a binary relation on processes, called the environment, and P and Q are processes compared under environment E. Intuitively, the environment represents the knowledge of the observer who compares P and Q. The bisimulations are constrained by several clauses so that related P's and Q's are expectedly equivalent. Roughly, when (E, P, Q) is in a bisimulation X, then

1. If P -т-> P', then Q -т-> Q' for some Q' such that (E, P', Q') is in X, that is, if P silently becomes P', then Q silently becomes a Q' such that continuations are still related.
2. If P -a!M-> P', then Q -a!N-> Q' for some N, Q', and (E U {(M,N)}, P', Q') is in X, that is, if P outputs some M, then Q must be able to output some N such that the continuations P' and Q' are related. Notice that the observer sees M and N, and therefore we compare P' and Q' under the extended knowledge E U {(M, N)}.
3. The knowledge E can be fed to P and Q as follows: if P -a?M-> P' then for any (M, N) built from E, Q -a?N-> Q', and (E, P', Q') is in X. In other words, the observer can use his knowledge to force P and Q to input processes made from it, hoping to tell them apart, should Q not be able to input or (E, P', Q') not be in the bisimulation.
4. The knowledge can also be used as follows: if (M, N) is in E, then (E, P | a[M], Q | a[N]) is in X. This represents the idea that the observer can always spawn the processes of his knowledge in parallel with the tested processes. The presence of locations "a[ ]" hosting M and N is necessary for our distributed calculus, as the observer can remove, duplicate, or migrate M and N at any moment, including in the middle of their execution. In fact, clause 4, is critical to the soundness of our bisimulations, and we discuss its motivation.
5. Finally, the converse of 1, 2 and 3 on Q's transitions is required too.

Then, we give an example of a non-trivial equivalence (which holds for both HOpiP and HOpiPc) by crafting and explaining an environmental bisimulation.

Finally, we list the main contributions of our thesis and give its outline.

Chapter 2: The Higher-Order Pi-Calculus with Passivation and Name Creation

Because name creation semantics is closer to implementations than name restriction semantics when modelling higher-order distributed systems, we introduce a new calculus: the Higher-Order Pi-Calculus with Passivation and Name Creation (HOpiPc). The syntax of HOpiPc is similar to that of the Higher-Order pi-Calculus with Passivation of [Lenglet et al.] on which it is based, but we show that, maybe unlike common expectations, name creation semantics in HOpiPc differs from name restriction semantics. We detail this difference, using simple examples of processes whose behaviours change depending on what semantics is chosen for the name binding operator.

Then, after defining our equivalences for HOpiPc, we define its environmental bisimulations and environmental bisimulations *up-to context*, as well as their asymmetric versions, namely environmental simulations and environmental simulations up-to context. Bisimulations up-to context have weaker requirements than bisimulations, therefore improving the practicality of our proof method. Soundness of environmental bisimulations (resp. simulations) up-to context is proven, and used in turn to prove soundness of environmental bisimulations (resp. simulations) with respect to reduction-closed barbed equivalence. Also, we extend our soundness results to that of bisimulations with respect to a (reasonable) form of reduction-closed barbed congruence (resp. pre-congruence) whose definition we justify. Finally, we prove the completeness of our environmental bisimulations with respect to reduction-closed barbed equivalence and congruence.

Next, to apply our theory, we prove the non-trivial equivalence of distributed left-fold and right-fold functions under arbitrary duplications of locations (and of their hosted processes) by the observer.

Inequivalence in HOpiPc is also discussed in details, by giving examples of processes that are told apart because the use of passivation enables distinguishing the number of created names, or their creation order. Nonetheless, we finally show that *simulation equivalence* (that is, mutual simulation) can still be used to relate processes that are not bisimilar, for it is not sensitive to deadlocks. We then give a non-trivial simulation equivalence proof that relates linear and logarithmic versions of the algorithm that computes f(a,b)=a^b.

Chapter 3: The Higher-Order Pi-Calculus with Passivation and Name Restriction

When [Lenglet et al.] defined HOpiP, the Higher-Order Pi-Calculus with Passivation (which uses name restriction), they provided for it sound and complete *context bisimulations,* which are hard by definition to handle in practice, and *normal bisimulations* that are sound only in the *absence* of name restriction. In order to overcome the limitations of context bisimulations and of [Lenglet et al.]'s normal bisimulations, we therefore study environmental bisimulations (standard, and up-to context) for HOpiP.

As in chapter 2, we define our equivalence and our environmental bisimulations for HOpiP. We then discuss important technical details regarding the fourth clause of our bisimulations for defining a sound (alas, incomplete) proof method for calculi with passivation and name restriction; we also discuss differences with variations on that clause used in previous research. We then show that our environmental bisimulation up-to context proof technique is sound under some constraints on the environments, and qualify as *simple* the bisimulations that verify these constraints. Simple environmental bisimulations up-to context are then used to show soundness of simple environmental bisimulations.

In this chapter, many non-trivial results are detailed and explained, notably with respect to *run-erasure*, a technique we use to get round subtleties arising during the soundness proof of bisimulations up-to context.

Finally, we conclude by giving several examples of non-trivial equivalences in HOpiP.

Chapter 4: Application: Modelling and Verifying GXP

In order to emphasise the usability of our proof method, we apply it to a realistic example, GXP [Taura]. GXP is a tool that enables the transparent execution of user-provided arbitrary commands on each node of a network, without requiring neither prior nor manual installation of software on those nodes. Basically, GXP works by replicating itself onto all machines it can transitively connect to, and setting up servers that wait for commands to execute and to forward to other servers. Because GXP self-replicates itself and transfers commands, and because nodes of the network are subject to failure at any moment, GXP corresponds to our definition of a higher-order distributed system.

We model the implementation of GXP by crafting a HOpiPc process that makes explicit use of dynamic discovery of hosts and self-replication, and model its specification by another process where servers and connections are statically set up. In both processes, the possibility of failure is represented by the possibility of passivation at any moment of locations that represent nodes of the network.

We then show that our models of the specification and implementation of GXP for a simple, yet non-trivial network topology, are equivalent by providing an environmental bisimulation up-to context relating them.

Finally, we discuss simplifications in our models with respect to the real GXP, and how the transparency of locations also impacted our modelling.

Chapter 5: Related Work

We discuss several related work, starting with the original research on HOpiP [Lenglet et al.] that provided bisimulations that either are unsound or suffer from heavy use of universal quantification on general contexts. We then discuss two more expressive calculi with name restriction and non-transparent locations: the Kell calculus [Schmitt and Stefani], and Homer [Hildebrandt et al.]. Non-transparency of locations allows observers to distinguish messages based on their provenance: from the same location, a location above, or one below. For both calculi, only context bisimulations were defined, i.e., proof methods of limited practicality compared to reduction-closed barbed equivalence.

Then, we discuss two first-order, less general, distributed calculi. The first, Dpi [Hennessy and Riely], provides a migration construct and non-nested locations, and focuses on modelling *crash-failure.* The second, the Ambient calculus [Cardelli and Gordon], provides migration constructs and nested locations, and focuses on modelling *mobility*. Both calculi identify name creation and name restriction semantics, and their bisimulations are akin to context bisimulations.

Finally, we discuss the Seal calculus [Castagna et al.], a model of mobility with nested locations, migration, and duplication. Unlike HOpiP, the Seal calculus cannot model reactions that involve arbitrary nesting of locations in a single step, and runs processes immediately after their transfer. For this calculus too, equivalence is proven with context bisimulations.

The last part of related work discusses environment-sensitive bisimulations. Such bisimulations were initially defined for first-order calculi and therefore do not require a clause to spawn processes from the environment. For higher-order calculi like variations on the lambda calculus or on the higher-order pi-calculus, environment-sensitive bisimulations have been successfully adapted. Notable differences with our bisimulations are the absence of clause 4, or its simplicity because of the absence of passivation.

Chapter 6: Conclusion

We conclude our thesis by discussing future work and applicability of our proof method. We first reconsider the original HOpiP and explain why we chose to modify it slightly before defining our environmental bisimulations for it. Then, we discuss HOpiP and HOpiPc together, and how we could improve their practicality by extending them with non-transparent locations.

Finally, we discuss the practicality of our proof method compared to direct use of the definition of equivalence, depending on whether our bisimulations have empty environments or not.

Appendix A: Proofs for Chapter 2.
Appendix B: Proofs for Chapter 3.

Bibliography

[Cardelli and Gordon] Luca Cardelli and Andrew D. Gordon. Mobile ambients. In Foundations of Software Science and Computation Structures, volume 1378 of Lecture Notes in Computer Science, pages 140–155. Springer, 1998.

[Castagna et al.] Giuseppe Castagna, Jan Vitek, and Francesco Zappa Nardelli. The Seal Calculus. Information and Computation, 201(1):1–54, 2005.

[Hennessy and Riely] Matthew Hennessy and James Riely. Resource access control in systems of mobile agents. Information and Computation, 173:82–120, 1998.

[Hildebrandt et al.] Thomas Hildebrandt, Jens Chr. Godskesen, and Mikkel Bundgaard. Bisimulation congruences for Homer: a calculus of higher-order mobile embedded resources. Technical Report TR-2004-52, IT University of Copenhagen, 2004.

[Lenglet et al.] Sergueï Lenglet, Alan Schmitt, and Jean-Bernard Stefani. Normal bisimulations in calculi with passivation. In Foundations of Software Science and Computational Structures, volume 5504 of Lecture Notes in Computer Science, pages 257–271. Springer, 2009.

[Schmitt and Stefani] Alan Schmitt and Jean-Bernard Stefani. The Kell calculus: A family of higher-order distributed process calculi. In Global Computing, volume 3267 of Lecture Notes in Computer Science, pages 146–178. Springer, 2004.

[Sumii and Pierce] Eijiro Sumii and Benjamin C. Pierce. A bisimulation for dynamic sealing. Theoretical Computer Science, 375(1-3):169–192, 2007. Extended abstract appeared in Proceedings of 31st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, pp. 161–172, 2004.

[Taura] Kenjiro Taura. GXP : An interactive shell for the grid environment. In Innovative

Architecture for Future Generation High-Performance Processors and Systems, pages 59–67, 2004.

論文審査結果の要旨

物理的に分散した複数の計算機が，データのみならずプログラム自体も送受信しながら情報処理を行う高階分散システムが一般的になり，そのようなシステムを形式的にモデル化し，検証するための手法の重要性が増している．既存の代表的な手法として，システムおよびその仕様をそれぞれ高階プロセス計算のプロセスとしてモデル化し，それらの等価性を示すという手法があるが，高階プロセス計算の表現力やプロセス等価性の証明手法が不十分であるという問題があった．本論文は，不動態化(passivation)と名前生成を持つ表現力の高い高階分散プロセス計算を定義し，そのプロセスの等価性証明手法として，環境双模倣と呼ばれる強力な手法を提案するものであり，全編6章からなる．

第1章は序論である．

第2章では，不動態化と名前生成を持つ高階分散プロセス計算 HOπPc を定義し，名前生成と名前制限の違いを議論している．さらに，HOπPc のための環境双模倣を定義し，それに基づく等価性証明手法の健全性および完全性を証明している．さらに例を通して，その手法の有効性を示している．提案手法は，健全かつ完全で，しかも実用性を備えた高階分散プロセスの等価性証明手法として初めてのものであり，きわめて重要な成果である．

第3章では，不動態化と名前制限を持つ高階分散プロセス計算 HOπP のための環境双模倣の定義を与え，観察者の知識を表す「環境」に名前制限が出現しない場合の健全性を証明するとともに，環境に名前制限が出現する場合には健全性が成り立たないことを示している．環境に名前制限が出現しないという制限の下であっても，従来の理論より幅広い等価性証明が可能となっており，有益な成果である．

第4章では，提案する等価性証明手法のさらなる有効性を示すため，現実に用いられている高階分散システム GXP を本研究の高階分散プロセス計算および環境双模倣を用いてモデル化・検証している．このような例は従来の理論では扱うことができなかったため，重要な結果である．

第5章では，関連研究について議論している．

第6章は結論である．

以上要するに本論文は，高階分散システムを形式的にモデル化・検証するための，より強力な計算モデルと等価性証明手法を与えたものであり，情報基礎科学および理論計算機科学の発展に寄与するところが少なくない．

よって，本論文は博士（情報科学）の学位論文として合格と認める．