Rule Formats for Nominal Modal Transition Systems *

Anke Stüber

Universitet Uppsala, Uppsala, Sweden
anke.stuber@it.uu.se

Abstract. Modal transition systems are specification languages that allow the expression of loose specifications. They can model a development process where an initially partial specification is stepwise refined until an implementation is obtained. We propose to generalize modal transition systems to a nominal setting to facilitate the handling of names and binders as encountered in name-passing process calculi. We need to show compositionality of the refinement relation. Beyond that we want to further generalize this idea by compiling SOS rule formats that ensure compositionality. The results of this research should give a general specification language where proofs of the theory carry over to other process calculi expressible in our framework.

1 Problem

In the networked world of today, concurrent communicating systems are ubiquitous. As such systems are complex, we need formal methods to reason about their properties. When developing software we want to start with a specification of the intended behavior of the system that we can later use to check whether our implementation meets the requirements. Process calculi [6, 7] based on labelled transition systems can be used for specification and modeling of concurrent communicating systems.

With process calculi, the languages for specification and implementation of a system are the same. Bisimulation is used to check the correctness of an implementation according to a specification. As bisimulation is an equivalence, this limits the set of possible implementations to a single equivalence class. This means that contrary to a real-world development process all behavior must already be defined in the initial specification. Furthermore for the specification of a system composed of several components it is necessary to specify all behavior of the components, even the behavior that is only internally visible.

Modal transition systems [5] are labelled transition systems that have both *may* and *must* transitions, which can be used to describe the necessary and admissible behavior of a system. This allows for coarser partial specifications that can be refined by either changing a *may* into a *must* transition or removing a *may* transition from the system. When no more *may* transitions are present an

* Initial stage of research, supervised by Joachim Parrow

implementation is reached. For checking the correctness of a refined specification a refinement relation is used. In contrast to the bisimulation equivalence this relation is not an equivalence but a preorder. This makes the correctness check easier in practice.

Name-passing process calculi such as the Pi calculus [7] make use of name generation and scope opening. Names can be scoped to represent local resources. A locally bound name b that is exported over a channel a in the transition $P \xrightarrow{\bar{a}(\nu b)} P'$ can undergo α-conversion. This leads to the the transition $P \xrightarrow{\bar{a}(\nu c)} P'\{c/b\}$ which we want to regard as α-equivalent for each fresh name c. In manual proofs this kind of problem is often treated only informally. Nominal logic [13] strictly formalizes notions of renaming via name-swapping and freshness. We want to generalize the concept of modal transition systems to a name-passing setting, using nominal logic to simplify the handling of names, binders and α-equivalence.

An important property of a specification language is compositionality: The problem of checking the correctness of a complete system according to a specification should be decomposable into the simpler problems of checking correctness of the components. For process calculi, bisimulation preserved by the process operators fulfills compositionality [5]. For our proposed nominal modal transition systems we therefore need to investigate whether our refinement relation is preserved by operators such as the parallel composition operator. For modal transition systems, when there are no *may* transitions, refinement coincides with strong bisimilarity. When there are no *must* transitions, it coincides with simulation. This brings up the question of whether we can find similar relations in the nominal setting.

The area of specification languages is large and there are many papers introducing new languages. Each time the underlying theory such as compositionality has to be proved again. Rule formats [8] are syntactic restrictions on rules for operational semantics that guarantee some desirable properties of the semantics induced by any transition system following the formats. We propose research on rule formats for a general specification language based on nominal modal transition systems that ensure compositionality. Such rule formats would eliminate the need to prove repeatedly the theory for all specification languages that can be expressed through an instance of the rules.

2 Related Work

The process calculus concept was first introduced by Milner in his work on the Calculus of Communicating Systems [6]. Process calculi provide a formal model for reasoning about the behavior and communication of processes. Equivalences between processes can be described in terms of bisimulation. Milner, Parrow and Walker generalize this concept with their Pi-calculus [7] in which processes are allowed to communicate channel names over the communication channels, making it possible to describe dynamic network structures.

In [5] and [4] Larsen and Thomsen argue that process calculi are ill-suited as specification languages, since they limit the set of possible implementations for a specification to a single equivalence class. They propose a specification language in the form of a labelled transition system with *may* and *must* transitions and a refinement relation between specifications.

Process calculi and specification languages rely on the concepts of names and binders. Pitts introduced nominal logic [13] to facilitate the reasoning about α-conversion, freshness and permutation of names. This theory was incorporated into the interactive theorem prover Isabelle [10] with the definitional extension Nominal Isabelle [3, 15] which supports binding multiple names at once. Other frameworks for reasoning about formal semantics such as Twelf [11] and Beluga [12] make use of higher-order abstract syntax to handle names and binders formally.

For process calculi where terms are elements of arbitrary nominal sets Parrow et al. [9] define nominal transition systems and introduce a modal logic that can be used to express bisimulation equivalence by testing whether two processes satisfy the same set of formulas. We will use this paper and its formalization in Nominal Isabelle as a starting point for our research, but unlike this approach start with modal transition systems and generalize them to a nominal setting.

Structural operational semantics (SOS) was first presented by Plotkin in [14] as a structured method to define rules for operational semantics. The idea was widely adopted and extended by the concept of rule formats [8] that guarantee semantic properties by imposing restrictions on the syntax of such rules. Rule formats for name-passing process calculi ensuring that open bisimilarity is a congruence have been studied in [2] by means of category theory and in [16] using the higher-order proof system fold-nabla. Cimini et al. define a framework based on nominal logic for handling names in SOS in [1].

3 Proposed Solution

We propose to combine the concepts of modal transition systems and nominal logic to define a specification language that improves the handling of names and binders as used in name-passing process calculi. This will give us the flexibility of modal transition systems on top of a rigorous theory for reasoning about objects depending on names. We need to define a refinement relation and prove compositionality with the parallel and other process operators. Furthermore we propose to generalize our specification language by finding SOS rule formats such that compositionality holds. We will find encodings of existing process algebras into instances of our rules, eliminating the need to repeat the proofs in each expressible instance.

To increase the confidence in our proofs we will formalize part of our theory in the interactive theorem prover Isabelle. In particular, we will rely on the definitional extension Nominal Isabelle, providing convenient measures of handling names and binders.

4 Preliminary Work

My research on this topic is still in the initial stage. I have conducted a literature survey on process calculi and modal transition systems. In earlier work I gained experience with proofs on properties of bisimulation, notably in a nominal setting. Furthermore, I have had some practice in formalizing proofs in Nominal Isabelle.

5 Expected Contributions

The expected contributions of this research are as follows:

- definition of a specification language using nominal transition systems with modal *may* and *must* transitions, a satisfaction relation and a notion of refinement that allows composition with logical and process operators,
- proofs that our specification language fulfills compositionality with operators such as the parallel operator and comparisons of our refinement relation with existing forms of bisimulation,
- methods for checking refinement of specifications,
- generalization of the concept through rule formats that ensure compositionality,
- applications of our framework by means of encodings of existing process algebras[1],
- formalization of some of the proofs in the interactive theorem prover Isabelle.

This research will generalize nominal process algebras to allow the expression of loose specifications that can be refined stepwise. Our specification language will simplify the task of verification for systems that are composed of several components because of compositionality: It will permit to check the correctness of a refinement step by reducing the test to a correctness check of the refined component. Furthermore, by describing a set of rule formats that ensure compositionality this research will be applicable to various existing process algebras while the desired properties need to be proved only once in the general framework. The proofs written in the theorem prover Isabelle can be reused for process algebras that can be expressed as instances of our rule formats.

6 Plan for Validation and Evaluation

A large part of this research will deal with the theoretical reasoning about features of our proposed rule formats for nominal modal transition systems. To validate that our framework meets the requirements, we will give proofs for the desired properties such as compositionality. We will formalize part of our proofs in Nominal Isabelle to achieve a high confidence in their correctness. Furthermore

[1] exactly which process algebras is not set yet

we will investigate how our specification language can be used in combination with existing tools for model checking.

The advantage of our modal specification language with refinement over approaches using bisimulation can be assessed through performance tests of tools that verify the correctness of an implementation according to a specification when components of a partial implementation are refined. As for the general framework of rule formats we will evaluate its usefulness by encoding existing process calculi in it. For the process calculi we can encode this way the theoretical results will carry over, eliminating the need for new proofs for every expressible calculus.

7 Current Status

Currently I am taking a course on interactive theorem proving in Isabelle to increase my knowledge in that area. Moreover, I am taking a course on category theory that is expected to help with the understanding of papers that use category theory as a means of reasoning about rule formats. As a next step I will conduct a literature survey on structural operational semantics for nominal process algebras. In Swedish universities it is common to write a thesis for a Licentiate degree after completion of half of the studies for a PhD. In this thesis I plan to include the definition of a nominal modal specification language, proofs of its properties such as compositionality, methods for refinement checking and the formalization of the definitions in Nominal Isabelle, due by fall 2018. Subsequently I will work on generalizing the results through rule formats, applications and further formal proofs in Nominal Isabelle for inclusion into the PhD thesis. I expect to finish in 2021.

References

[1] M. Cimini, M. R. Mousavi, M. A. Reniers, and M. J. Gabbay. "Nominal SOS". In: *Electronic Notes in Theoretical Computer Science* 286 (2012), pp. 103–116.

[2] M. Fiore and S. Staton. "A congruence rule format for name-passing process calculi". In: *Information and Computation* 207.2 (2009), pp. 209–236.

[3] B. Huffman and C. Urban. "A New Foundation for Nominal Isabelle". In: *1st International Conference on Interactive Theorem Proving – ITP '10*. Ed. by M. Kaufmann and Paulson L. C. Vol. 6172. Lecture Notes in Computer Science. Berlin a.o.: Springer, 2010, pp. 35–50.

[4] K. G. Larsen. "Modal Specifications". In: *Automatic Verification Methods for Finite State Systems*. Ed. by J. Sifakis. Vol. 407. Lecture Notes in Computer Science. Berlin a.o.: Springer, 1989, pp. 232–246.

[5] K. G. Larsen and B. Thomsen. "A Modal Process Logic". In: *3rd Annual Symposium on Logic in Computer Science – LICS '88*. Washington a.o.: IEEE Computer Society, 1988, pp. 203–210.

[6] R. Milner. "A Calculus of Communicating Systems". In: vol. 92. Lecture Notes in Computer Science. Berlin a.o.: Springer, 1980.

[7] R. Milner, J. Parrow, and D. Walker. "A Calculus of Mobile Processes, I". In: *Information and Computation* 100.1 (1992), pp. 1–40.

[8] M. R. Mousavi, M. A. Reniers, and J. F. Groote. "SOS formats and meta-theory: 20 years after". In: *Theoretical Computer Science* 373.3 (2007), pp. 238–272.

[9] J. Parrow, J. Borgström, L.-H. Eriksson, R. Gutkovas, and T. Weber. "Modal Logics for Nominal Transition Systems". In: *26th International Conference on Concurrency Theory – CONCUR '15*. Ed. by L. Aceto and D. de Frutos Escrig. Vol. 42. Leibniz International Proceedings in Informatics. Wadern: Leibniz-Zentrum für Informatik, 2015, pp. 198–211.

[10] L. C. Paulson. *Isabelle – A Generic Theorem Prover*. Vol. 828. Lecture Notes in Computer Science. Berlin a.o.: Springer, 1994.

[11] F. Pfenning and C. Schürmann. "System Description: Twelf – A Meta-Logical Framework for Deductive Systems". In: *Proceedings of the 16th International Conference on Automated Deduction – CADE-16*. Ed. by H. Ganzinger. Vol. 1632. Lecture Notes in Artificial Intelligence. Berlin a.o.: Springer, 1999, pp. 202–206.

[12] B. Pientka and J. Dunfield. "Beluga: A Framework for Programming and Reasoning with Deductive Systems (System Description)". In: *Proceedings of the 5th International Conference on Automated Reasoning – IJCAR '10*. Ed. by J. Giesl and R. Hähnle. Vol. 6173. Lecture Notes in Computer Science. Berlin a.o.: Springer, 2010, pp. 15–21.

[13] A. M. Pitts. "Nominal Logic: A First Order Theory of Names and Binding". In: *Theoretical Aspects of Computer Software: 4th International Symposium*. Ed. by N. Kobayashi and B. C. Pierce. Vol. 2215. Lecture Notes in Computer Science. Berlin a.o.: Springer, 2001, pp. 219–242.

[14] G. D. Plotkin. *A Structural Approach to Operational Semantics*. Technical Report DAIMI FN-19. Aarhus: Computer Science Department, Aarhus University, 1981.

[15] C. Urban and C. Kaliszyk. "General Bindings and Alpha-Equivalence in Nominal Isabelle". In: *Logical Methods in Computer Science* 8.2 (2012), pp. 1–35.

[16] A. Ziegler, D. Miller, and C. Palamidessi. "A Congruence Format for Name-passing Calculi". In: *Electronic Notes in Theoretical Computer Science* 156.1 (2006), pp. 169–189.