Prof. Luca Aceto

(School of Computer Science, Reykjavik University)

Nominal Structural Operational Semantics (Acronym: NoSOS)

The Icelandic Research Fund 2014

Project Grant — New proposal

Detailed project description

1 State of the art and proficiency

The development of a formal semantics for programming and specification languages is a necessary first step towards rigorous reasoning about them. For instance, a formal semantics allows one to show the correctness of language implementations and is a prerequisite for proving the validity of program optimizations. Operational semantics is a widely-used methodology to define formal semantics for computer languages, which represents the execution of programs as step-by-step development of an abstract machine. *Structural Operational Semantics* (SOS) was introduced by Gordon Plotkin in [74], reprinted in [75], as a logical and structural approach to defining operational semantics. The logical structure of SOS specifications supports a variety of reasoning principles that can be used to prove properties of programs whose semantics is given using SOS. Moreover, SOS language specifications can be used for rapid prototyping of language designs and to provide experimental implementations of computer languages. A recent application field for SOS is the area of model-based systems/software engineering, where increasing emphasis is put on model transformations. SOS plays a crucial role in proving preservation properties of such model transformations. For example, the research group at Eindhoven University of Technology led by Michel Reniers, with whom we have an established track record of research collaboration on SOS-related topics, studies the SOS of languages such as the Compositional Interchange Format (CIF) [29] and Uppaal [24] because, since they wish to specify the behaviour of hybrid systems in CIF and to verify their correctness in Uppaal, they are concerned with the correctness of the transformation from CIF to UPPAAL models [20]. See, e.g., the recent papers [84, 85] for further uses of SOS in model-based systems/software engineering.

Thanks to its intuitive appeal and flexibility, SOS has become the de facto standard for defining operational semantics, and a wealth of programming and executable specification languages have been given formal semantics using it. In recent years much work on the underlying theory as well as on the practice of SOS has been carried out—see, e.g., [14, 68] and [39, 52, 70], respectively. However, a substantial amount of work remains to be done in this area of research. This project will focus on one of the crucial aspects in the definition of semantic models for programming and specification languages that is currently receiving a lot of attention within the framework of the theory of SOS, i.e., a general treatment of concepts such as variables, names and binders.

Many modern programming and specification languages make use of the concepts of names and binders. For example, in the π-calculus [66, 67, 81], names are first-class objects and the whole language is built on the idea that concurrent agents communicate by exchanging names. Binders are syntactic constructs that are used to scope the use of names in expressions. Examples of binders are input-prefixing operators, recursion combinators, restriction operators, infinite-sum operators and the time-integration operator [22, 65, 78, 81]. However, many existing formalizations of languages with the above-mentioned features skirt over the issue of name/variable binding. This has led to a gap between informal intuitions about languages and their formalizations.

Recently, some of the proposers of this project have suggested in [36, 38] a formal and uniform framework called *Nominal SOS* for the treatment of names and binders in SOS, showed that this framework can naturally describe basic notions such as α-conversion and substitution, and applied it to specify two prominent examples of nominal calculi, viz. the lazy λ-calculus [4] and the early π-calculus [81]. In addition, the paper [38] presents a notion of *nominal bisimilarity*, which is the natural counterpart of the classic notion of bisimilarity from [65, 71] in a nominal setting. In the case of the early π-calculus, nominal bisimilarity has been shown in [38] to coincide with Sangiorgi's well-known open bisimilarity [80]. On the other hand, in the setting of the lazy λ-calculus, nominal bisimilarity coincides with Abramsky's applicative bisimilarity [4].

Nominal SOS is based on some aspects of the nominal techniques developed by Gabbay, Pitts and Urban in the seminal papers [45, 73, 88]. Even though such techniques also play an important role in the study of the semantics of the family of the psi-calculi (see, e.g., [26, 27] and the doctoral theses [25, 55]), the Nominal SOS framework develops them in the general setting of the meta-theory of SOS (see, e.g., [14, 68]) and makes them applicable to a wide variety of specific languages. The main aim of this project is to develop the theory and applications of Nominal SOS so that they reach a level of maturity that is comparable to that of the meta-theory of classic SOS, as surveyed in, e.g., [14, 68].

In the light of the importance of the research problem addressed by this project, it is not surprising that Nominal SOS is not the only approach studied so far in the literature that aims at a uniform treatment of binders and names in programming and specification languages. Other existing approaches that accommodate variables and binders within the SOS framework are those proposed by Fokkink and Verhoef in [42], by Middelburg in [62, 63], by Bernstein in [30], by Ziegler, Palamidessi and Miller in [90] and by Fiore and Staton in [40] (originally, by Fiore and Turi in [41]). The aim of all of the above-mentioned frameworks is to establish sufficient syntactic conditions guaranteeing the validity of a semantic result (congruence in the case of [30, 40, 62, 90] and conservativity in the case of [63, 42]). In addition, Gabbay and Mathijssen present a nominal axiomatization of the λ-calculus in [44].

In closely related work, Gacek, Miller and Nadathur in [46, 47], Lakin and Pitts in [57, 58, 59, 60], Miller and Tiu in [64] and Sewell et al. in [82] have also proposed formalizations of binders within SOS-like frameworks with associated tool support. A detailed comparison of these approaches with the one based on Nominal SOS we intend to pursue in this project may be found in [36, Section 5.8]. Nominal techniques have also been implemented within classic automatic theorem provers, such as Isabelle [86, 87], in order to simplify formal reasoning involving binders within such systems.

Proficiency The proposer and his collaborators have been amongst the main contributors to the study of the meta-theory of Structural Operational Semantics for nearly twenty years. To wit, the principal investigator was one of the authors of the invited chapter on SOS [14] in the *Handbook of Process Algebra*. He was also one of the original promoters of the Workshop on SOS, which has run yearly since 2004, and one of the editors of a high-profile journal issue on SOS [12]. Moreover, the further development of the meta-theory of Structural Operational Semantics has been one of the main research goals of the research team over the last five years. (See, e.g., the recent papers [5, 8, 9, 10, 15, 16, 17, 19].) Davide Sangiorgi is one of the leading researchers in the field of calculi for mobile processes, where names and binders play a crucial role, and much of the research to be carried out within the project aims at generalizing some of his work. The results of this project will further strengthen the leading role of the research team within the SOS and the concurrency-theory communities.

The research team has carried out the foundational work in the area of Nominal SOS [38], and has a long history of successful research cooperation leading to about 15 joint papers and the de facto joint supervision of doctoral and MSc. students. The proposed project is a natural continuation of the three-year projects "New Developments in Operational Semantics" (project nr. 060013021) and "Meta-Theory of Algebraic Process Theories" (project nr. 100014021) supported by the Icelandic Research Fund in the period 2008–2010 and 2010–2012, respectively. The wealth of scientific results of those projects may be gleaned by browsing the list of publications from the projects' web pages. The results of the earlier projects and the extensive network of collaborators of the proposers offer an excellent foundation for the development of the proposed research.

2 Objectives of the project and originality

General aim of the proposed research As mentioned earlier, so far Nominal SOS has been applied in [38] to specify two prominent examples of nominal calculi, viz. the lazy λ-calculus [4] and the early π-calculus [81]. These two applications of Nominal SOS indicate that this approach to formalizing languages with names and binders is promising. However, the development of the theory and applications of Nominal SOS is still in its infancy and much more remains to be done. The main general goals of the present proposal are

- to provide further evidence that Nominal SOS is expressive enough to capture the original semantics of nominal calculi, such as value-passing CCS, variants of the (higher-order) π-calculus, the spi-calculus, the psi-calculi and the object calculi, and to prove formally the correspondence between the presentation in terms of Nominal SOS and the original ones;
- to develop the meta-theory of Nominal SOS and to extend a wealth of classic SOS meta-results and techniques to the framework of Nominal SOS; and
- to provide tool support for Nominal SOS.

The overarching aim of the project is to bring the framework of Nominal SOS to a level of maturity that is comparable to that of the standard theory of SOS. The achievement of the goals of the project will offer designers of languages with names and binders, such as those for describing mobile computing devices, a uniform framework and accompanying tool support for the study and rapid prototyping of such languages that is close to that of the classic theory of SOS.

Research problems In order to achieve the aforementioned general goals of the project, we shall have to address a wide variety of research problems. On the one hand, in order to provide compelling evidence for its general applicability, we will apply Nominal SOS to formalize several well-known nominal calculi, such as value-passing CCS [53, 65], variants of the π-calculus [81] and its higher-order version HOπ [79], the psi-calculi [26], the spi-calculus [3] and the object calculi [1]. We also plan to define a variety of standard behavioural semantics for nominal calculi, such as the late, the early and the open bisimilarities [80, 81] and variants of applicative bisimilarity [48, 49, 61], within the Nominal SOS framework. A crucial and large part of the work on the project will be devoted to the development of the meta-theory of Nominal SOS, so that it can reach a level of maturity that is akin to that of the meta-theory of standard SOS. There are a wealth of research questions that can be addressed in this endeavour. Some of them, such as those related to the study of "binding-aware" notions of bisimilarity, are specific to the treatment of names and binders. Some others, such as the development of congruence rule formats and rule formats guaranteeing the validity of a variety of semantic properties, are part of an effort devoted to lifting results from the meta-theory of standard SOS to the setting of Nominal SOS. As part and parcel of the project work, we will develop a prototype tool for the specification, simulation and analysis of languages whose operational semantics is given in terms of Nominal SOS. The tool could be used for rapid prototyping of nominal languages and for checking their semantic properties by syntactic means, much like the framework presented in [15] for classic SOS specifications. Apart from providing an implementation of languages defined using Nominal SOS, such a tool will also allow language designers to experiment with their designs, and will assist them in uncovering design errors and unexpected behaviours at the early stage of the language definition. A more detailed description of the research problems to be tackled during the project work may be found in Section 3 of this proposal.

Originality The importance of the problems addressed by this research proposal is witnessed by the significant and substantial research addressing them in recent years. We have discussed some of the main approaches and results in Section 1, and our work will build on those contributions where appropriate. In particular, some seminal meta-theoretic results about π-calculus-like languages, as well as sophisticated techniques to prove congruence results for such languages, have been developed by Bernstein in [30], by Ziegler, Palamidessi and Miller in [90] and by Fiore and Staton in [40], amongst others. Moreover, Gacek, Miller and Nadathur in [46, 47], Lakin and Pitts in [57, 58, 59, 60], Miller and Tiu in [64] and Sewell et al. in [82] provide mature frameworks for languages with variable binding as well as software tools. However, to the best of our knowledge, nobody has produced a substantial meta-theory for a general framework for languages with variable binding and we strongly believe that this is a worthwhile research goal.

So far, only the basic machinery of the meta-theory of SOS has been extended to the framework of Nominal SOS. The development of a meta-theory of Nominal SOS that can deal with the complexity of nominal languages has not been attempted before and is original in this proposal. Finally, the implementation of the proposed framework and of its associated results in a general-purpose software tool for the specification, rapid prototyping, simulation and analysis of Nominal SOS specifications would extend earlier work on, e.g., the Mobility Workbench [89] and would complement the implementation of the psi-calculi, a family of process calculi that uses the theory of nominal data types, offered in [28, 51].

The meta-theory of languages with variable binding that we intend to develop within the project could conceivably be worked out within one of the frameworks mentioned above. Indeed, some of those frameworks have matured over the years, and so has their associated tool support. Our choice to carry out the work in this project within the (admittedly less developed) framework of Nominal SOS is due to our desire to enrich the classic meta-theory of SOS, based on the notion of *transition system specification* [50], with explicit support for names and name-binding. This will give operational semanticists yet another approach for formalizing the meta-theory of languages with names and binders, and should make the theory we shall develop in the project easily accessible to researchers familiar with the classic theory of SOS.

Timeliness We believe that this is the right time to undertake this research because we now have several necessary ingredients for it. First of all, as we have already mentioned, the results of our earlier joint work and those presented in [38, 11] offer an excellent scientific foundation for the development of the proposed research. During the work on this project, we will also be able to build on the wealth of techniques and results developed in one or more of the frameworks discussed in Section 1.

Moreover, the growing popularity and importance of calculi for mobile and higher-order processes, and the increasing role played by SOS in the area of model-based systems/software engineering (e.g., in proving preservation properties of model transformations) provide further motivation for undertaking this research now. The impact of calculi for mobile and higher-order processes is witnessed, for instance, by their use as foundational formalisms for languages for service-oriented computing [35] (and distributed computation in general [43, 56]), for security [2, 3, 34] and for modelling biological systems [72, 76, 77]. We remark that, in essentially all these applications, foundational calculi such as the π-calculus are not used in their basic form, but are extended with new "domain-specific constructs" that make programming in the resulting languages easier. In general, the "domain-specific" extended languages should inherit desirable semantic properties afforded by the core, foundational languages on which they are built. As argued in, e.g., [31] in the setting of standard SOS, the development of a suitable SOS meta-theory for languages with names and binders will provide a useful foundational framework for such language extensions.

3 Methodology, work plan and timescale

The research work within the project will be carried out by one postdoctoral researcher in very close cooperation with the proposer, his co-proposers and their research teams. In addition, the research project will see the involvement of one MSc. student per year, who will work on some of the technically less challenging research problems that will be tackled during the project and on the implementation of tool support for Nominal SOS. The postdoctoral researcher and the MSc. student will work at the School of Computer Science, Reykjavik University. The embedding of these young researchers within the cooperating research teams is described in Section 5.

Methodology We divide our project into the following four work packages.

WP 1: Modelling calculi using Nominal SOS. In this work package, we will apply Nominal SOS to formalize several well-known nominal calculi. We will start by offering formalizations of value-passing CCS [53, 65] and of several labelled transition system semantics for variants of the π-calculus [81], such as the asynchronous π-calculus [33, 54], and its higher-order version HOπ [79]. We will also provide a formulation of the family of the psi-calculi [26] within Nominal SOS. For each of the above-mentioned calculi, we will study which standard notion of behavioural equivalence between terms, if any, corresponds to the notion of nominal bisimilarity, which is the natural counterpart of the classic notion of bisimilarity from [65, 71] in a nominal setting. Part of this work package will also be devoted to studying the formalization within Nominal SOS of calculi for security, such as the seminal spi-calculus [3], and of some of the object calculi studied by Abadi and Cardelli [1]. We also intend to study Nominal SOS formalizations for a programming language such as Fresh Objective Caml [83] or the recently proposed Nλ [32].

WP 2: Defining different semantics. In [38] a notion of nominal bisimilarity is defined. Nominal bisimilarity is the "cornerstone notion of equivalence" between terms in a language formalized within the framework of Nominal SOS. However, in the standard theory of the π-calculus different bisimulation semantics have been proposed. The most important such equivalences are the late, the early and the open bisimilarities [80, 81]. It is well known that these semantics are all different in the π-calculus. In the λ-calculus [23], important equivalence notions are variants of applicative bisimilarity [48, 49, 61] and the equivalence induced by the notion of Böhm tree. If Nominal SOS is to be a basis to study the meta-theory of computational phenomena in calculi with binders, all these notions must be defined and studied in this general framework. This is the goal of this work package.

WP 3: Meta-theory of Nominal SOS. The goal of the meta-theory of standard SOS is to present a collection of sufficient, syntactic conditions on the rules used to define the operational semantics of programming and specification languages guaranteeing the validity of desirable semantic properties. The development of that meta-theory is, moreover, a way to discover foundational insights on the reason why certain languages afford some semantic properties, while others do not, and the link that these properties have with the syntactic form of SOS rules. The meta-theory of SOS has been extensively studied—see, e.g., the surveys [14, 68]. The aim of this work package is to develop the meta-theory of Nominal SOS so that it can reach a level of maturity that is akin to that of the standard theory of SOS. There are a wealth of research questions that can be addressed in this work package, which is therefore split into the sub-packages listed below.

WP 3.1: Lifting SOS meta-theory to Nominal SOS. Given a semantic specification within the framework of Nominal SOS, an important research challenge is to derive from its syntactic structure, or by means of some algorithm, properties of the labelled transition system that it generates—often in relation to some semantic equivalence that equates labelled transition systems describing the same "behaviour". Typical questions are: Is the labelled transition system finite, finitely branching or image finite? Are the relevant behavioural equivalences congruences for the defined language constructs? Can one automatically generate a sound and complete axiomatization from the SOS specification? Can one guarantee the validity of algebraic laws, such as the associativity, commutativity and idempotence of some operators? Can one automatically derive a fully abstract denotational semantics from the operational specification? Can one use the rules describing the operational semantics to prove suitable equivalences between program contexts algorithmically? These kinds of questions have been addressed and largely answered in the setting of the meta-theory of standard SOS specifications. Our aim in this work package is to generalize some of the meta-theorems and rule formats from the area of SOS to the setting of Nominal SOS. Examples are operational and equational conservativity results like those offered in [13, 42, 63, 69], algorithms for generating sound and complete equational axiomatizations from Nominal SOS specifications [6, 21], rule formats guaranteeing the validity of algebraic properties [5, 8, 9, 10, 19], rule formats for determinism and bounded non-determinism [5, 42], and approaches for generating fully abstract denotational models from Nominal SOS specifications [18].

We are aware that this work package is the riskier one in our proposal. In order to maximize the chances of success, we plan to build as much as possible on the earlier work discussed in Section 1. The first research steps in our work related to this work package will therefore be based on a careful study of the techniques developed by Ziegler, Palamidessi and Miller in [90] and by Fiore and Staton in [40] to prove their congruence results. We shall also build on our own on-going work [11] on the development of a congruence format for nominal bisimilarity, which will be completed as part of the project and will be an important contribution of this work package.

We find it important to remark that, despite its risky nature, this work package has a broad range of largely independent goals, none of which is crucial for the success of the package as a whole. We are confident that a good number of the research questions we plan to pursue in this work package will be addressed successfully.

WP 3.2: Generalizing bisimulation semantics. Nominal SOS opens the possibility to investigate, at a syntactic level and in great generality, many phenomena that are typical of calculi with binders. For instance, in the theory of value-passing CCS and the π-calculus, different bisimulation semantics have been proposed. Amongst these notions of bisimilarity, the most important are the late, the early and, in the case of the π-calculus, the open bisimilarity. It is well known that these semantics are all different in the full π-calculus. However, there are expressive fragments of the π-calculus, such as the asynchronous π-calculus [33, 54], over which those semantics turn out to coincide. One of our goals in this work package is to investigate, at a syntactic level, why this happens and to provide syntactic constraints over language specifications given in the Nominal SOS framework guaranteeing the coincidence of these semantics. A more general goal of this work package is to generalize to classes of Nominal SOS specifications some of the results related to various notions of bisimulation semantics from calculi such as value-passing CCS, the psi-calculi and calculi for security such as the spi-calculus and the applied π-calculus [2].

WP 4: Tool development for Nominal SOS. The theoretical work carried out in the abovementioned work packages will form the basis for the development of a prototype tool for the specification, simulation and analysis of languages whose operational semantics is given in terms of Nominal SOS. The tool could be used for rapid prototyping of nominal languages and for checking their semantic properties by syntactic means.

The work within this package will begin with a careful study of existing work by Gacek, Miller and Nadathur in [46, 47], Lakin and Pitts in [57, 58, 59, 60] and Sewell et al. in [82] and its associated tool support. We will experiment with the use of those existing tools for implementing Nominal SOS specifications and their meta-theory. The tool development will also build on the experience obtained by two of our collaborators in [70] and on work that is currently taking place within the project "Meta-Theory of Algebraic Process Theories" (project nr. 100014021 of the Icelandic Research Fund), which is led by the proposer; see [7, 15]. The implementation of the proposed framework and of its associated results in a general-purpose software tool for the specification, rapid prototyping, simulation and analysis of Nominal SOS specifications will also draw on the earlier work on the Mobility Workbench [89] and on the implementation of the psi-calculi presented in [25, 28, 51].

Work plan and timescale The work on the project will span three years and involves a joint effort from all participants. In keeping with the work pattern in our collaborations so far, the proposer, his collaborators and the postdoctoral researcher will work together on work packages 1–4. The work on work package 4 will mostly be carried by the postdoctoral researcher and the MSc. students hired within the project, under the supervision of the proposer and his senior co-workers. Moreover, the MSc. students may contribute to the less technically challenging research problems that will be addressed during the project.

The work packages described above are largely independent and our work on them can proceed in various alternative, and equally justifiable, ways. However, for the sake of concreteness, we now present the planning for the project work that we favour at the time of writing.

Year 1. The first year of the project will be devoted to the development of our research agenda in work packages 1, 2, 3.1 and 4. At the end of the first year of the project work, we will have formalized value-passing CCS, as well as variants of the π-calculus and its higher-order version HOπ within Nominal SOS, defined and studied the late, the early and the open bisimilarities from value-passing CCS and the π-calculus, as well as variants of applicative bisimilarity within Nominal SOS, lifted some of the simplest meta-results from the theory of SOS to the setting of Nominal SOS, and developed a first prototype tool for inputting and simulating Nominal SOS specifications.

Year 2. The second year of the project will be devoted to the development of our research agenda in work packages 1, 3.1 and 4. At the end of the second year of the project work, we will have formalized the psi-calculi and the spi-calculus within Nominal SOS, lifted further meta-results from the theory of SOS to the setting of Nominal SOS, provided syntactic constraints over language specifications given in the Nominal SOS framework guaranteeing the coincidence of late, early and open bisimilarity, and extended our prototype tool with syntactic checks guaranteeing the validity of some of the semantic properties studied in the first two years of the project.

Year 3. The third and final year of the project will be devoted to the development of our research agenda in work packages 1, 3.1 and 4. At the end of the third year of the project work, we will have formalized some of the object calculi within Nominal SOS, lifted further meta-results from the theory of SOS to the setting of Nominal SOS, and extended our prototype tool with syntactic checks guaranteeing the validity of some of the semantic properties studied in the third year of the project.

Of course, the progress of research work is largely unpredictable, and so is the time that each of the tasks will take to complete. The tentative schedule and milestones described above should therefore be taken with a large pinch of salt.

The proposed research work is undoubtedly ambitious. However, as remarked earlier, the work packages described above are largely independent and partial, or even total, failure in any of them will not have a major negative impact on the success of the project as a whole. We find it therefore unnecessary to develop explicit contingency plans.

4 Milestones and deliverables

As mentioned above, the work packages comprising this proposal are largely independent and so are the different stages in the project work. This means that the work on the project can be very flexible and take advantage of serendipitous discovery and unexpected advances. Based on the work plan described in the previous section, the main milestones and deliverables at the end of each year of the project are expected to be as follows. (These milestones and deliverables should be considered as a complement to the work plan discussed in the previous section. In particular, the list of deliverables is by no means exhaustive. For instance, as is customary for the members of the research team, we shall strive to publish all the project results in the form of research papers.)

Year 1.

Milestones.

1. Formalization of value-passing CCS, as well as variants of the π-calculus and its higher-order version HOπ within Nominal SOS.

2. Congruence format for nominal bisimilarity.

3. Development of a first prototype tool for inputting and simulating Nominal SOS specifications.

Deliverables.

1. Research papers reporting on the results achieved within the first project year.

2. A survey paper describing the Nominal SOS framework and the results of the first year of the project.

3. A publicly available software tool.

Year 2.

Milestones.

1. Formalization of the psi-calculi and the spi-calculus within Nominal SOS.

2. Syntactic constraints over language specifications given in the Nominal SOS framework guaranteeing the coincidence of late, early and open bisimilarity.

3. Rule formats for algebraic properties such as commutativity and idempotence in the setting of Nominal SOS.

Deliverables.

1. Research papers reporting on the results achieved within the second project year.

2. An extended prototype tool with syntactic checks guaranteeing the validity of some of the semantic properties studied in the first two years of the project.

3. Graduation of one MSc. student.

4. Booklet with the preliminary proceedings of the Nordic Workshop on Programming Theory 2015, which we shall organize at Reykjavik University and where the team members will present the work carried out on the project so far.

Year 3.
Milestones.

1. Formalization of some of the object calculi within Nominal SOS.

2. Study of algorithms for the automatic generation of axiomatizations of nominal bisimilarity for classes of Nominal SOS specifications.

3. Extension of our prototype tool with algorithms for the automatic generation of axiomatizations and syntactic checks guaranteeing the validity of some of the semantic properties studied in the third year of the project.

Deliverables.

1. Research papers reporting on the results achieved within the third project year.

2. A survey paper describing the results of the project.

3. A full-blown software tool for the animation and analysis of Nominal SOS specifications.

4. Graduation of one MSc. student.

5. Special issue of an international journal devoted to selected papers from the Nordic Workshop on Programming Theory 2015.

5 Co-operation (domestic/foreign)

The research team involved in the project will consist of Prof. Luca Aceto (proposer, Reykjavik University), Prof. Anna Ingólfsdóttir (co-proposer, Reykjavik University), Prof. MohammadReza Mousavi (co-proposer, Halmstad University, Sweden), Prof. Davide Sangiorgi (co-proposer, University of Bologna, Italy), Dr. Michel Reniers (research collaborator, Eindhoven University of Technology, NL), Dr. Matteo Cimini (postdoctoral researcher who will be hired within the project, presently at LIX, École Polytechnique, France) and two MSc. students. The principal investigator and Prof. Anna Ingólfsdóttir have cooperated in research for over twenty years, and have co-authored one book and over 80 publications. They lead the concurrency group at the School of Computer Science, Reykjavik University, which presently includes two PhD. students, one postdoctoral researcher and two MSc. students. This group has a long history of successful cooperation with Prof. Mousavi and Dr. Reniers leading to over 10 joint papers—see, e.g., our recent joint papers [5, 8, 9, 10, 17, 19]—and the de facto joint supervision of doctoral and master students, as well as the mentoring of young researchers. Prof. Mousavi and Dr. Reniers have done the foundational work on Nominal SOS with Dr. Cimini and will contribute to all the work packages in the project with their expertise on the meta-theory of SOS. Prof. Sangiorgi has been one of the prime movers in the development of the theory of calculi for mobile processes and has already collaborated with Dr. Cimini on the work reported in [37]. Several of the work packages aim at extending some of Sangiorgi's work to the setting of Nominal SOS. In particular, Sangiorgi will contribute to work packages 1, 2 and 3.2.

The postdoctoral researcher and students participating in the project will benefit from the research environment provided by the Icelandic Centre of Excellence in Theoretical Computer Science (ICE-TCS) at Reykjavik University.

This project will also provide us with an ideal opportunity for instigating communication and forging active research collaborations with the research groups led by Prof. Dale Miller at LIX, École Polytechnique, and by Prof. Andrew Pitts at the University of Cambridge, UK. As can be gleaned from this proposal, Dale Miller and Andrew Pitts are two of the leading researchers in the research area to which this project aims to contribute. (It is worth noting that Dr. Cimini is currently a postdoctoral researcher within Dale Miller's group.) Moreover, we shall also explore cooperation possibilities with the group led by Prof. Joachim Parrow and Dr. Björn Victor, the prime movers behind the work on the psi-calculi, at Uppsala University.

6 Contribution of doctoral and master's degree students to the project

The research project will see the involvement of two MSc. students, who will work on some of the technically less challenging research problems that will be tackled during the project and on the implementation of tool support for Nominal SOS. The postdoctoral researcher and the MSc. students will be based at the School of Computer Science, Reykjavik University. The MSc. students will be affiliated with ICE-TCS and will be supervised by Luca Aceto and Anna Ingólfsdóttir. The postdoctoral researcher will also collaborate with the MSc. students and will be actively involved in their supervision.

7 Impact

The project will produce two MSc. theses, technical reports describing preliminary results obtained within the project, peer-reviewed papers and a software tool suite. The results of the project will further enhance the body of knowledge on the meta-theory of SOS and, by extending it to a setting with a first-class treatment of names and binders, will make it more applicable and wide ranging. The supporting software tools that will be developed during the project will be made freely available on the world-wide web. This will make it possible for other academics to use the tools in their research, offer feedback to us and contribute to the tool development themselves. We feel that it is likely that tool development will contribute to making the project very visible within the research communities on SOS and programming-language semantics in general. In order to increase the impact and visibility of the project work, the proposers will also strive to give tutorials at conferences and courses at doctoral schools. Moreover, the work resulting from the first two years of the project will be presented at the Nordic Workshop on Programming Theory 2015, which will be organized by Luca Aceto and Anna Ingólfsdóttir at Reykjavik University. This workshop will provide us with an opportunity to communicate the results of the project to the broad community of programming language researchers, which is an important target audience for us. We also plan to organize an informal workshop in the autumn of 2016 to mark the end of the project and to discuss its findings with some of the colleagues whose work is mentioned in this proposal and with members of the programming language community.

Last, but not least, the publication of the results of the project in internationally recognized, peer-reviewed outlets will increase the visibility of TCS research in Iceland on the international scale, and will entice top scientists in Theoretical Computer Science at different stages in their career to visit the ICE-TCS centre, to develop research collaborations with local scientists, and to contribute to a healthy development of graduate study programmes in Computer Science in Icelandic universities.

8 Proposed publication of results

The results of the project will be published in high-visibility international conferences (such as CONCUR, ETAPS, ICALP and LICS). Full-length papers will be submitted to high-quality scholarly journals such as ACM Transactions on Computational Logic, Information and Computation, Journal of Logic and Algebraic Programming, Logical Methods in Computer Science and Theoretical Computer Science. Moreover, the participants will prepare expository accounts of the achieved results for the Bulletin of EATCS and similar venues. The material developed for tutorials and summer schools will be made available on the world-wide web. The proposers will also publish a booklet with the preliminary proceedings of the Nordic Workshop on Programming Theory 2015 and a special issue of an international journal devoted to selected papers from that meeting.

References

[1] M. Abadi and L. Cardelli. *A Theory of Objects*. Monographs in Computer Science. Springer-Verlag, New York, 1996.

[2] M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In C. Hankin and D. Schmidt, editors, *Conference Record of POPL 2001: The 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, London, UK, January 17–19, 2001*, pages 104–115. ACM, 2001.

[3] M. Abadi and A. D. Gordon. A calculus for cryptographic protocols: The spi calculus. *Information and Computation*, 148(1):1–70, 1999.

[4] S. Abramsky. The lazy lambda calculus. In D. Turner, editor, *Research topics in functional programming*, University Of Texas at Austin Year of Programming, pages 65–116. Addison-Wesley Longman Publishing Co., 1990.

[5] L. Aceto, A. Birgisson, A. Ingólfsdóttir, M. R. Mousavi, and M. A. Reniers. Rule formats for determinism and idempotence. In F. Arbab and M. Sirjani, editors, *Fundamentals of Software Engineering, Third IPM International Conference, FSEN 2009, Kish Island, Iran, April 15–17, 2009, Revised Selected Papers*, volume 5961 of *Lecture Notes in Computer Science*, pages 146–161. Springer, 2010.

[6] L. Aceto, B. Bloom, and F. W. Vaandrager. Turning SOS rules into equations. *Information and Computation*, 111(1):1–52, 1994.

[7] L. Aceto, G. Caltais, E.-I. Goriac, and A. Ingólfsdóttir. PREG axiomatizer - a ground bisimilarity checker for GSOS with predicates. In A. Corradini, B. Klin, and C. Cîrstea, editors, *Algebra and Coalgebra in Computer Science - 4th International Conference, CALCO 2011, Winchester, UK, August 30-September 2, 2011. Proceedings*, volume 6859 of *Lecture Notes in Computer Science*, pages 378–385. Springer, 2011.

[8] L. Aceto, M. Cimini, A. Ingólfsdóttir, M. R. Mousavi, and M. A. Reniers. On rule formats for zero and unit elements. *Electr. Notes Theor. Comput. Sci.*, 265:145–160, 2010.

[9] L. Aceto, M. Cimini, A. Ingólfsdóttir, M. R. Mousavi, and M. A. Reniers. Rule formats for distributivity. In *Proceedings of LATA 2011*, volume 6638 of *Lecture Notes in Computer Science*. Springer, 2011. To appear.

[10] L. Aceto, M. Cimini, A. Ingólfsdóttir, M. R. Mousavi, and M. A. Reniers. SOS rule formats for zero and unit elements. *Theoretical Computer Science*, 2011. To appear.

[11] L. Aceto, M. Cimini, A. Ingólfsdóttir, M. R. Mousavi, and M. A. Reniers. A congruence format for calculi with binders in Nominal SOS. Working paper, 2013.

[12] L. Aceto and W. Fokkink, editors. *Special issue on Structural Operational Semantics*, volume 60–61 of *Journal of Logic and Algebraic Programming*. Elsevier, 2004. 460 pages.

[13] L. Aceto, W. Fokkink, and C. Verhoef. Conservative extension in structural operational semantics. In *Current trends in theoretical computer science*, pages 504–524. World Sci. Publ., River Edge, NJ, 2001.

[14] L. Aceto, W. Fokkink, and C. Verhoef. Structural operational semantics. In *Handbook of Process Algebra*, pages 197–292. Elsevier, Amsterdam, 2001.

[15] L. Aceto, E.-I. Goriac, and A. Ingólfsdóttir. Meta SOS — a Maude based SOS meta-theory framework. To appear, June 2013.

[16] L. Aceto, E.-I. Goriac, and A. Ingólfsdóttir. SOS rule formats for idempotent terms and idempotent unary operators. In P. van Emde Boas, F. C. A. Groen, G. F. Italiano, J. R. Nawrocki, and H. Sack, editors, *SOFSEM 2013: Theory and Practice of Computer Science, 39th International Conference on Current Trends in Theory and Practice of Computer Science, pindlerv Mlýn, Czech Republic, January 26–31, 2013. Proceedings*, volume 7741 of *Lecture Notes in Computer Science*, pages 108–120. Springer-Verlag, 2013.

[17] L. Aceto, E.-I. Goriac, A. Ingólfsdóttir, M. Mousavi, and M. Reniers. Exploiting algebraic laws to improve mechanized axiomatizations. In R. Heickel and S. Milius, editors, *Proceedings of the 5th Conference on Algebra and Coalgebra in Computer Science*, Lecture Notes in Computer Science. Springer-Verlag, 2013. To appear.

[18] L. Aceto and A. Ingólfsdóttir. CPO models for compact GSOS languages. *Information and Computation*, 129(2):107–141, 1996.

[19] L. Aceto, A. Ingólfsdóttir, M. R. Mousavi, and M. A. Reniers. A rule format for unit elements. In J. van Leeuwen, A. Muscholl, D. Peleg, J. Pokorný, and B. Rumpe, editors, *SOFSEM 2010: Theory and Practice of Computer Science, 36th Conference on Current Trends in Theory and Practice of Computer Science, Spindleruv Mlýn, Czech Republic,*

January 23–29, 2010. Proceedings, volume 5901 of *Lecture Notes in Computer Science*, pages 141–152. Springer, 2010.

[20] D. N. Agut, M. Reniers, R. Schiffelers, K. Jørgensen, and D. van Beek. A semantic-preserving transformation from the Compositional Interchange Format to UPPAAL. In *Proceedings of the 18th IFAC World Congress, August 28–September 2, 2011, Milano, Italy*, 2011. In press.

[21] J. C. M. Baeten and E. P. de Vink. Axiomatizing GSOS with termination. *Journal of Logic and Algebraic Programming*, 60–61:323–351, 2004.

[22] J. C. M. Baeten and C. A. Middelburg. *Process Algebra with Timing*. Monographs in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2002.

[23] H. P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, revised edition, 1984.

[24] G. Behrmann, A. David, K. G. Larsen, P. Pettersson, and W. Yi. Developing UPPAAL over 15 years. *Software - Practice and Experience*, 41(2):133–142, 2011.

[25] J. Bengtson. *Formalising Process Calculi*. PhD thesis, Uppsala University, Department of Information Technology, 2010.

[26] J. Bengtson, M. Johansson, J. Parrow, and B. Victor. Psi-calculi: Mobile processes, nominal data, and logic. In *Proceedings of the 24th Annual IEEE Symposium on Logic in Computer Science, LICS 2009, 11–14 August 2009, Los Angeles, CA, USA*, pages 39–48. IEEE Computer Society, 2009.

[27] J. Bengtson, M. Johansson, J. Parrow, and B. Victor. Psi-calculi: A framework for mobile processes with nominal data and logic. *Logical Methods in Computer Science*, 7(1), 2011.

[28] J. Bengtson and J. Parrow. Psi-calculi in Isabelle. In S. Berghofer, T. Nipkow, C. Urban, and M. Wenzel, editors, *Theorem Proving in Higher Order Logics, 22nd International Conference, TPHOLs 2009, Munich, Germany, August 17–20, 2009. Proceedings*, volume 5674 of *Lecture Notes in Computer Science*, pages 99–114. Springer, 2009.

[29] H. Beohar, D. E. N. Agut, D. A. van Beek, and P. J. L. Cuijpers. Hierarchical states in the Compositional Interchange Format. In L. Aceto and P. Sobocinski, editors, *Proceedings*

Seventh Workshop on Structural Operational Semantics, volume 32 of *EPTCS*, pages 42–56, 2010.

[30] K. L. Bernstein. A congruence theorem for structured operational semantics of higher-order languages. In *Thirteenth Annual IEEE Symposium on Logic in Computer Science, Indianapolis, Indiana, USA, June 21–24, 1998*, pages 153–164. IEEE Computer Society, 1998.

[31] B. Bloom, A. Cheng, and A. Dsouza. Using a protean language to enhance expressiveness in specification. *IEEE Transactions on Software Engineering*, 23(4):224–234, 1997.

[32] M. Bojanczyk, L. Braud, B. Klin, and S. Lasota. Towards nominal computation. In J. Field and M. Hicks, editors, *Proceedings of the 39th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2012, Philadelphia, Pennsylvania, USA, January 22–28, 2012*, pages 401–412. ACM, 2012.

[33] G. Boudol. Asynchrony and the pi-calculus. Research Report 1702, INRIA, May 1992.

[34] S. Briais and U. Nestmann. A formal semantics for protocol narrations. *Theoretical Computer Science*, 389(3):484–511, 2007.

[35] L. Caires, R. De Nicola, R. Pugliese, V. T. Vasconcelos, and G. Zavattaro. Core calculi for service-oriented computing. In M. Wirsing and M. M. Hölzl, editors, *Rigorous Software Engineering for Service-Oriented Systems - Results of the SENSORIA Project on Software Engineering for Service-Oriented Computing*, volume 6582 of *Lecture Notes in Computer Science*, pages 153–188. Springer, 2011.

[36] M. Cimini. *Contributions to the Meta-Theory of Structural Operational Semantics*. PhD thesis, School of Computer Science, Reykjavik University, 2011. Available at `http://cimini.info/archive/MatteoCimini_PhdThesis.pdf`.

[37] M. Cimini, C. S. Coen, and D. Sangiorgi. Functions as processes: Termination and the $\overline{\lambda}\mu\widetilde{\mu}$-calculus. In M. Wirsing, M. Hofmann, and A. Rauschmayer, editors, *Trustworthly Global Computing - 5th International Symposium, TGC 2010, Munich, Germany, February 24–26, 2010, Revised Selected Papers*, volume 6084 of *Lecture Notes in Computer Science*, pages 73–86. Springer, 2010.

[38] M. Cimini, M. Mousavi, M. Reniers, and M. J. Gabbay. Nominal SOS. In *Proceedings of the 28th Conference on the Mathematical Foundations of Programming Semantics (MFPS*

XXVIII), Electronic Notes in Theoretical Computer Science. Elsevier, 2012. To appear. A full version of this paper is available at `http://cimini.info/publications/documents/NominalSOS-OnlineResource.pdf`.

[39] C. de O. Braga, E. H. Haeusler, J. Meseguer, and P. D. Mosses. Mapping Modular SOS to rewriting logic. In M. Leuschel, editor, *Logic Based Program Synthesis and Tranformation, 12th International Workshop, LOPSTR 2002, Madrid, Spain, September 17–20,2002, Revised Selected Papers*, volume 2664 of *Lecture Notes in Computer Science*, pages 262–277. Springer, 2003.

[40] M. P. Fiore and S. Staton. A congruence rule format for name-passing process calculi. *Inform. and Comput.*, 207(2):209–236, 2009.

[41] M. P. Fiore and D. Turi. Semantics of name and value passing. In *16th Annual IEEE Symposium on Logic in Computer Science, Boston, Massachusetts, USA, June 16–19, 2001, Proceedings*, pages 93–104. IEEE Computer Society, 2001.

[42] W. Fokkink and C. Verhoef. A conservative look at operational semantics with variable binding. *Inform. and Comput.*, 146(1):24–54, 1998.

[43] L. Fossati, K. Honda, and N. Yoshida. Intensional and extensional characterisation of global progress in the -calculus. In M. Koutny and I. Ulidowski, editors, *CONCUR 2012 - Concurrency Theory - 23rd International Conference, CONCUR 2012, Newcastle upon Tyne, UK, September 4–7, 2012. Proceedings*, volume 7454 of *Lecture Notes in Computer Science*, pages 287–301. Springer, 2012.

[44] M. J. Gabbay and A. Mathijssen. A nominal axiomatization of the lambda calculus. *J. Log. Comput.*, 20(2):501–531, 2010.

[45] M. J. Gabbay and A. M. Pitts. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing*, 13(3–5):341–363, 2002.

[46] A. Gacek, D. Miller, and G. Nadathur. Combining generic judgments with recursive definitions. In *Proceedings of the Twenty-Third Annual IEEE Symposium on Logic in Computer Science, LICS 2008, 24–27 June 2008, Pittsburgh, PA, USA*, pages 33–44. IEEE Computer Society, 2008.

[47] A. Gacek, D. Miller, and G. Nadathur. Reasoning in Abella about structural operational semantics specifications. *Electr. Notes Theor. Comput. Sci.*, 228:85–100, 2009.

[48] A. D. Gordon. *Functional Programming and Input/Output*. Distinguished Dissertations in Computer Science. Cambridge University Press, September 1994.

[49] A. D. Gordon and G. D. Rees. Bisimilarity for a first-order calculus of objects with subtyping. In *Conference Record of POPL'96: The 23rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Papers Presented at the Symposium, St. Petersburg Beach, Florida, 21–24 January 1996. ACM Press, New York, 1996*, pages 386–395, 1996.

[50] J. F. Groote and F. W. Vaandrager. Structured operational semantics and bisimulation as a congruence. *Information and Computation*, 100(2):202–260, 1992.

[51] R. Gutkovas. Exercising Psi-calculi: A Psi-calculi workbench. Master's thesis, Uppsala Universitet, June 2011.

[52] P. H. Hartel. Letos - a lightweight execution tool for operational semantics. *Software - Practice and Experience*, 29(15):1379–1416, 1999.

[53] M. Hennessy and A. Ingólfsdóttir. A theory of communicating processes with value passing. *Information and Computation*, 107(2):202–236, 1993.

[54] K. Honda and M. Tokoro. An object calculus for asynchronous communication. In P. America, editor, *ECOOP'91 European Conference on Object-Oriented Programming, Geneva, Switzerland, July 15–19, 1991, Proceedings*, volume 512 of *Lecture Notes in Computer Science*, pages 133–147. Springer, 1991.

[55] M. Johansson. *Psi-calculi: a Framework for Mobile Process Calculi*. PhD thesis, Uppsala University, Department of Information Technology, 2010.

[56] N. Kobayashi and D. Sangiorgi. A hybrid type system for lock-freedom of mobile processes. *ACM Transactions on Programming Languages and Systems*, 32(5), 2010.

[57] M. R. Lakin. *An Executable Meta-language for Inductive Definitions with Binders*. PhD thesis, University of Cambridge, 2010.

[58] M. R. Lakin and A. M. Pitts. A metalanguage for structural operational semantics. In *Symposium on Trends in Functional Programming*, pages 1–16, 2007.

[59] M. R. Lakin and A. M. Pitts. Resolving inductive definitions with binders in higher-order typed functional programming. In G. Castagna, editor, *Programming Languages and Sys-*

tems, 18th European Symposium on Programming, ESOP 2009, volume 5502 of *Lecture Notes in Computer Science*, pages 47–61. Springer, 2009.

[60] M. R. Lakin and A. M. Pitts. Encoding abstract syntax without fresh names. *Journal of Automated Reasoning*, 49(2):115–140, 2012.

[61] S. B. Lassen. *Relational Reasoning about Functions and Nondeterminism*. PhD thesis, Department of Computer Science, University of Aarhus, 1998.

[62] C. A. Middelburg. Variable binding operators in transition system specifications. *J. Log. Algebr. Program.*, 47(1):15–45, 2001.

[63] C. A. Middelburg. An alternative formulation of operational conservativity with binding terms. *J. Log. Algebr. Program.*, 55(1-2):1–19, 2003.

[64] D. Miller and A. Tiu. A proof theory for generic judgments. *ACM Trans. Comput. Log.*, 6(4):749–783, 2005.

[65] R. Milner. *Communication and Concurrency*. Prentice-Hall International, Englewood Cliffs, 1989.

[66] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, i. *Inf. Comput.*, 100(1):1–40, 1992.

[67] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, ii. *Inf. Comput.*, 100(1):41–77, 1992.

[68] M. Mousavi, M. Reniers, and J. F. Groote. SOS formats and meta-theory: 20 years after. *Theoretical Computer Science*, 373(3):238–272, 2007.

[69] M. R. Mousavi and M. A. Reniers. Orthogonal extensions in structural operational semantics. In L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11–15, 2005, Proceedings*, volume 3580 of *Lecture Notes in Computer Science*, pages 1214–1225. Springer, 2005.

[70] M. R. Mousavi and M. A. Reniers. Prototyping SOS meta-theory in Maude. *Electron. Notes Theor. Comput. Sci.*, 156:135–150, May 2006.

[71] D. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *5th GI Conference,* Karlsruhe, Germany, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, 1981.

[72] A. Phillips, L. Cardelli, and G. Castagna. A graphical representation for biological processes in the stochastic pi-calculus. *Transactions on Computational Systems Biology VII*, 4230:123–152, 2006.

[73] A. M. Pitts. Nominal logic, a first order theory of names and binding. *Information and Computation*, 186(2):165–193, 2003.

[74] G. D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.

[75] G. D. Plotkin. A structural approach to operational semantics. *Journal of Logic and Algebraic Programming*, 60–61:17–139, 2004.

[76] C. Priami, A. Regev, E. Y. Shapiro, and W. Silverman. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, 80(1):25–31, 2001.

[77] A. Regev, W. Silverman, and E. Y. Shapiro. Representation and simulation of biochemical processes using the pi-calculus process algebra. In *Pacific Symposium on Biocomputing*, pages 459–470, 2001.

[78] M. A. Reniers, J. F. Groote, M. van der Zwaag, and J. van Wamel. Completeness of timed mcrl. *Fundamenta Informaticae*, 50(3–4):361–402, 2002.

[79] D. Sangiorgi. pi-calculus, internal mobility, and agent-passing calculi. *Theoretical Computer Science*, 167(1–2):235–274, 1996.

[80] D. Sangiorgi. A theory of bisimulation for the π-calculus. *Acta Informatica*, 33:69–97, 1996.

[81] D. Sangiorgi and D. Walker. *The π-Calculus: A Theory of Mobile Processes*. Cambridge University Press, Cambridge, 2001. With a foreword by Robin Milner.

[82] P. Sewell, F. Z. Nardelli, S. Owens, G. Peskine, T. Ridge, S. Sarkar, and R. Strnisa. Ott: Effective tool support for the working semanticist. *J. Funct. Program.*, 20(1):71–122, 2010.

[83] M. R. Shinwell and A. M. Pitts. Fresh Objective Caml user manual. Technical Report UCAM-CL-TR-621, University of Cambridge Computer Laboratory, Feb. 2005.

[84] F. Stappers, S. Weber, M. A. Reniers, S. Andova, and I. Nagy. Formalizing a domain specific language using SOS: An industrial case study. In *Proceedings of the 4th International Conference on Software Language Engineering (SLE 2011)*, Lecture Notes in Computer Science. Springer, 2011. To appear.

[85] F. P. M. Stappers, M. A. Reniers, and S. Weber. Transforming SOS specifications to linear processes. In G. Salaün and B. Schätz, editors, *Formal Methods for Industrial Critical Systems - 16th International Workshop, FMICS 2011, Trento, Italy, August 29–30, 2011. Proceedings*, volume 6959 of *Lecture Notes in Computer Science*, pages 196–211. Springer, 2011.

[86] C. Urban. Nominal techniques in Isabelle/HOL. *J. Autom. Reasoning*, 40(4):327–356, 2008.

[87] C. Urban and C. Kaliszyk. General bindings and alpha-equivalence in Nominal Isabelle. In G. Barthe, editor, *Programming Languages and Systems - 20th European Symposium on Programming, ESOP 2011, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2011, Saarbrücken, Germany, March 26–April 3, 2011. Proceedings*, volume 6602 of *Lecture Notes in Computer Science*, pages 480–500. Springer, 2011.

[88] C. Urban, A. M. Pitts, and M. J. Gabbay. Nominal unification. *Theoretical Computer Science*, 323(1–3):473–497, 2004.

[89] B. Victor and F. Moller. The mobility workbench - a tool for the pi-calculus. In D. L. Dill, editor, *Computer Aided Verification, 6th International Conference, CAV '94, Stanford, California, USA, June 21–23, 1994, Proceedings*, volume 818 of *Lecture Notes in Computer Science*, pages 428–440. Springer, 1994.

[90] A. Ziegler, D. Miller, and C. Palamidessi. A congruence format for name-passing calculi. *Electr. Notes Theor. Comput. Sci.*, 156(1):169–189, 2006.