

Integration von Polymorphismus und Subtypen für das Pi-Kalkül

In diesem Paper möchten wir Fortschritte unserer laufenden Arbeit präsentieren, das Sigma-Kalkül zweiter Stufe in einem geeigneten beschränkt polymorphen Pi-Kalkül zu interpretieren.

Der Pi-Kalkül [Mil92] ist eine Prozeßalgebra, die das Konzept der nachrichtenbasierten Kooperation mittels Kommunikation von Kanalnamen formalisiert. Er erlaubt die Beschreibung von Prozeß-Systemen, in denen sich die Kommunikationstopologie dynamisch ändert. Der Pi-Kalkül eröffnete eine neue Perspektive der Untersuchung von Konzepten objektorientierter Programmiersprachen [Wal91]. Der Sigma-Kalkül [AC96] ist als minimale objektbasierte Programmiersprache mit Objekt-Formation, Methoden-Aufruf und Methoden-Update aufzufassen.

Fokus unserer Arbeit ist der Ansatz, den Sigma-Kalkül im Pi-Kalkül zu interpretieren. Dieser Ansatz ist für Typsysteme erster Ordnung bereits gründlich ausgearbeitet. Die Motivation liegt im wechselseitig vertieften Verständnis von Objekten und Prozessen einerseits [San98], in der Erschließung neuer Beweistechniken andererseits [Kle00]. Vor diesem Hintergrund gilt unser Interesse nun fortgeschrittenen Typsystemen. Die bestehenden Interpretationen sollen konsequent bzgl. Polymorphismus fortgesetzt werden. Die Schwierigkeit ist dabei, daß sich im Kontext von Prozessen existenzquantifizierte Typen als die natürliche Form von Polymorphismus erwiesen haben [San01], während Generezität in objektbasierten Sprachen hingegen durch allquantifizierte Typen modelliert wird, in voller Analogie zu parametrischem Polymorphismus funktionaler Sprachen [Rey83].

Im vorliegenden Paper werden Pi-Kalkül-Typsysteme mit Polymorphismus einerseits [Tur96] und Subtypen andererseits [PS96] zu einem neuen Pi-Kalkül mit beschränktem Polymorphismus integriert. Als motivierendes Beispiel wird die natürliche Darstellung einer objektorientierten Server-Hierarchie im resultierenden Kalkül präsentiert. Technische Beiträge sind zum einen Beweise der notwendigen Eigenschaften des Typsystem, das heißt ein Subjekt-Reduktionstheorem und Theoreme zur Typ-Stärkung und -Schwächung. Außerdem beweisen wir korrektes Laufzeitverhalten wohlgetypter Prozesse. Darüberhinaus werden Aspekte der denotationellen Semantik des Kalküls im Stil einer intuitiven Semantik [MP88] auf Basis existierender vollständig abstrakter Modelle [Sta96] behandelt .

[AC96] Martin Abadi, Luca Cardelli. A Theory of Objects. Monographs in Computer Science, Springer-Verlag, 1996

[Kle00] Josva Kleist. Reasoning about Objects using Process Techniques, Ph.D. Thesis. Aalborg University, 2000

[Mil92] Robin Milner, Joachim Parrow, David Walker. A Calculus for Mobile Processes, Part I/II. Information and Computation 100, pp. 1-40/41-77, 1992

[MP88] John C. Mitchell, Gordon D. Plotkin. Abstract Types have Existential Type. ACM Transaction on Programming Languages and Systems, vol. 10, no. 3, 1988

[PS96] Benjamin Pierce, Davide Sangiorgi. Typing and Subtyping for Mobile Processes. In: Mathematical Structures in Computer Science, pp. 409-454, 1996

[Rey83] John C. Reynolds. Types, Abstraction and Parametric Polymorphism. In: Information Processing 83, Elsevier Science Publishers, 1983

[San98] Davide Sangiorgi. An Interpretation of Typed Objects into Pi-Calculus. In: Information and Computation, pp. 34-73, Academic Press, 1998

[San01] Davide Sangiorgi. The Pi-calculus: a Theory of Mobile Processes. to appear

[Sta96] Ian Stark. A Fully Abstract Domain Model for the Pi-calculus. In: Logic in Computer Science 96, pp. 43-55. IEEE, 1996

[Tur 96] David N. Turner. The Polymorphic Pi-Calculus: Theory and Implementation. Ph.D. Thesis. University of Edinburgh. 1996

[Wal91] David Walker. Pi-calculus Semantics of Object-Oriented Programming Languages. In: Theoretical Aspects of Computer Science, pp. 532-547. Springer-Verlag, 1991