Fakultät II: Informatik, Wirtschafts- und Rechtswissenschaften
Department für Informatik
Abteilung: Entwicklung korrekter Systeme

Transformational semantics of the combination π-*OZ* for mobile processes with data

Masterarbeit

- post version -

Name:	Muhammad Ekbal Ahmad
E-Mail:	muhammad.ekbal.ahmad@uni-oldenburg.de

Studiengang:	Fach-Master Informatik
Erstgutachter:	Prof. Dr. Ernst-Rüdiger Olderog
Zweitgutachter:	M.Sc. Manuel Gieseking
Datum:	17. Januar 2020

Contents

List of Figures

1 Introduction

In many cases of modern computing it is of interest to describe and model concurrency. Computers no longer just solve a problem by subsequently working off the single tasks of their own, but they decompose and concurrently calculate the problem even together in a network. The increase in the number of CPU cores and more heavily of GPU cores within one single computer convincingly demonstrates how fundamental concurrency is for modern computing. Moreover, the rapidly increasing spread of the Internet is one of the most common examples which shows the importance of networks.

This thesis is divided into five chapters. In Chapter 2 we briefly introduce sequences and properly investigate the π-calculus and its operational semantics (the *early transition system* [SW01]). Thereby, we investigate its properties and define the refinement based on the trace semantics. Finally, the conclusion in Chapter 3 gives a brief summary of our results and presents ideas for future work.

2 Preliminaries

At the heart of the refinement of π-*calculus* processes is the theory of *sequences*. Thus, in this chapter, we recall the model of sequences to gain a formal construct to handle ordered elements.

Furthermore, we introduce the π-calculus and investigate its behavior properly. In particular, we carefully explain the operational semantics of π-calculus processes, since its peculiarities induce the characteristics of the refinement and its properties. Moreover, we discuss why we choose this particular operational semantics for the following work in this thesis and compare it to other semantics.

The majority of those definitions and notions can, for example, be found in [Mil99, SW01].

As mathematical notations, we consider the natural numbers starting with zero ($\mathbb{N} = \{0, 1, 2, \dots\}$) and use $\fatsemi$ as the composition of relations. Furthermore, we denote R^* as the reflexive and transitive closure of a relation R.

2.1 The π-calculus

The π-*calculus* is a process algebra that can be used to describe the behavior. This section introduces the polyadic pure version of the π-calculus as decriped in [Mil99].

2.1.1 Syntax

Definition 2.1.1 (Process syntax) The syntax of a π-calculus process P is defined by:

$$P ::= \sum_{i \in I} \pi_i.P_i \;\Big|\; P_1 \mid P_2 \;\Big|\; \underline{\texttt{new}}\, \vec{y}\, P.$$

where:

- $\sum_{i \in I} \pi_i.P_i$ is the guarded sum.

- $P_1 \mid P_2$ is the parallel composition of processes.

- $\underline{\texttt{new}}\, \vec{y}\, P$ is the restriction of the scope of the names $\vec{y}$ to the process P $\ast$

***Guarded sum*:**

Is the *choice* between multiple guarded processes. If the guard of one process took place, other guarded processes will be discarded. For example, the processes: $x().A + b().B$ will evolve to the process A if the guard $x()$ occurred.

***Guard*:**

Also called *action prefix* and denoted by π. It's syntax is defined by:

Definition 2.1.2 (Action prefix syntax)

$$\pi ::= x(\vec{y}) \;\Big|\; \overline{x}\langle\vec{y}\rangle \;\Big|\; \tau.$$

where:

- $x(\vec{y})$ represents the action: receive $\vec{y}$ via the channel x.

- $\overline{x}\langle\vec{y}\rangle$ represents the action: send $\vec{y}$ via the channel x.

- τ represents an internal non observable action. $\ast$

***Parallel composition*:**

The parallel composition operator $\mid$ represent the concept of concurrency in the π-calculus, where two processes can evolve in concurrent.It represents an interleaving behavior of the concurrency. For example let: $P =_{\text{def}} P_1 \mid (P_2 \mid P_3)$ where: $P_1 =_{\text{def}} x(y).A$, $P_2 =_{\text{def}} \overline{x}\langle y\rangle.B$ and $P_3 =_{\text{def}} x(y).C$. So $P =_{\text{def}} x(y).A \mid (\overline{x}\langle y\rangle.B \mid x(y).C)$. Possible evolution cases of P are:

- $P_1 \mid (B \mid C)$. P_2 sends y via x to P_3.

- $A \mid (B \mid P_3)$. P_2 sends y via x to P_1.

The example above illustrated the privacy nature of the parallel operator in the π-calculus. A process can via a channel communicate with only one process pro time. P_2 cannot communicate with both P_1, P_3 in the same time, while in Communicating Sequential Processes (CSP) a process can communicate with multiple processes in the same time via the same channel by sending multiple copies of the same message, in other words: in CSP the channel represents a Hub.

***Restriction*:**

The expression $\underline{\texttt{new}}\,\vec{y}\ P$ binds the names $\vec{y}$ to the process P. In other words: the visibility scope of the names $\vec{y}$ is restricted to the process P. It is similar to declaring a private variable in programming languages. Thus the names $\vec{y}$ are not visible outside P and P cannot use them to communicate with outside. For example, let $P =_{\text{def}} P_1 \mid P_2$ where: $P_1 =_{\text{def}} \underline{\texttt{new}}\, y\ \overline{y}\langle z\rangle.A$ and $P_2 =_{\text{def}} y(z).B$. The process P cannot evolute to $A \mid B$, since the name y in P_1 is only visible inside it, i.e., from the P_2's point of view P_1 doesn't have a channel called y. This takes us to the definition of the Bound and free names.

Definition 2.1.3 (Free names) are all the restricted names in a process. $*$

Definition 2.1.4 (Bound names) are all the name that occur in a process except the bound names. $*$

For example, let $P_1 =_{\text{def}} \underline{\texttt{new}}\, x\ \overline{x}\langle y\rangle.P_2$ where $P_2 =_{\text{def}} \underline{\texttt{new}}\, z\ \overline{x}\langle z\rangle.B$. The name x is bound in P_1 but free in P_2.

2.2 The OZ

The OZ

3 Conclusion and future work

In this thesis ...

Bibliography

[Mil99] R. Milner. *Communicating and Mobile Systems: the π-Calculus.* Cambridge University Press, Cambridge, England, 1999.

[SW01] D. Sangiorgi and D. Walker. *The π-calculus: a Theory of Mobile Processes.* Cambridge University Press, Cambridge, England, 2001.

Index

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Außerdem versichere ich, dass ich die allgemeinen Prinzipien wissenschaftlicher Arbeit und Veröffentlichung, wie sie in den Leitlinien guter wissenschaftlicher Praxis der Carl von Ossietzky Universität Oldenburg festgelegt sind, befolgt habe.

Oldenburg, den 17. Januar 2020

(Muhammad Ekbal Ahmad)