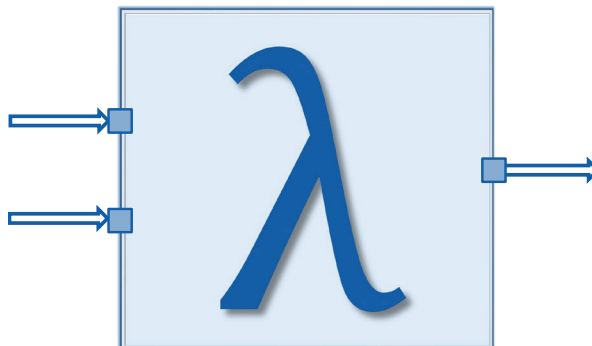


RWTH Aachen University
Software Engineering Group

Towards an Isabelle Theory for distributed, interactive systems – the untimed case

Technical Report



Jens Christoph Bürger
Hendrik Kausch
Deni Raco
Jan Oliver Ringert
Bernhard Rumpe
Sebastian Stüber
Marc Wiartalla

Aachener Informatik-Berichte,
Software Engineering

Hrsg: Prof. Dr. rer. nat. Bernhard Rumpe

Band 45

Aachener Informatik-Berichte, Software Engineering

herausgegeben von
Prof. Dr. rer. nat. Bernhard Rumpe
Software Engineering
RWTH Aachen University

Band 45

**Jens Christoph Bürger, Hendrik Kausch, Deni Raco,
Jan Oliver Ringert, Prof. Dr. rer. nat. Bernhard Rumpe,
Sebastian Stüber, Marc Wiartalla,**
RWTH Aachen University

Towards an Isabelle Theory for distributed, interactive systems

The untimed case

Shaker Verlag
Düren 2020

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <http://dnb.d-nb.de>.

Copyright Shaker Verlag 2020

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publishers.

Printed in Germany.

ISBN 978-3-8440-7265-5

ISSN 1869-9170

Shaker Verlag GmbH • Am Langen Graben 15a • 52353 Düren

Phone: 0049/2421/99011-0 • Telefax: 0049/2421/99011-9

Internet: www.shaker.de • e-mail: info@shaker.de

Abstract

This report describes a specification and verification framework for distributed interactive systems. The framework encodes the untimed part of the formal methodology FOCUS [BS01] in the proof assistant Isabelle [Pau90] using domain-theoretical concepts. The key concept of FOCUS, the stream data type, together with the corresponding prefix-order, is formalized as a pointed complete partial order. Furthermore, a high-level API is provided to hide the explicit usage of domain theoretical concepts by the user in typical proofs. Realizability constraints for modeling component networks with potential feedback loops are implemented. Moreover, a set of commonly used functions on streams are defined as least fixed points of the corresponding functionals and are proven to be prefix-continuous.

As a second key concept the stream processing function (SPF) is introduced describing a statefull, deterministic behavior of a message-passing component. The denotational semantics of components in this work is a defined set of stream processing functions, each of which maps input streams to output streams.

Furthermore, an extension of the framework is presented by using an isomorphic transformation of tuples of streams to model component interfaces and allowing composition. The structures for modeling component networks are implemented by giving names to channels and defining composition operators. This is motivated by the advantage that a modular modeling of component networks offers, based on the correctness of components of the decomposed system and using proper composition operators, the correctness of the whole system is automatically derived by construction.

To facilitate automated reasoning, a set of theorems is proven covering the main properties of these structures. Moreover, essential proof methods such as stream-induction are introduced and support these by further theorems. These examples demonstrate the principle usability of the modeling concepts of FOCUS and the realized verification framework for distributed systems with security and safety issues such as cars, airplanes, etc. Finally, a running example extracted from a controller in a car is realized to demonstrate and validate the framework.

Contents

1	Introduction	1
1.1	Goals and Results	4
2	Foundations of Domain Theory	6
2.1	Partial Orders	6
2.2	Domains	7
2.3	Functions	8
	Function Domains	8
2.4	Fixed-Points	9
2.4.1	Motivation: Recursive Definitions	10
2.4.2	Fixed-Point Theorems	11
2.4.3	Relation Between Monotonic/Continuous Functions and Least Fixed-Points	12
2.4.4	Predicates and Admissibility	13
2.4.5	Fixed-Point Induction	13
2.4.6	Construction of Admissible Predicates and Continuous Functions	13
3	Introduction to Isabelle/HOLCF	14
3.1	Isabelle/HOL	14
3.1.1	Isabelle's Type System	14
3.1.2	Defining Types	15
3.2	Function and Class Definitions	16
3.3	Domains in Isabelle	16
	Lifting Datatypes to Domains	17
	The Domain Type-Constructor	17
	CPOs on Subtypes	18
3.4	Continuous Functions and Fixed Points	19
3.5	Proofs in Isabelle	19

4	Extensions of HOLCF	22
4.1	Prelude	22
4.2	Properties of Set Orderings	23
4.3	Lazy Natural Numbers	24
4.3.1	Definition	24
4.3.2	Properties of the Data Type	25
5	Streams	27
5.1	Mathematical Definition and Construction	28
5.1.1	Properties of Streams	29
5.2	Streams in Isabelle	29
5.2.1	Running Example: The Addition-Component	30
5.2.2	The Take-Functional and Induction on Streams	32
5.2.3	Concatenation of Streams	32
5.2.4	Reusing List Theories	33
5.2.5	The Length Operator	34
5.2.6	The Domain Operator	35
5.2.7	Defining Functions with Explicitly Memorized State	36
5.2.8	Map, Filter, Zip, Project, Merge and Removing Duplicates	36
5.2.9	Infinite Streams and Kleene Theorem	38
5.3	Further Kinds of Streams	38
6	Stream Bundles	40
6.1	Mathematical Definition	40
6.2	System specific Datatypes	41
6.2.1	Channel Datatype	41
6.2.2	Message Datatype	42
6.2.3	Domain Classes	42
6.2.4	Interconnecting Domain Types	45
	Union Type	45
	Minus Type	46
6.3	Stream Bundle Elements	46
6.4	Stream Bundles Datatype	47
6.5	Functions for Stream Bundles	48

Converter from sbElem to SB	48
Extracting a single stream	49
Concatenation	51
Length of SBs	51
Dropping Elements	52
Taking Elements	53
Concatenating sbElems with SBs	53
Converting Domains of SBs	55
Union of SBs	56
Renaming of Channels	57
Lifting from Stream to Bundle	58
Overview of all functions	59
7 Stream Processing Functions	61
7.1 Mathematical Definition	61
7.2 Composition of SPFs	62
Sequential Composition Operator	62
Parallel Composition Operator	63
Feedback Composition Operator	63
7.3 Stream Processing Functions in Isabelle	64
7.4 General Composition Operators	65
7.5 Overview of SPF Functions	68
8 Stream Processing Specification	70
8.1 Mathematical Definition	70
8.2 General Composition of SPSs	70
8.3 Special Composition of SPSs	72
8.4 SPS Completion	72
8.5 Overview of SPS Functions	74
9 Case Study: Cruise Control	76
References	81
Glossary	85

Appendices	87
A Extensions of HOLCF Theories	89
A.1 Prelude	89
A.2 Set Orderings	96
A.3 Lazy Naturals	102
B Stream Theories	114
B.1 Streams	114
C Stream Bundle Theories	177
C.1 Datatype	177
C.2 Channel	178
C.3 SBelem Data Type	181
C.4 SB Data Type	184
D Stream Processing Function Theories	213
D.1 SPF Data Type	213
D.2 Composition	217
E Stream Processing Specification Theories	222
F Case Study	226