Walter Dosch
Roger Y. Lee
Chisu Wu (Eds.)

LNCS 3647

Software Engineering Research, Management and Applications

Second International Conference, SERA 2004
Los Angeles, CA, USA, May 2004
Selected Revised Papers

Springer

Walter Dosch Roger Y. Lee
Chisu Wu (Eds.)

Software Engineering Research, Management and Applications

Second International Conference, SERA 2004
Los Angeles, CA, USA, May 5-7, 2004
Selected Revised Papers

Springer

Volume Editors

Walter Dosch
University of Lübeck
Institute of Software Technology and Programming Languages
Ratzeburger Allee 160, 23538 Lübeck, Germany
E-mail: dosch@isp.uni-luebeck.de

Roger Y. Lee
Central Michigan University
Software Engineering and Information Technology Institute
Mt.Pleasant,MI 48858, P.O. Box , USA
E-mail: lee@cps.cmich.edu

Chisu Wu
Seoul National University
School of Computer Science and Engineering
Seoul 151-742, South Korea
E-mail: wuchisu@selab.snu.ac.kr

Library of Congress Control Number: 2005939044

CR Subject Classification (1998): D.2, I.2.11, C.2.4, H.5.2-3, K.6

LNCS Sublibrary: SL 2 – Programming and Software Engineering

ISSN	0302-9743
ISBN-10	3-540-32133-0 Springer Berlin Heidelberg New York
ISBN-13	978-3-540-32133-0 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11668855 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3647

Lecture Notes in Computer Science

For information about Vols. 1–3770

please contact your bookseller or Springer

Vol. 3870: S. Spaccapietra, P. Atzeni, W.W. Chu, T. Catarci, K.P. Sycara (Eds.), Journal on Data Semantics V. XIII, 237 pages. 2006.

Vol. 3863: M. Kohlhase (Ed.), Mathematical Knowledge Management. XI, 405 pages. 2006. (Sublibrary LNAI).

Vol. 3861: J. Dix, S.J. Hegner (Eds.), Foundations of Information and Knowledge Systems. X, 331 pages. 2006.

Vol. 3860: D. Pointcheval (Ed.), Topics in Cryptology – CT-RSA 2006. XI, 365 pages. 2006.

Vol. 3858: A. Valdes, D. Zamboni (Eds.), Recent Advances in Intrusion Detection. X, 351 pages. 2006.

Vol. 3857: M. Fossorier, H. Imai, S. Lin, A. Poli (Eds.), Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. XI, 350 pages. 2006.

Vol. 3855: E. A. Emerson, K.S. Namjoshi (Eds.), Verification, Model Checking, and Abstract Interpretation. XI, 443 pages. 2005.

Vol. 3853: A.J. Ijspeert, T. Masuzawa, S. Kusumoto (Eds.), Biologically Inspired Approaches to Advanced Information Technology. XIV, 388 pages. 2006.

Vol. 3852: P.J. Narayanan, S.K. Nayar, H.-Y. Shum (Eds.), Computer Vision - ACCV 2006, Part II. XXXI, 977 pages. 2005.

Vol. 3851: P.J. Narayanan, S.K. Nayar, H.-Y. Shum (Eds.), Computer Vision - ACCV 2006, Part I. XXXI, 973 pages. 2006.

Vol. 3850: R. Freund, G. Păun, G. Rozenberg, A. Salomaa (Eds.), Membrane Computing. IX, 371 pages. 2006.

Vol. 3848: J.-F. Boulicaut, L. De Raedt, H. Mannila (Eds.), Constraint-Based Mining and Inductive Databases. X, 401 pages. 2006. (Sublibrary LNAI).

Vol. 3847: K.P. Jantke, A. Lunzer, N. Spyratos, Y. Tanaka (Eds.), Federation over the Web. X, 215 pages. 2006. (Sublibrary LNAI).

Vol. 3844: J.-M. Bruel (Ed.), Satellite Events at the MoDELS 2005 Conference. XIII, 360 pages. 2006.

Vol. 3843: P. Healy, N.S. Nikolov (Eds.), Graph Drawing. XVII, 536 pages. 2006.

Vol. 3842: H.T. Shen, J. Li, M. Li, J. Ni, W. Wang (Eds.), Advanced Web and Network Technologies, and Applications. XXVII, 1057 pages. 2006.

Vol. 3841: X. Zhou, J. Li, H.T. Shen, M. Kitsuregawa, Y. Zhang (Eds.), Frontiers of WWW Research and Development - APWeb 2006. XXIV, 1223 pages. 2006.

Vol. 3840: M. Li, B. Boehm, L.J. Osterweil (Eds.), Unifying the Software Process Spectrum. XVI, 522 pages. 2006.

Vol. 3839: J.-C. Filliâtre, C. Paulin-Mohring, B. Werner (Eds.), Types for Proofs and Programs. VIII, 275 pages. 2006.

Vol. 3838: A. Middeldorp, V. van Oostrom, F. van Raamsdonk, R. de Vrijer (Eds.), Processes, Terms and Cycles: Steps on the Road to Infinity. XVIII, 639 pages. 2005.

Vol. 3837: K. Cho, P. Jacquet (Eds.), Technologies for Advanced Heterogeneous Networks. IX, 307 pages. 2005.

Vol. 3836: J.-M. Pierson (Ed.), Data Management in Grids. X, 143 pages. 2006.

Vol. 3835: G. Sutcliffe, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XIV, 744 pages. 2005. (Sublibrary LNAI).

Vol. 3834: D.G. Feitelson, E. Frachtenberg, L. Rudolph, U. Schwiegelshohn (Eds.), Job Scheduling Strategies for Parallel Processing. VIII, 283 pages. 2005.

Vol. 3833: K.-J. Li, C. Vangenot (Eds.), Web and Wireless Geographical Information Systems. XI, 309 pages. 2005.

Vol. 3832: D. Zhang, A.K. Jain (Eds.), Advances in Biometrics. XX, 796 pages. 2005.

Vol. 3831: J. Wiedermann, G. Tel, J. Pokorný, M. Bieliková, J. Štuller (Eds.), SOFSEM 2006: Theory and Practice of Computer Science. XV, 576 pages. 2006.

Vol. 3829: P. Pettersson, W. Yi (Eds.), Formal Modeling and Analysis of Timed Systems. IX, 305 pages. 2005.

Vol. 3828: X. Deng, Y. Ye (Eds.), Internet and Network Economics. XVII, 1106 pages. 2005.

Vol. 3827: X. Deng, D.-Z. Du (Eds.), Algorithms and Computation. XX, 1190 pages. 2005.

Vol. 3826: B. Benatallah, F. Casati, P. Traverso (Eds.), Service-Oriented Computing - ICSOC 2005. XVIII, 597 pages. 2005.

Vol. 3824: L.T. Yang, M. Amamiya, Z. Liu, M. Guo, F.J. Rammig (Eds.), Embedded and Ubiquitous Computing – EUC 2005. XXIII, 1204 pages. 2005.

Vol. 3823: T. Enokido, L. Yan, B. Xiao, D. Kim, Y. Dai, L.T. Yang (Eds.), Embedded and Ubiquitous Computing – EUC 2005 Workshops. XXXII, 1317 pages. 2005.

Vol. 3822: D. Feng, D. Lin, M. Yung (Eds.), Information Security and Cryptology. XII, 420 pages. 2005.

Vol. 3821: R. Ramanujam, S. Sen (Eds.), FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science. XIV, 566 pages. 2005.

Vol. 3820: L.T. Yang, X. Zhou, W. Zhao, Z. Wu, Y. Zhu, M. Lin (Eds.), Embedded Software and Systems. XXVIII, 779 pages. 2005.

Vol. 3819: P. Van Hentenryck (Ed.), Practical Aspects of Declarative Languages. X, 231 pages. 2005.

Vol. 3818: S. Grumbach, L. Sui, V. Vianu (Eds.), Advances in Computer Science – ASIAN 2005. XIII, 294 pages. 2005.

Vol. 3817: M. Faundez-Zanuy, L. Janer, A. Esposito, A. Satue-Villar, J. Roure, V. Espinosa-Duro (Eds.), Nonlinear Analyses and Algorithms for Speech Processing. XII, 380 pages. 2006. (Sublibrary LNAI).

Vol. 3816: G. Chakraborty (Ed.), Distributed Computing and Internet Technology. XXI, 606 pages. 2005.

Vol. 3815: E.A. Fox, E.J. Neuhold, P. Premsmit, V. Wuwongse (Eds.), Digital Libraries: Implementing Strategies and Sharing Experiences. XVII, 529 pages. 2005.

Vol. 3814: M. Maybury, O. Stock, W. Wahlster (Eds.), Intelligent Technologies for Interactive Entertainment. XV, 342 pages. 2005. (Sublibrary LNAI).

Vol. 3813: R. Molva, G. Tsudik, D. Westhoff (Eds.), Security and Privacy in Ad-hoc and Sensor Networks. VIII, 219 pages. 2005.

Vol. 3810: Y.G. Desmedt, H. Wang, Y. Mu, Y. Li (Eds.), Cryptology and Network Security. XI, 349 pages. 2005.

Vol. 3809: S. Zhang, R. Jarvis (Eds.), AI 2005: Advances in Artificial Intelligence. XXVII, 1344 pages. 2005. (Sublibrary LNAI).

Vol. 3808: C. Bento, A. Cardoso, G. Dias (Eds.), Progress in Artificial Intelligence. XVIII, 704 pages. 2005. (Sublibrary LNAI).

Vol. 3807: M. Dean, Y. Guo, W. Jun, R. Kaschek, S. Krishnaswamy, Z. Pan, Q.Z. Sheng (Eds.), Web Information Systems Engineering – WISE 2005 Workshops. XV, 275 pages. 2005.

Vol. 3806: A.H. H. Ngu, M. Kitsuregawa, E.J. Neuhold, J.-Y. Chung, Q.Z. Sheng (Eds.), Web Information Systems Engineering – WISE 2005. XXI, 771 pages. 2005.

Vol. 3805: G. Subsol (Ed.), Virtual Storytelling. XII, 289 pages. 2005.

Vol. 3804: G. Bebis, R. Boyle, D. Koracin, B. Parvin (Eds.), Advances in Visual Computing. XX, 755 pages. 2005.

Vol. 3803: S. Jajodia, C. Mazumdar (Eds.), Information Systems Security. XI, 342 pages. 2005.

Vol. 3802: Y. Hao, J. Liu, Y.-P. Wang, Y.-m. Cheung, H. Yin, L. Jiao, J. Ma, Y.-C. Jiao (Eds.), Computational Intelligence and Security, Part II. XLII, 1166 pages. 2005. (Sublibrary LNAI).

Vol. 3801: Y. Hao, J. Liu, Y.-P. Wang, Y.-m. Cheung, H. Yin, L. Jiao, J. Ma, Y.-C. Jiao (Eds.), Computational Intelligence and Security, Part I. XLI, 1122 pages. 2005. (Sublibrary LNAI).

Vol. 3799: M. A. Rodríguez, I.F. Cruz, S. Levashkin, M.J. Egenhofer (Eds.), GeoSpatial Semantics. X, 259 pages. 2005.

Vol. 3798: A. Dearle, S. Eisenbach (Eds.), Component Deployment. X, 197 pages. 2005.

Vol. 3797: S. Maitra, C. E. V. Madhavan, R. Venkatesan (Eds.), Progress in Cryptology - INDOCRYPT 2005. XIV, 417 pages. 2005.

Vol. 3796: N.P. Smart (Ed.), Cryptography and Coding. XI, 461 pages. 2005.

Vol. 3795: H. Zhuge, G.C. Fox (Eds.), Grid and Cooperative Computing - GCC 2005. XXI, 1203 pages. 2005.

Vol. 3794: X. Jia, J. Wu, Y. He (Eds.), Mobile Ad-hoc and Sensor Networks. XX, 1136 pages. 2005.

Vol. 3793: T. Conte, N. Navarro, W.-m.W. Hwu, M. Valero, T. Ungerer (Eds.), High Performance Embedded Architectures and Compilers. XIII, 317 pages. 2005.

Vol. 3792: I. Richardson, P. Abrahamsson, R. Messnarz (Eds.), Software Process Improvement. VIII, 215 pages. 2005.

Vol. 3791: A. Adi, S. Stoutenburg, S. Tabet (Eds.), Rules and Rule Markup Languages for the Semantic Web. X, 225 pages. 2005.

Vol. 3790: G. Alonso (Ed.), Middleware 2005. XIII, 443 pages. 2005.

Vol. 3789: A. Gelbukh, Á. de Albornoz, H. Terashima-Marín (Eds.), MICAI 2005: Advances in Artificial Intelligence. XXVI, 1198 pages. 2005. (Sublibrary LNAI).

Vol. 3788: B. Roy (Ed.), Advances in Cryptology - ASIACRYPT 2005. XIV, 703 pages. 2005.

Vol. 3787: D. Kratsch (Ed.), Graph-Theoretic Concepts in Computer Science. XIV, 470 pages. 2005.

Vol. 3785: K.-K. Lau, R. Banach (Eds.), Formal Methods and Software Engineering. XIV, 496 pages. 2005.

Vol. 3784: J. Tao, T. Tan, R.W. Picard (Eds.), Affective Computing and Intelligent Interaction. XIX, 1008 pages. 2005.

Vol. 3783: S. Qing, W. Mao, J. Lopez, G. Wang (Eds.), Information and Communications Security. XIV, 492 pages. 2005.

Vol. 3782: K.-D. Althoff, A. Dengel, R. Bergmann, M. Nick, T.R. Roth-Berghofer (Eds.), Professional Knowledge Management. XXIII, 739 pages. 2005. (Sublibrary LNAI).

Vol. 3781: S.Z. Li, Z. Sun, T. Tan, S. Pankanti, G. Chollet, D. Zhang (Eds.), Advances in Biometric Person Authentication. XI, 250 pages. 2005.

Vol. 3780: K. Yi (Ed.), Programming Languages and Systems. XI, 435 pages. 2005.

Vol. 3779: H. Jin, D. Reed, W. Jiang (Eds.), Network and Parallel Computing. XV, 513 pages. 2005.

Vol. 3778: C. Atkinson, C. Bunse, H.-G. Gross, C. Peper (Eds.), Component-Based Software Development for Embedded Systems. VIII, 345 pages. 2005.

Vol. 3777: O.B. Lupanov, O.M. Kasim-Zade, A.V. Chaskin, K. Steinhöfel (Eds.), Stochastic Algorithms: Foundations and Applications. VIII, 239 pages. 2005.

Vol. 3776: S.K. Pal, S. Bandyopadhyay, S. Biswas (Eds.), Pattern Recognition and Machine Intelligence. XXIV, 808 pages. 2005.

Vol. 3775: J. Schönwälder, J. Serrat (Eds.), Ambient Networks. XIII, 281 pages. 2005.

Vol. 3774: G. Bierman, C. Koch (Eds.), Database Programming Languages. X, 295 pages. 2005.

Vol. 3773: A. Sanfeliu, M.L. Cortés (Eds.), Progress in Pattern Recognition, Image Analysis and Applications. XX, 1094 pages. 2005.

Vol. 3772: M.P. Consens, G. Navarro (Eds.), String Processing and Information Retrieval. XIV, 406 pages. 2005.

Vol. 3771: J.M.T. Romijn, G.P. Smith, J. van de Pol (Eds.), Integrated Formal Methods. XI, 407 pages. 2005.

¥385.00元

Foreword

It was our great pleasure to extend a welcome to all who participated in SERA 2004, the second International Conference on Software Engineering Research, Management and Applications, held at the Omni Hotel, Los Angeles, California, USA. The conference would not have been possible without the cooperation of Seoul National University, Korea, the University of Lübeck, Germany, and Central Michigan University, USA. SERA 2004 was sponsored by the International Association for Computer and Information Science (ACIS).

The conference brought together researchers, practitioners, and advanced graduate students to exchange and share their experiences, new ideas, and research results in all aspects (theory, applications, and tools) of Software Engineering Research and Applications. At this conference, we had keynote speeches by Barry Boehm, C.V. Ramamoorthy, Raymond Yeh, and Con Kenney.

We would like to thank the publicity chairs, the members of our program committees, and everyone else who helped with the conference for their hard work and time dedicated to SERA 2004. We hope that SERA 2004 was enjoyable for all participants.

Barry Boehm

May 2004

Preface

The 2nd ACIS International Conference on Software Engineering – Research, Management and Applications (SERA 2004) was held at the Omni Hotel in Los Angeles, California, during May 5–7, 2004. The conference particularly welcomes contributions at the junction of theory and practice disseminating basic research with immediate impact on practical applications. The SERA conference series has witnessed a short, but successful history:

SERA 2003	San Francisco, California	June 25–27, 2003
SERA 2004	Los Angeles, California	May 5–7, 2004
SERA 2005	Mt. Pleasant, Michigan	August 11–13, 2005

The conference covers a broad range of topics from the field of software engineering including theory, methods, applications, and tools. The conference received 103 submissions from the scientific community in 18 different countries all over the world. Each paper was evaluated by three members of the International Program Committee and additional referees judging the originality, significance, technical contribution, and presentation style. After the completion of the review process 46 papers were selected for presentation at the conference which gave us an acceptance rate of about 45%.

The conference was structured into 14 sessions running in two parallel tracks. The conference sessions covered the following topics: formal methods and tools, data mining and knowledge discovery, requirements engineering, component-based software engineering, object-oriented technology and software architectures, Web engineering and Web-based applications, software reuse and software metrics, agent technology and information engineering, reverse engineering, communication systems and middleware design, XML applications and multimedia computing, parallel and distributed computing, and cost modelling and analysis.

Based on the opinion of the Program Committee and the recommendations of the session chairs, authors of selected papers were invited to submit a substantially revised and extended version for inclusion in the postconference proceedings. The submissions were subject to a second refereeing process. With great pleasure, we finally present 18 papers that were accepted for publication in Springer's LNCS as the post-conference proceedings.

We would like to express our sincere thanks to the Honorary General Chair Barry Boehm and to the Conference Chair Con Kenney. We appreciate the dedication of Barry Boem, C.V. Ramamoorthy, Raymond T. Yeh, and Con Kenney in contributing keynote speeches. We gratefully acknowledge the professional work of the International Program Committee and the subreviewers. We thank the Publicity Chairs for their commitment and, finally, the editorial staff at Springer for the smooth cooperation.

June 2005

Walter Dosch,
Roger Y. Lee,
Chisu Wu

Organizing Committee

Honorary General Chair

Dr. Barry Boehm
Directory, USC Center for Software Engineering
University of Southern California (USA)

Conference Chair

Dr. Con Kenney
Chief IT Enterprise Architect Federal Aviation Administration (USA)

Program Co-chairs

Prof. Roger Lee
Computer Science Department Central Michigan University (USA)

Prof. Chisu Wu Computer Science & Engineering National University (Seoul, Korea)

Prof. Dr. Walter Dosch Institute of Software Technology & Programming Languages University of Lubeck (Germany)

Publicity Chairs

Canada
Sylvanus Ehikioya
University of Manitoba

France
Pascale Minet
INRIA, France

Hong Kong, China
Pak Lok Poon
Hong Kong Polytechnic University

Italy
Susanna Pelagatti
University of Pisa

Japan
Naohiro Ishii
Aichi Institute of Technology

Korea
Haeng-Kon Kim
Catholic University of Daegu

Lebanon
Ramzi Haraty
Lebanese American University

Spain
Ana Moreno
Universidad Politecnica de Madrid

USA
Lawrence Chung University of Texas-Dallas

International Program Committee

Doo-Hwan Bae
(KAIST, Korea)

Jongmoon Baik
(Motorola Labs, USA)

Ken Barker
(Univ. of Calgary, Canada)

Chia-Chu Chiang
(University of Arkansas-Little Rock, USA)

Byoungju Choi
(Ewha Womans University, Korea)

Lawrence Chung
(University of Texas-Dallas, USA)

Jozo Dujmovic
(San Francisco State University, USA)

Sylvanus Ehikioya
(University of Manitoba, Canada)

Paul Garratt
(University of Southampton, UK)

Gonzhu Hu
(Central Michigan University, USA)

Chih Cheng Hung
(Southern Polytechnic State Univ., USA)

Naohiro Ishii
(Aichi Institute of Technology, Japan)

Taewoong Jeon
(Korea University, Korea)

Lianxing Jia
(Univ. of Manchester, UK)

Soo-Dong Kim
(Soongsil University, Korea)

Tai-Hoon Kim
(Korea Information Security Agency, Korea)

Juhnyoung Lee
(IBM T.J. Watson Research Center, USA)

Jun Li
(University of Oregon, USA)

Brian Malloy
(Clemson University, USA)

Pascale Minet
(INRIA, France)

Ana Moreno
(Universidad Politecnica de Madrid, Spain)

Nohpill Park
(Oklahoma State University, USA)

Susanna Pelagatti
(University of Pisa, Italy)

Frantisek Plasil
(Charles University, Czech Republic)

Vinod Prasad
(Bradley University, USA)

David Primeaux
(Virginia Commonwealth University, USA)

Stuart Rubin
(SPAWAR, USA)

Laura Simini
(University of Pisa, Italy)

Gyu-Sang Shin
(ETRI, Korea)

Wei Wang
(University of North Carolina-Chapel Hill, USA)

Cheol Jung Yoo
(Chonbuk National University, Korea)

Cui Zhang
(California State University-Sacramento, USA)

ZhiQuan Zhou
(Swinburne University, Australia)

Table of Contents

Web Engineering and Web-Based Applications

Parallel and Distributed Computing

Software Reuse and Metrics

Object-Oriented Technology and Information Technology

Communications Systems and Networks

Transforming Stream Processing Functions into State Transition Machines

Walter Dosch and Annette Stümpel

Institute of Software Technology and Programming Languages,
University of Lübeck, Germany
http://www.isp.uni-luebeck.de

Abstract. The black-box view of an interactive component in a distributed system concentrates on the input/output behaviour based on communication histories. The glass-box view discloses the component's internal state with inputs effecting an update of the state. The black-box view is modelled by a stream processing function, the glass-box view by a state transition machine. We present a formal method for transforming a stream processing function into a state transition machine with input and output. We introduce states as abstractions of the input history and derive the machine's transition functions using history abstractions. The state refinement is illustrated with three applications, viz. an iterator component, a scan component, and an interactive stack.

1 Introduction

A distributed system consists of a network of components that communicate asynchronously via unidirectional channels. The communication histories are modelled by sequences of messages, called streams. Streams abstract from discrete or continuous time, since they record only the succession of messages. The input/output behaviour of a communicating component is described by a stream processing function [14,15] mapping input histories to output histories.

During the development of a component, the software designer employs different points of view. On the specification level, a component is considered as a black box whose behaviour is determined by the relation between input and output histories. The external view is relevant for the service provided to the environment.

On the implementation level, the designer concentrates on the component's internal state where an input is processed by updating the internal state. The internal view, also called glass-box view, is described by a state transition machine with input and output.

A crucial design step amounts to transforming the specified behaviour of a communicating component into a state-based implementation. In our approach, we conceive machine states as abstractions of the input history. The state stores information about the input history that influences the component's output on future input. In general, there are different abstractions of the input history which lead to state spaces of different granularity [10].

W. Dosch, R.Y. Lee, and C. Wu (Eds.): SERA 2004, LNCS 3647, pp. 1–18, 2005.

This paper presents a formal method, called *state refinement*, for transforming stream processing functions into state transition machines. The transformation is grounded on history abstractions which identify subsets of input histories as the states of the machine. The state refinement preserves the component's input/output behaviour, if we impose two requirements. Upon receiving further input, a history abstraction must be compatible with the state transitions and with the generation of the output stream. The formal method supports a top-down design deriving the state-based implementation from a behavioural specification in a safe way.

The paper is organized as follows. In Section 2 we summarize the basic notions for the functional description of interactive components with communication histories. Section 3 introduces state transition machines with input and output. Section 4 presents the systematic construction of a state transition machine that implements a stream processing function in a correctness preserving way. History abstractions relate input histories to machine states. With their help, the transition functions of the machine can be derived involving the output extension of the stream processing function.

In the subsequent sections, we demonstrate the state refinement for different types of applications. In Section 5, the transformation of an iterator component leads to state transition machines with a trivial state space resulting from the constant history abstraction. Section 6 presents a general implementation scheme for scan components based on the reduce function as a history abstraction. Section 7 discusses the state-based implementation of an interactive stack. The history abstraction leading to a standard implementation results from combining a control state and a data state in a suitable way.

2 Streams and Stream Processing Functions

In this section we briefly summarize the basic notions about streams and stream processing functions to render the paper self-contained. The reader is referred to [24] for a survey and to [25] for a comprehensive treatment. Streams constitute a basic concept for describing different types of interactive systems [19].

2.1 Finite Streams

A stream models the communication history of a channel which is determined by the sequence of data transferred. Untimed streams record only the succession of messages and provide no further information about the timing.

Given a non-empty set $\mathcal{A}$ of data, the set $\mathcal{A}^\star$ of finite *communication histories*, for short *streams*, over $\mathcal{A}$ is the least set with respect to set inclusion defined by the recursion equation

$$\mathcal{A}^\star = \{\langle\rangle\} \cup \mathcal{A} \times \mathcal{A}^\star . \tag{1}$$

A stream is either the *empty stream* $\langle\rangle$ or it is constructed by the prefix operation $\triangleleft : \mathcal{A} \times \mathcal{A}^\star \to \mathcal{A}^\star$ attaching an element to the front of a stream. We denote streams by capital letters and elements of streams by small letters. A stream $X = x_1 \triangleleft x_2 \triangleleft \ldots \triangleleft x_n \triangleleft \langle\rangle$ $(n \geq 0)$ is denoted by $\langle x_1, x_2, \ldots, x_n \rangle$ for short.

The *concatenation* $X \& Y$ of two streams $X = \langle x_1, \ldots, x_k \rangle$ and $Y = \langle y_1, \ldots, y_l \rangle$ over the same set $\mathcal{A}$ of data yields the stream $\langle x_1, \ldots, x_k, y_1, \ldots, y_l \rangle$ of length $k + l$. The concatenation $X \& \langle x \rangle$ appending an element x at the rear of a stream X is abbreviated as $X \triangleright x$.

2.2 Prefix Order

Operational progress is modelled by the prefix order. The longer stream forms an extension of the shorter history, and, vice versa, the shorter stream is an initial segment of the longer stream.

A stream X is called a *prefix* of a stream Y, denoted $X \sqsubseteq Y$, iff there exists a stream R with $X \& R = Y$. The set of finite streams over a data set forms a partial order under the prefix relation with the empty stream as the least element. Monotonic functions on finite streams possess unique continuous extensions to infinite streams [18].

2.3 Stream Processing Functions

The sequence of data passing along a communication channel between two components is captured by the notion of a stream. Thus, a deterministic component which continuously processes data from its input ports and emits data at its output ports can be considered as a function mapping input histories to output histories.

A *stream processing function* $f : \mathcal{A}^\star \to \mathcal{B}^\star$ maps an input stream to an output stream. The input type $\mathcal{A}$ and the output type $\mathcal{B}$ determine the syntactic *interface* of the component.

We require that a stream processing function is *monotonic* with respect to the prefix order:

$$f(X) \sqsubseteq f(X \& Y) \tag{2}$$

This property ensures that a prolongation of the input history leads to an extension of the output history. A communicating component cannot change the past output when receiving future input.

A stream processing function describes the *(input/output) behaviour* of a component.

2.4 Output Extension

A stream processing function characterizes the behaviour of a component on entire input streams. A finer view reveals the causal relationship between single elements in the input stream and corresponding segments of the output stream.

The output extension isolates the effect of a single input on the output stream after processing a prehistory, compare Fig. 1.