

Session Types

Session types are type-theoretic specifications of communication protocols in concurrent or distributed systems. By codifying the structure of communication, they make software more reliable and easier to construct. Over recent decades, the topic has become a large and active research area within the field of programming language theory and implementation.

Written by leading researchers in the field, this is the first text to provide a comprehensive introduction to the key concepts of session types. The thorough theoretical treatment is complemented by examples and exercises, suitable for use in a lecture course or for self-study. It serves as an entry point to the topic for graduate students and researchers.

Simon J. Gay is Professor and Head of Computing Science at the University of Glasgow. He has co-edited two previous books: *Semantic Techniques in Quantum Computation* (2010) and *Behavioural Types: From Theory to Tools* (2017).

Vasco T. Vasconcelos is Professor at the University of Lisbon. He co-authored one of the first papers on session types, for which he received the ETAPS Test of Time Award (2019).

Session Types

SIMON J. GAY

University of Glasgow

VASCO T. VASCONCELOS

University of Lisbon



CAMBRIDGE
UNIVERSITY PRESS



CAMBRIDGE
UNIVERSITY PRESS

Shaftesbury Road, Cambridge CB2 8EA, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre, New Delhi – 110025, India

103 Penang Road, #05–06/07, Visioncrest Commercial, Singapore 238467

Cambridge University Press is part of Cambridge University Press & Assessment,
a department of the University of Cambridge.

We share the University's mission to contribute to society through the pursuit of
education, learning and research at the highest international levels of excellence.

www.cambridge.org

Information on this title: www.cambridge.org/9781316518274

DOI: 10.1017/9781009000062

© Simon J. Gay and Vasco T. Vasconcelos 2025

This publication is in copyright. Subject to statutory exception and to the provisions
of relevant collective licensing agreements, no reproduction of any part may take
place without the written permission of Cambridge University Press & Assessment.

When citing this work, please include a reference to the DOI 10.1017/9781009000062

First published 2025

A catalogue record for this publication is available from the British Library

A Cataloging-in-Publication data record for this book is available from the Library of Congress

ISBN 978-1-316-51827-4 Hardback

Cambridge University Press & Assessment has no responsibility for the persistence
or accuracy of URLs for external or third-party internet websites referred to in this
publication and does not guarantee that any content on such websites is, or will
remain, accurate or appropriate.

Contents

	<i>Preface</i>	<i>page ix</i>
1	Introduction	1
	1.1 From Data Processing to Communication	1
	1.2 From Data Types to Session Types	2
	1.3 Assumptions about Communication	4
	1.4 Session Types in Programming Languages	5
	1.5 The Safety Guarantees of Session Types	5
	1.6 Binary and Multiparty Session Types	6
2	Basic Concepts	7
	2.1 Session Types	7
	2.2 The Pi Calculus with Sessions	9
	2.3 Statics	12
	2.4 Dynamics	17
	2.5 Choice	22
	2.6 Runtime Errors	26
	2.7 Type Safety	30
	2.8 Sending a Channel on Itself	39
	2.9 Linear Types and Affine Types	42
	2.10 Notes	43
3	Infinite Behaviour	45
	3.1 Infinite Session Types	46
	3.2 Recursive Types and Coalgebras	50
	3.3 Equivalence and Duality	56
	3.4 Processes	64
	3.5 Statics	66
	3.6 Dynamics	68
	3.7 Type Safety	69
	3.8 Notes	70
4	Sharing	72
	4.1 Shared Channels	72

vi	Contents	
	4.2 Unrestricted Types	78
	4.3 Statics	80
	4.4 Replicated Input	85
	4.5 Type Safety	89
	4.6 The Unrestricted end Type	96
	4.7 Sending a Channel on Itself	97
	4.8 Notes	98
5	Subtyping	99
	5.1 Subtyping Session Types	99
	5.2 Properties of Subtyping	105
	5.3 Statics	107
	5.4 Type Safety	110
	5.5 Subtyping, Linearity and Sharing	113
	5.6 Notes	116
6	Algorithmic Typing	117
	6.1 From Declarative to Algorithmic Type Checking	117
	6.2 Algorithmic Relations and Predicates on Types	118
	6.3 A Typechecking Algorithm	120
	6.4 Correctness of the Algorithmic Relations on Types	128
	6.5 Correctness of Algorithmic Typechecking	131
	6.6 Notes	140
7	Functional Programming	141
	7.1 Example: Online Shopping	141
	7.2 Types	147
	7.3 Expressions	149
	7.4 Dynamics	152
	7.5 Statics	159
	7.6 Type Safety	164
	7.7 Notes	174
8	Linear Pi Calculus with Values	176
	8.1 Processes	176
	8.2 Dynamics	179
	8.3 Types	182
	8.4 Statics	183
	8.5 Type Safety	185
	8.6 Encoding Session Types	188
	8.7 Notes	196
9	Propositions as Sessions	197
	9.1 Processes	197

	Contents	vii
9.2	Statics	200
9.3	Dynamics	205
9.4	Exponentials and Sharing	207
9.5	Clients in Parallel	214
9.6	Results	215
9.7	The Relationship with Classical Linear Logic	223
9.8	Notes	225
	<i>References</i>	227
	<i>Index</i>	232

Preface

The topic of *session types* concerns the use of programming language type systems to specify and verify the communication behaviour of programs. Just as data types describe the structure of data and constrain the operations that can be performed on it, session types describe the structure of communication and constrain the communication operations that can be performed. A session type is a formal description of a communication protocol, made accessible to programming language tools such as typecheckers, compilers, IDEs and runtime monitors.

The theory of session types was developed by Kohei Honda and his collaborators at Keio University during the 1990s. Honda's paper *Types for Dyadic Interaction* [35] contains the key ideas, but is not always cited as the original paper. It was followed by *An Interaction-Based Language and Its Typing System* [69], co-authored with Kaku Takeuchi and Makoto Kubo, and *Language Primitives and Type Discipline for Structured Communication-Based Programming* [36], co-authored with Vasco Vasconcelos and Makoto Kubo. The latter paper included a range of examples showing that the type system could describe interesting protocols such as FTP. It was published at the ESOP conference, and is noteworthy for being 17 pages long at a time when ESOP papers were restricted to 15 pages. Kohei persuaded the programme committee chair that the paper was important enough to justify extra space.

None of the original papers used the phrase “session types”, although they referred to both sessions and (of course) types. The phrase “session types” first appeared in the paper *Types and Subtypes for Client-Server Interactions* [28] by Simon Gay and Malcolm Hole. Subsequently it became established as the name of the research topic, and at the time of writing, Google Scholar lists more than 500 papers with “session types” in the title. Most conferences on programming languages now include papers on session types every year, often with a whole session devoted to them.

The purpose of this book is to give a clear presentation of the core theory of session types. More precisely, we cover *binary* session types, which deal with interaction between pairs of agents. There is a large literature on *multiparty session types*, introduced by Kohei Honda, Nobuko Yoshida and Marco Carbone in the paper *Asynchronous Multiparty Session Types* [37], which extends the theory to collective protocols among multiple agents. In order to keep the book

to a reasonable size with a clear focus, we have not included multiparty session types. They deserve a book of their own.

After an introductory overview in Chapter 1, Chapter 2 covers the basic theory of binary session types. In this chapter, all behaviour is finite and there is no sharing of channels.

Chapter 3 introduces infinite behaviour in both types and processes, detailing a coinductive foundation for a convenient finite syntax. Chapter 4 adds the possibility of shared channels. At this point we reach a language with similar expressiveness to the one presented in Vasco Vasconcelos' paper *Fundamentals of Session Types* [71], although the presentation is a little different.

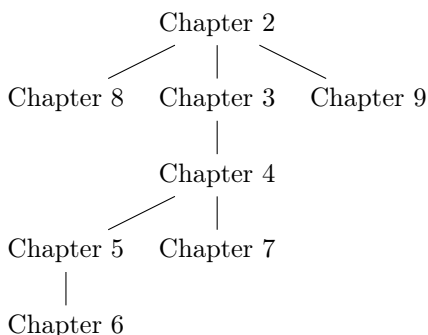
Chapter 5 extends the theory to include subtyping, to support a more flexible approach to judging when two components can communicate safely. Chapter 6 adapts the declarative type system of Chapter 5 into an algorithmic system suitable for implementation as a type checker.

In Chapter 7 we present the ideas from our own paper *Linear Type Theory for Asynchronous Session Types* [32], integrating session types into a multithreaded functional programming language.

Chapter 8 presents work by Ornella Dardha, Elena Giachino and Davide Sangiorgi, showing how session types can be encoded in more primitive type-theoretic constructs. The encoding was originally published in *Session Types Revisited* [23].

Chapter 9 covers a Curry–Howard correspondence between session types and linear logic. This important direction in the field was opened up by Luís Caires and Frank Pfenning in the paper *Session Types as Intuitionistic Linear Propositions* [14] and is still being explored. Our presentation follows Philip Wadler's reformulation of the correspondence in the paper *Propositions as Sessions* [73].

The dependencies between chapters are shown in the diagram below. Even for binary session types, there is a huge amount of material that we have not been able to include. Nevertheless, we hope that by giving a coherent presentation of the core of the theory, we will make the research literature more accessible.



Acknowledgements

Our research on session types has been funded by a variety of projects over the years. Since we started working on this book in 2019, we have been supported by the following projects.

Simon J. Gay:

- From Data Types to Session Types: A Basis for Concurrency and Distribution (UK EPSRC EP/K034413/1).
- Stardust: Session Types for Reliable Distributed Systems (UK EPSRC EP/T014628/1).
- BehAPI: Behavioural Application Program Interfaces (EU H2020 RISE programme, Marie Skłodowska-Curie grant agreement 778233).

Vasco T. Vasconcelos:

- SafeSessions, Safe Concurrent Programming with Session Types (PTDC/CCI-COM/6453/2020).
- The LASIGE Research Unit (UIDB/00408/2020 and UIDP/00408/2020).

We thank the University of Glasgow and the University of Lisbon for sabbatical leave during the academic year 2019/2020.

We have enjoyed learning from a great many colleagues and collaborators, too many to list, both about session types and about more general topics in programming language theory.

We are grateful to these colleagues for reading and commenting on drafts of the book: Ilaria Castellani, Ornela Dardha, Mariangiola Dezani-Ciancaglini, José Espírito Santo, Simon Fowler, Luca Padovani, Jorge Pérez, Diogo Poças, António Ravara, Bernardo Toninho.