CAMBRIDGE

Cambridge University Press
978-0-521-14709-5 - Transitions and Trees: An Introduction to Structural Operational Semantics
Hans Huttel
Frontmatter
More information

Transitions and Trees

An Introduction to Structural Operational Semantics

Structural operational semantics is a simple, yet powerful mathematical theory for describing the behaviour of programs in an implementation-independent manner. This book provides a self-contained introduction to structural operational semantics, featuring semantic definitions using big-step and small-step semantics of many standard programming language constructs, including control structures, structured declarations and objects, parameter mechanisms and procedural abstraction, concurrency, non-determinism and the features of functional programming languages. Along the way, the text introduces and applies the relevant proof techniques, including forms of induction and notions of semantic equivalence (including bisimilarity).

Thoroughly class-tested, this book has evolved from lecture notes used by the author over a 10-year period at Aalborg University to teach undergraduate and graduate students. The result is a thorough introduction that makes the subject clear to students and computing professionals without sacrificing its rigour. No experience with any specific programming language is required.

HANS HÜTTEL is Associate Professor in the Department of Computer Science at Aalborg University, Denmark.

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-14709-5 - Transitions and Trees: An Introduction to Structural Operational Semantics
Hans Huttel
Frontmatter
More information

Transitions and Trees

An Introduction to Structural Operational Semantics

HANS HÜTTEL
Aalborg University, Denmark

CAMBRIDGE
UNIVERSITY PRESS

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-14709-5 - Transitions and Trees: An Introduction to Structural Operational Semantics
Hans Huttel
Frontmatter
More information

CAMBRIDGE UNIVERSITY PRESS
Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore,
São Paulo, Delhi, Dubai, Tokyo

Cambridge University Press
The Edinburgh Building, Cambridge CB2 8RU, UK

Published in the United States of America by Cambridge University Press, New York

www.cambridge.org
Information on this title: www.cambridge.org/9780521197465

© H. Hüttel 2010

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.

First published 2010

Printed in the United Kingdom at the University Press, Cambridge

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

ISBN 978-0-521-19746-5 Hardback
ISBN 978-0-521-14709-5 Paperback

Additional resources for this publication at www.operationalsemantics.net

Cambridge University Press 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.

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-14709-5 - Transitions and Trees: An Introduction to Structural Operational Semantics
Hans Huttel
Frontmatter
More information

Contents

Preface *page* ix
About the illustrations xiii
List of illustrations xiv
List of tables xv

PART I BACKGROUND 1

1 A question of semantics 3
1.1 Semantics is the study of meaning 3
1.2 Examples from the history of programming languages 4
1.3 Different approaches to program semantics 6
1.4 Applications of program semantics 10

2 Mathematical preliminaries 16
2.1 Mathematical induction 16
2.2 Logical notation 17
2.3 Sets 19
2.4 Operations on sets 20
2.5 Relations 21
2.6 Functions 22

PART II FIRST EXAMPLES 25

3 The basic principles 27
3.1 Abstract syntax 27
3.2 Transition systems 30
3.3 Big-step vs. small-step semantics 31
3.4 Operational semantics of arithmetic expressions 31

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-14709-5 - Transitions and Trees: An Introduction to Structural Operational Semantics
Hans Huttel
Frontmatter
More information

3.5 Proving properties 38
3.6 A semantics of Boolean expressions 39
3.7 The elements of an operational semantics 40

4 Basic imperative statements 43
4.1 Program states 43
4.2 A big-step semantics of statements 45
4.3 A small-step semantics of statements in **Bims** 53
4.4 Equivalence of the two semantics 55
4.5 Two important proof techniques 60

PART III LANGUAGE CONSTRUCTS 63

5 Control structures 65
5.1 Some general assumptions 65
5.2 Loop constructs 66
5.3 Semantic equivalence 70
5.4 Abnormal termination 72
5.5 Nondeterminism 73
5.6 Concurrency 76

6 Blocks and procedures (1) 79
6.1 Abstract syntax of **Bip** 79
6.2 The environment–store model 80
6.3 Arithmetic and Boolean expressions 83
6.4 Declarations 84
6.5 Statements 85
6.6 Scope rules 86

7 Parameters 94
7.1 The language **Bump** 94
7.2 Call-by-reference 96
7.3 On recursive and non-recursive procedure calls 97
7.4 Call-by-value 99
7.5 Call-by-name 100
7.6 A comparison of parameter mechanisms 110

8 Concurrent communicating processes 113
8.1 Channel-based communication – **Cab** 113
8.2 Global and local behaviour 114
8.3 Synchronous communication in **Cab** 115
8.4 Other communication models 119

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-14709-5 - Transitions and Trees: An Introduction to Structural Operational Semantics
Hans Huttel
Frontmatter
More information

8.5 Bisimulation equivalence 122
8.6 Channels as data – the π-calculus 123

9 Structured declarations 134
9.1 Records 134
9.2 The language **Bur** 135
9.3 The class-based language **Coat** 142

10 Blocks and procedures (2) 154
10.1 Run-time stacks 154
10.2 Declarations 155
10.3 Statements 155

11 Concurrent object-oriented languages 161
11.1 The language **Cola** 161
11.2 A small-step semantics of concurrent behaviour 163
11.3 Transition systems 163

12 Functional programming languages 171
12.1 What is a functional programming language? 171
12.2 Historical background 173
12.3 The λ-calculus 174
12.4 **Flan** – a simple functional language 177
12.5 Further reading 181

PART IV RELATED TOPICS 183

13 Typed programming languages 185
13.1 Type systems 185
13.2 Typed **Bump** 187
13.3 Typed **Flan** 198
13.4 Type polymorphism and type inference 209

14 An introduction to denotational semantics 211
14.1 Background 211
14.2 λ-Notation 212
14.3 Basic ideas 214
14.4 Denotational semantics of statements 216
14.5 Further reading 220

15 Recursive definitions 222
15.1 A first example 222
15.2 A recursive definition specifies a fixed-point 224
15.3 The fixed-point theorem 225

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-14709-5 - Transitions and Trees: An Introduction to Structural Operational Semantics
Hans Huttel
Frontmatter
More information

15.4 How to apply the fixed-point theorem 231
15.5 Examples of cpos 232
15.6 Examples of continuous functions 236
15.7 Examples of computations of fixed-points 240
15.8 An equivalence result 241
15.9 Other applications 246
15.10 Further reading 248

Appendix A **A big-step semantics of Bip** 249

Appendix B **Implementing semantic definitions in SML** 257

References 264

Index 269

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-14709-5 - Transitions and Trees: An Introduction to Structural Operational Semantics
Hans Huttel
Frontmatter
More information

Preface

About this book

This is a book about structural operational semantics; more precisely it is a book that describes how this approach to semantics can be used to describe common programming language constructs and to reason about the behaviour of programs.

The text grew out of the lecture notes that I have used over a period of more than 10 years in the course *Syntax and semantics* which is taught to all students following the various degree programmes in computer science at Aalborg University. What began as a 10-page set of notes in Danish is now a textbook in English.

The book also includes chapters on related material, namely short introductions to type systems, denotational semantics and the mathematics necessary to understand recursive definitions.

Related work

This work was inspired by lecture notes by Plotkin (1981) (also written in Denmark), where this approach to programming language semantics was first presented.

The topic of structural operational semantics also appears in later books, three of which I will mention here.

Reynolds' book (Reynolds, 1999) is an excellent text that covers some of the same topics as this book but uses denotational and axiomatic semantics as well as structural operational semantics.

The book by Winskel (1993) is another very good textbook that covers many of the same topics as Reynolds' book.

Finally, I should mention Nielson and Nielson (2007), which introduces

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-14709-5 - Transitions and Trees: An Introduction to Structural Operational Semantics
Hans Huttel
Frontmatter
More information

and relates denotational, axiomatic and structural operational semantics and then gives an introduction to how these can be used in connection with static program analysis.

The book that you are now reading differs from the ones just mentioned in three important ways. First, the main topic is exclusively that of structural operational semantics. Second, both Reynolds and Winskel introduce domain theory early on; this book aims at developing the theory of structural operational semantics and making use of it with mathematical prerequisites of a more modest nature. Third, unlike the book by the Nielsons, the focus here is not that of program analysis. Instead, it is on how operational semantics can be used to describe common features of programming languages.

What you need to know in advance

This text is *not* intended as an introduction to programming; if you are a reader expecting this to be the case, you will probably be disappointed! The ideal reader should already have some experience with programming in a high-level imperative programming language such as C, Java or Pascal.

Programming language semantics is a mathematical theory. Therefore, the reader should also have some mathematical maturity. In particular, you should be familiar with basic notions of discrete mathematics – sets, functions and graphs and the proof techniques of proof by induction and proof by contradiction. Chapter 2 gives a short overview of some of this material. There are several good textbooks that you can consult as a supplement; one that I would recommend in particular is Velleman's book (Velleman, 2006).

Ways through the book

The book falls into four parts. The first two parts must be covered in any course for which this book is the main text, since the contents of these first four chapters are necessary to understand the material in the rest of the book. After that, there are the following dependences:

- Chapter 7 and Chapter 9 are independent of each other but both extend the language introduced in Chapter 6,
- Chapter 8 assumes knowledge of the parallel operator introduced in Chapter 5,
- Chapter 10 on small-step semantics for procedures and blocks also assumes knowledge of Chapter 6,
- Chapter 11 assumes knowledge of Chapters 8 and 10 and finally

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-14709-5 - Transitions and Trees: An Introduction to Structural Operational Semantics
Hans Huttel
Frontmatter
More information

- Chapter 15 assumes knowledge of the contents of Chapter 14.

Problems and thoughts

To learn a mathematical subject, one should of course read the text carefully but also learn to apply the content. For this reason, there are quite a few problems scattered throughout the text. As a rule of thumb, a problem will appear at the point in the text where it becomes relevant. I have chosen this approach since I would like you, the reader, to focus on the connection between the problem and the context in which it appears. You can read most of the text without solving the problems but I encourage you to solve as many of them as possible. In some places, I have put problems that are important for understanding the text and they are then marked as such.

I have also introduced mini-problems which I call **A moment's thought**. Here, the idea is to make you think carefully about what you have just read. Do *not* read the text without finding the answers to these mini-problems.

Related resources

The book has its own website, `http://www.operationalsemantics.net`, which has more information, including hints to the mini-problems. The website also holds information about the Danish-language version of the book, *Pilen ved træets rod*, including how to obtain a copy of it.

Acknowledgements

This book grew out of the years I have spent teaching, so, first, I would like to thank the students who have lived with the various incarnations of this text over the years and have made many useful comments that have helped improve and shape its content and form.

Second, I want to thank the people who have inspired me to reflect on the task of teaching mathematical subjects and teaching in general over the years: Jens Friis Jørgensen, Steffen Lauritzen, Finn Verner Jensen, Anette Kolmos, Helle Alrø and Ole Skovsmose.

Third, I would like thank all those who helped me make this book a reality. My thanks go to the people and organizations who have kindly allowed me to use the pictures in Chapter 1 and to David Tranah from Cambridge University Press for his encouragement.

A number of colleagues read parts of the manuscript and provided me with lots of important feedback. Special thanks are due to Denis Bertelsen,

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-14709-5 - Transitions and Trees: An Introduction to Structural Operational Semantics
Hans Huttel
Frontmatter
More information

Morten Dahl, Ulrich Fahrenberg, Morten Kühnrich, Michael Pedersen, Willard Rafnsson and last, but by no means least, Gordon Plotkin.

On an entirely personal level, there are others who also deserve thanks. Over the years, I have come to know many inspiring people through my extracurricular activities in human rights activism and music and, most recently, through the extended family of sisters and brothers that I now have. I am very grateful for knowing you all.

Finally, and most importantly, I want to thank my wife Maria and our daughter Nadia for being in my life. This book is dedicated to you.

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-14709-5 - Transitions and Trees: An Introduction to Structural Operational Semantics
Hans Huttel
Frontmatter
More information

About the illustrations

- The picture of Alfred Tarski on p. 4 is by George M. Bergman and used courtesy of Wikimedia Commons under the GNU Free Documentation License.
- The picture of Dana Scott on p. 7 is used courtesy of Dana Scott.
- The picture of Christopher Strachey on p. 7 is used courtesy of Martin Campbell-Kelly.
- The picture of Gordon Plotkin on p. 8 is used courtesy of Gordon Plotkin.
- The picture of Robin Milner on p. 9 is used courtesy of Robin Milner.
- The picture of Tony Hoare on p. 9 is used courtesy of Tony Hoare.
- The picture of Joseph Goguen on p. 10 is used courtesy of Healfdene Goguen.
- The pictures of Ariane 5 on p. 12 are used courtesy of the ESA/CNES and are ©AFP/Patrick Hertzog, 1996.
- The picture of the Mars Climate Orbiter on p. 14 is used courtesy of NASA.

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-14709-5 - Transitions and Trees: An Introduction to Structural Operational Semantics
Hans Huttel
Frontmatter
More information

Illustrations

1.1	Alfred Tarski	4
1.2	An ALGOL 60 procedure. What is its intended behaviour?	5
1.3	Dana Scott (left) and Christopher Strachey (right)	7
1.4	Gordon Plotkin	8
1.5	Robin Milner	9
1.6	Tony Hoare	9
1.7	Joseph Goguen	10
1.8	The start and end of the maiden voyage of the Ariane 5	12
1.9	The Mars Climate Orbiter	14
3.1	A very small transition system	31
3.2	Derivation tree for a big-step transition for $(\underline{2}+\underline{3})*(\underline{4}+\underline{9}) \rightarrow 65$	35
3.3	Comparison between the derivation trees for the individual steps of a small-step transition sequence and that of a big-step transition	41
6.1	Example of a variable environment and a store	82
6.2	An example **Bip** statement whose behaviour is dependent on the choice of scope rules	88
7.1	A **Bump** statement with recursive calls	98
7.2	Exploiting call-by-name for computing the sum $\sum_{i=1}^{10} i^2$	104
7.3	A **Bump** statement showing where name clashes could occur as a result of an incorrectly defined substitution	108
9.1	A program with nested record declarations	135
9.2	A **Coat** program example	144
13.1	A type system is an overapproximation of safety	198
15.1	Part of the Hasse diagram for $(\mathbb{N}, \leq)$	226
15.2	A Hasse diagram for $(\mathcal{P}(\{1,2,3\}), \subseteq)$	233

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-14709-5 - Transitions and Trees: An Introduction to Structural Operational Semantics
Hans Huttel
Frontmatter
More information

Tables

3.1	Abstract syntax of **Bims**	29
3.2	Big-step transition rules for **Aexp**	33
3.3	Small-step transition rules for **Aexp**	37
3.4	Big-step transition rules for **Bexp**	40
4.1	Big-step transition rules for **Aexp**	45
4.2	Big-step transition rules for **Bexp**	46
4.3	Big-step transition rules for **Stm**	47
4.4	Small-step transition rules for **Stm**	53
5.1	Big-step transition rules for repeat-loops	67
5.2	Big-step transition rules for for-loops	72
5.3	Big-step transition rules for the `or`-statement	74
5.4	Small-step transition rules for the `or`-statement	75
5.5	Small-step semantics of the `par`-statement	77
5.6	An attempt at big-step transition rules for the `par`-statement	77
6.1	Big-step operational semantics of **Aexp** using the environment–store model	83
6.2	Big-step semantics of variable declarations	84
6.3	Big-step transition rules for **Bip** statements (except procedure calls)	87
6.4	Transition rules for procedure declarations (assuming fully dynamic scope rules)	89
6.5	Transition rule for procedure calls (assuming fully dynamic scope rules)	89
6.6	Transition rules for procedure declarations assuming mixed scope rules (dynamic for variables, static for procedures)	90
6.7	Transition rules for procedure calls assuming mixed scope rules (dynamic for variables, static for procedures)	91
6.8	Transition rules for procedure declarations assuming fully static scope rules	92

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-14709-5 - Transitions and Trees: An Introduction to Structural Operational Semantics
Hans Huttel
Frontmatter
More information

6.9 Transition rules for procedure calls assuming fully static scope rules 93
7.1 Rules for declaring procedures with a single parameter assuming static scope rules 96
7.2 Transition rule for calling a call-by-reference procedure 97
7.3 Revised transition rule for procedure calls that allow recursive calls 99
7.4 Transition rules for procedure calls using call-by-value 101
7.5 Transition rules for declaration of call-by-name procedures assuming fully static scope rules 105
7.6 Transition rules for procedure calls using call-by-name 105
7.7 Substitution in statements 111
8.1 Transition rules defining local transitions 116
8.2 Rules defining the capability semantics 117
8.3 Synchronous communication: transition rules for the global level 118
8.4 Asynchronous communication: communication capabilities 120
8.5 Asynchronous communication: the global level 121
8.6 The structural congruence rules 128
8.7 Rules of the reduction semantics of the π-calculus 130
8.8 Rules of the labelled semantics of the π-calculus 132
9.1 Transition rules for generalized variables 137
9.2 Transition rules for arithmetic expressions 138
9.3 Transition rules for variable declarations 139
9.4 Transition rules for procedure declarations 139
9.5 Transition rules for generalized procedure names 140
9.6 Transition rules for record declarations 140
9.7 Transition rules for statements in **Bur** 141
9.8 The semantics of class declarations 146
9.9 The semantics of variable declarations 147
9.10 The semantics of method declarations 147
9.11 The semantics of object declarations 148
9.12 The semantics of object sequences 149
9.13 The semantics of object expressions 150
9.14 Evaluating extended variables (in the semantics of arithmetic expressions) 150
9.15 Important transition rules for statements 151
9.16 Transition rule for programs 152
10.1 Transition rules for statements other than procedure calls 157
10.2 Transition rules for procedure calls assuming static scope rules 159
10.3 Transition rule for procedure calls assuming dynamic scope rules 160
11.1 Transition rules for the local transition system 165
11.2 Transition rules of the labelled transition systems 166
11.3 Transition rules for the global level (1) – initialization 167

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-14709-5 - Transitions and Trees: An Introduction to Structural Operational Semantics
Hans Huttel
Frontmatter
More information

11.4 Transition rules for the global level (2) – the connection between global and local behaviour 168
11.5 Transition rules for the global level (3) – rendezvous 169
12.1 Big-step semantics for **Flan** 179
12.2 Some of the small-step semantics of **Flan** 180
13.1 Big-step operational semantics of **Exp** (arithmetic part) 188
13.2 Big-step transition rules for Bump statements (except procedure calls) 189
13.3 Type rules for **Bump** expressions 191
13.4 Type rules for variable and procedure declarations in **Bump** 191
13.5 Type rules for **Bump** statements 192
13.6 The error predicate for variable declarations 194
13.7 The error predicate for statements 195
13.8 Type rules for **Flan** 200
13.9 The small-step semantics of **Flan** 201
13.10 The type rule for `letrec` 209
A.1 Big-step operational semantics of **Aexp** 252
A.2 Big-step transition rules for $\rightarrow_b$ 253
A.3 Big-step semantics of variable declarations 254
A.4 Transition rules for procedure declarations assuming fully static scope rules 254
A.5 Big-step transition rules for **Bip** statements 256

© in this web service Cambridge University Press www.cambridge.org