

LOGIC FOR CONCURRENCY AND SYNCHRONISATION

TRENDS IN LOGIC
Studia Logica Library

VOLUME 18

Managing Editor

Ryszard Wójcicki, *Institute of Philosophy and Sociology,
Polish Academy of Sciences, Warsaw, Poland*

Editors

Daniele Mundici, *Department of Mathematics “Ulisse Dini”,
University of Florence, Italy*

Ewa Orłowska, *National Institute of Telecommunications,
Warsaw, Poland*

Graham Priest, *Department of Philosophy, University of Queensland,
Brisbane, Australia*

Krister Segerberg, *Department of Philosophy, Uppsala University,
Sweden*

Alasdair Urquhart, *Department of Philosophy, University of Toronto, Canada*

Heinrich Wansing, *Institute of Philosophy, Dresden University of Technology,
Germany*

SCOPE OF THE SERIES

Trends in Logic is a bookseries covering essentially the same area as the journal *Studia Logica* – that is, contemporary formal logic and its applications and relations to other disciplines. These include artificial intelligence, informatics, cognitive science, philosophy of science, and the philosophy of language. However, this list is not exhaustive, moreover, the range of applications, comparisons and sources of inspiration is open and evolves over time.

Volume Editor

Heinrich Wansing

The titles published in this series are listed at the end of this volume.

LOGIC FOR CONCURRENCY AND SYNCHRONISATION

Edited by

RUY J.G.B. DE QUEIROZ

Universidade Federal de Pernambuco, Recife, Brazil

KLUWER ACADEMIC PUBLISHERS

NEW YORK, BOSTON, DORDRECHT, LONDON, MOSCOW

eBook ISBN: 0-306-48088-3
Print ISBN: 1-4020-1270-5

©2003 Kluwer Academic Publishers
New York, Boston, Dordrecht, London, Moscow

Print ©2003 Kluwer Academic Publishers
Dordrecht

All rights reserved

No part of this eBook may be reproduced or transmitted in any form or by any means, electronic, mechanical, recording, or otherwise, without written consent from the Publisher

Created in the United States of America

Visit Kluwer Online at: <http://kluweronline.com>
and Kluwer's eBookstore at: <http://ebooks.kluweronline.com>

Contents

List of Figures	xi
List of Tables	xv
Foreword <i>Johan van Benthem</i>	xvii
Preface	xix
Contributing Authors	xxi
Part I From a Structural Perspective	
1	
Geometry of Deduction via Graphs of Proofs	3
<i>Anjolina Grisi de Oliveira and Ruy J. G. B. de Queiroz</i>	
1 Motivation	4
1.1 Relevance to proof theory: towards the geometry of natural deduction	6
1.2 Relevance to theoretical computer science: complexity and concurrency	8
2 The idea of studying proofs as geometric objects	12
2.1 Representing proofs in a graph	12
2.2 Some results	16
2.2.1 Elementary results on isomorphisms	16
2.2.2 On the genus of a derivation	16
2.2.3 On the relationship between the notions of direct proof, normal proof and proof, and the subformula property	17
3 Proof-nets	17
3.1 Criticisms of natural deduction	18
3.1.1 Classical symmetry	18
3.1.2 Global rules	18
3.1.3 The commutative conversions	19
3.2 Analyzing the sequent calculus	19
3.2.1 The non-determinism of Hauptsatz	19
3.2.2 Redundancies in proofs	20
3.2.3 Non-unicity of proofs	20
3.3 Definition of links	21

3.4	Proof-structures	22
3.4.1	Examples	23
3.5	The soundness criterion	24
3.5.1	No short trip condition	24
3.5.2	Danos–Regnier criterion	28
3.5.3	Asperti criterion	30
3.6	The soundness of the criteria	30
3.7	Cut-elimination	34
4	Logical flow graphs	34
4.1	Motivation	35
4.2	Defining logical flow graphs	36
4.2.1	The sequent calculus	36
4.2.2	Building a logical flow graph	36
4.2.3	A more precise definition	38
4.2.4	Examples	40
4.3	Analyzing the cut-elimination process	40
4.3.1	An example of expansion after cut-elimination	42
4.4	Cycles in proofs	44
4.5	A combinatorial model	45
4.5.1	Optical graphs	45
4.5.2	The duplication operation	46
4.5.3	Connections with cut-elimination	48
5	Multiple-conclusion classical calculi	50
5.1	Lack of symmetry in natural deduction	51
5.2	Kneale’s tables of development	53
5.2.1	Classical and intuitionistic logic	55
5.2.2	Some examples	55
5.2.3	The inadequacy	56
5.3	Shoesmith and Smiley refinements	57
5.3.1	Developments as graph arguments	57
5.3.2	Solving the inadequacy	59
5.3.3	Second solution: multiple junction	62
5.3.4	Third solution: identification of premises and conclusions	65
5.4	Ungar’s system	67
5.4.1	The problem of substitution	68
5.4.2	Solving the problem of substitution	69
6	Finale	70
6.1	Towards a new perspective: N-graphs	71
6.1.1	Proof-graphs	72
6.1.2	Unsound proof-graphs	74
6.1.3	Global soundness criterion	78
6.1.4	The soundness of the criterion	79
6.1.5	Some examples	80
2		
	Chu’s Construction: A Proof-Theoretic Approach	89
	<i>Gianluigi Bellin</i>	
1	Preface	89
2	The trip translation	93
2.1	Languages, intuitionistic and classical MALL	93
2.2	The functorial trip translation for MALL	95
3	Chu’s construction	98
4	Proof-nets, trips and translations	100

4.1	Proof-nets: basic definitions	100
4.2	Trips and linear λ terms	103
4.3	Reduced translations from CMLL ⁻ to IMLL ⁻	103
4.4	Chu's construction in MLL with units	106
3		
	Two Paradigms of Logical Computation in Affine Logic?	111
	<i>Gianluigi Bellin</i>	
1	Introduction	112
2	Sequent calculus of MAL + Mix	119
2.1	Mix and weakening permutations	120
2.2	Weakening reductions and expansions	121
3	Additive mix	123
3.1	Properties of additive mix	123
3.2	Additive mix: reductions and permutations	124
4	Proof-nets for MAL + Mix	126
4.1	Proof-structures and proof-nets	126
4.2	Computation and elimination of irrelevance	128
4.3	Examples and properties of irrelevance elimination	131
4.4	Mix-elimination	134
5	Cut-elimination modulo irrelevance	138
6	Symmetric reductions require Mix	141
4		
	Proof Systems for π -Calculus Logics	145
	<i>Mads Dam</i>	
1	Introduction	146
2	Preliminaries on the π -calculus	148
3	A π - μ -calculus	153
4	Example specifications	159
5	Proof system, modal fragment	162
5.1	Basic judgments	164
5.2	A proof system for the modal fragment	166
5.3	Structural rules	166
5.4	Logical rules	168
5.5	Rules for equality and inequality	168
5.6	Rules for transition modalities	170
5.7	Rules for input/output modalities	174
5.8	Examples	177
6	Soundness and completeness for the modal fragment	179
7	Proof rules for recursive formulas	180
7.1	Rules for fixed point unfolding and approximation	181
7.2	Rule of discharge	183
8	Finite control completeness	188
9	Natural numbers	192
10	Buffers	194
11	Conclusion	196

Part II From a Descriptive Perspective

5

A Tutorial Introduction to Symbolic Model Checking 215

David Déharbe

1	Introduction	215
2	Kripke structures	216
2.1	Definitions	216
2.2	Computation tree	217
3	Temporal logic model checking	218
3.1	Computation tree logic	218
3.1.1	Syntax	219
3.1.2	Semantics	219
3.2	Algorithm	221
3.3	Experimental results and conclusion	225
4	Symbolic model checking	225
4.1	Kripke structures and propositional logic	226
4.1.1	Representing states and transitions	226
4.1.2	State space traversal	226
4.2	Binary decision diagrams	227
4.3	Algorithms	229
4.3.1	Fundamentals of lattices and fixpoints	229
4.3.2	Fixpoint characterization of CTL operators	230
4.3.3	Results and extensions to symbolic model checking	232
5	Conclusion	233

6

Modal Logics for Finite Graphs 239

Mario R. F. Benevides

1	Introduction	239
2	Finite directed graphs	241
2.1	Language and models	241
2.2	Proof theory	242
2.3	Soundness and completeness results	243
3	Finite acyclic directed graphs	244
3.1	Proof theory	244
3.2	Canonical models	245
3.3	Completeness for finite acyclic directed graphs	248
4	Undirected graphs	257
5	Loopless undirected graphs	257
6	Modal definability	258
6.1	Colouring	259
6.2	Hamiltonian graphs	260
6.3	Eulerian graphs	260
6.4	Connectivity	261
6.5	Planarity	261
7	k -Colourable graphs	261
7.1	Completeness for k -colourable graphs	263
8	Conclusions	266

Bisimulation and Language Equivalence

269

Colin Stirling

1 Introduction

269

2 Background

270

3 Caucal's hierarchy

274

4 Richer logics

276

5 Finite model theory

278

List of Figures

1.1	Graph representing a proof	14
1.2	Identity links	21
1.3	Modus ponens	22
1.4	Proof-structure	23
1.5	Proof-structure	23
1.6	Proof-structure	24
1.7	Formula as a box	24
1.8	Axiom link	25
1.9	Terminal formula	26
1.10	Times link switched on “L”	26
1.11	Times link switched on “R”	27
1.12	Par link switched on “L”	27
1.13	Par link switched on “R”	28
1.14	D-R graphs	29
1.15	D-R graph	30
1.16	Proof-net of $\vdash C^\perp, C \otimes (A \wp B), A^\perp \otimes B^\perp$	32
1.17	Proof-nets P_1 and P_2 , respectively	33
1.18	Connecting P_1 and P_2 through a times link	33
1.19	Connecting P_1 and P_2 through a par link	33
1.20	The orientation of the edges in a LFG	37
1.21	Logical flow graph to the formulas p and q	38
1.22	Logical flow graph to the formula A	40
1.23	Logical flow graph to the formulas A and B	41
1.24	An example of expansion	42
1.25	The building block B_{j+1}	43
1.26	Building blocks linked	43
1.27	General scheme of a cycle in a proof	45
1.28	Split of a cycle in a proof	45
1.29	Optical graphs	46

1.30	Duplication operation	48
1.31	An exponential growth	48
1.32	Focussing point followed by a defocussing one in the <i>LFG</i> of a proof	49
1.33	Focussing point followed by a defocussing one	49
1.34	An operation of duplication breaking a cycle	50
1.35	Development of $\neg P/P \rightarrow Q$	54
1.36	Development of $\neg P/P \rightarrow Q$	55
1.37	A fallacy	55
1.38	Development versus Graph argument	58
1.39	Development with a corner	59
1.40	Corner at B	59
1.41	Junction	59
1.42	Non-standard proof from R to Q	60
1.43	Graph arguments Π_1 and Π_2	61
1.44	Graph arguments Π'_1 and Π'_2	61
1.45	Junction of Π'_1 and Π'_2	62
1.46	Invalid arguments	63
1.47	Circuit in an invalid argument	63
1.48	Circuit in a valid argument	64
1.49	Multiple-junction	64
1.50	“key example” with multiple-junction	65
1.51	Multiple-junction	65
1.52	Identification of premises and conclusions	66
1.53	Identification of premises and conclusions: “key example”	66
1.54	The problem of substitution	68
1.55	An attempt to derive $A \vee A \vdash A \wedge A$	69
1.56	Combination on graphs	70
1.57	Links in a proof-graph	72
1.58	Logical links	75
1.59	Structural links	76
1.60	Proof-graphs	77
1.61	Expansion link in proof-graphs	78
1.62	Contraction link in proof-graphs	78
1.63	N-graph of $(A \wedge B) \rightarrow (A \wedge B \wedge C) \vee \neg C$	81
1.64	N-graph of $\vdash ((A \rightarrow B) \rightarrow A) \rightarrow A$	82
2.1	Links	101
2.2	Girard’s trips	101
2.3	Linear λ -terms	103

2.4	Admissible orientations.	104
3.1	Links	126
3.2	Prune	132
3.3	Big prune	133
3.4	Cover	138
3.5	Cut	140
3.6	Example 33	141
3.7	Mix	142
4.1	Buffer with reclaimable cells	151
4.2	Fixed point interference	183
5.1	State transition diagram for Kripke structure <i>ABP sender</i>	217
5.2	Partial state transition diagram of <i>ct(ABP sender)</i>	219
5.3	Illustration of universal CTL operators	220
5.4	Labeling algorithm for CTL formulas	222
5.5	Labeling algorithm for EXf formulas	222
5.6	Labeling algorithm for E[fUg] formulas	223
5.7	Labeling algorithm for EGf formulas	224
5.8	Binary tree for $g \wedge \neg s \wedge \neg w$.	228
5.9	BDD for $g \wedge \neg s \wedge \neg w$.	228
5.10	Algorithm for symbolic model checking CTL formulas	230
5.11	Algorithm for EXf formulas	231
5.12	Algorithm for E[fUg] formulas	231
5.13	Algorithm for EGf formulas	232
5.14	Trace of function call <i>SmcEG(ABPsender, EG¬s ∧ ¬w)</i>	232
6.1	$vR_t w$ and $wR_f v$.	240
6.2	Graph G , where a formula A is true at vertex v .	241
6.3	f-unsatisfied point	249
6.4	Abbreviation for undirected graphs	259
6.5	Square 1,2,3,4 is 4-colourable and triangle a,b,c is not.	259
6.6	Graph 1,2,3,4,5 is Hamiltonian and graph a,b,c,d is not.	260
6.7	Graph 1,2,3,4,5 is Eulerian and graph a,b,c,d is not.	260
6.8	Planar Graph.	262
6.9	Formation rule for the graph presented in Figure 6.8 .	262
6.10	Graph K5 .	262
7.1	Simple transition graphs	270
7.2	Quotiented transition graph	272
7.3	A pushdown automaton	274
7.4	Ingredients of the graphs G and G'	281

List of Tables

1.1	Basic development schemata for propositional fragment	54
1.2	Inadequacy of Kneale's developments	57
1.3	Ungar's system	67
1.4	Circuits in a multiple-conclusion calculus	69
4.1	Transition rules	150
4.2	Formula semantics	155
4.3	Proof rules for parallel composition vs. transition modalities	172
4.4	Monotonicity rules for io-modalities	175
4.5	Rules for parallel composition vs. io-modalities	176

Foreword

The study of information-based actions and processes has been a vibrant interface between logic and computer science for several decades now. Indeed, several natural perspectives come together here. On the one hand, logical systems may be used to describe the dynamics of arbitrary computational processes – as in the many sophisticated process logics available today. But also, key logical notions such as model checking or proof search are themselves informational processes involving agents with goals. The interplay between these descriptive and dynamic aspects shows even in our ordinary language. A word like “proof” denotes both a static ‘certificate’ of truth, and an activity which humans or machines engage in. Increasing our understanding of logics of this sort tells us something about computer science, and about cognitive actions in general.

The individual chapters of this book show the state of the art in current investigations of process calculi such as linear logic, π -calculus, and μ -calculus – with mainly two major paradigms at work, namely, linear logic and modal logic. These techniques are applied to the title themes of concurrency and synchronisation, but there are also many repercussions for topics such as the geometry of proofs, categorial semantics, and logics of graphs. Viewed together, the chapters also offer exciting glimpses of future integration, as the reader moves back and forth through the book. Obvious links include modal logics for proof graphs, labeled deduction merging modal and linear logic, Chu spaces linking proof theory and model theory, and bisimulation-style equivalences as a tool for analyzing proof processes.

The combination of approaches and the pointers for further integration in this book also suggests a grander vision for the field. In classical computation theory, Church’s Thesis provided a unification and driving force. Likewise, modern process theory would benefit immensely from a synthesis bringing together paradigms like modal logic, process algebra, and linear logic – with their currently still separate worlds of bisimulations, proofs, and normalisation. If this Grand Synthesis is ever going to happen, books like this are needed!

JOHAN VAN BENTHEM, ILLC Amsterdam & CSLI Stanford

Preface

The contributions published in this volume arose in the context of the project *Logic for Concurrency and Synchronisation (LOCUS)* which was concerned with the relationship between proof theory (*à la* Curry–Howard-like calculi) and concurrency theory (*à la* π -calculus, μ -calculus), as well as the application of those formalisms to the verification of group-based protocols.

The project also sought to investigate the possibility of defining a unifying methodology (algebraic methods vs. logical methods) for the formalisation of distributed systems, concurrency and synchronisation, using the most recent techniques coming from mathematical logic (in particular, labelled deduction, type theory, and modal logic), proof theory and semantics of concurrent processes.

Four institutions participated in the project: Universidade Federal de Pernambuco (UFPE), Universidade Federal de Alagoas (UFAL), Universidade Federal da Bahia (UFBA), and Universidade Federal do Rio de Janeiro (UFRJ).

Outline

Chapter 1 reviews a collection of recent and less recent work around graph-theoretical tools used in proof theory, leading to some ideas for bringing together the old (e.g., Kneale’s symmetric proof system) and the new (Girard’s graph-theoretic criterion to check soundness of graphs of proof) in order to further enhance the tools for the understanding of natural deduction (ND): the geometry of interaction of ND-proofs, their lack of symmetry and their proof complexity.

In Chapter 2, Bellin argues that the essential interaction between classical and intuitionistic features in the system of linear logic is best described in the language of category theory. The main result is to show that the intuitionistic translations induced by Girard’s trips determine the functor from the free $*$ -autonomous category \mathcal{A} on a set of atoms $\{P, P', \dots\}$ to $\mathcal{C} \times \mathcal{C}^{op}$, where \mathcal{C} is the free monoidal closed category with products and coproducts on the set of atoms $\{P_O, P_I, P'_O, P'_I, \dots\}$ (a pair P_O, P_I in \mathcal{C} for each atom P of \mathcal{A}).

In Chapter 3, Bellin proposes a notion of *symmetric reduction* for a system of proof-nets for *Multiplicative Affine Logic with Mix* (**MAL** + Mix) (namely, multiplicative linear logic with the mix-rule the unrestricted weakening-rule), and proves that such a reduction has the strong normalisation and Church–Rosser properties.

In Chapter 4, Dam studies the problem of verifying general temporal and functional properties of mobile and dynamic process networks, cast in terms of the π -calculus.

In Chapter 5, Déharbe gives a tutorial introduction to CTL model checking and its symbolic BDD-based version implementation.

In Chapter 6, Benevides presents modal logics for four classes of finite graphs: finite directed graphs, finite acyclic directed graphs, finite undirected graphs and finite loopless undirected graphs.

In Chapter 7, Stirling looks at the relationships between bisimulation equivalence and language equivalence.

Acknowledgements. The project *Lógica para Concorrência e Sincronização* (LOCUS) received a 2-year (December 1996 to December 1998) funding from the Brazilian national council for scientific and technological development CNPq (*Conselho Nacional de Desenvolvimento Científico e Tecnológico*), under the *Programa Temático Multi-institucional em Ciência da Computação – Fase III* (ProTeM-CC), grant 680067/95-7.

I should also like to thank the following people who helped to keep the standards high: A. Asperti, M. Benevides, P. Blackburn, A. Carbone, S. Dal-Zilio, R. Goldblatt, O. Grumberg, D. Hughes, M. Kracht, M. Otto, D. Vakarelov.

Last, but not least, I am most grateful to thank Heinrich Wansing (member of the editorial board of the *Trends in Logic* series) for his great care in dealing with the present volume. Let me also thank the Kluwer staff involved, in particular, Jolanda Voogd (Publishing Assistant) and Charles Erkelens (Acquisitions Editor), Humanities Unit – History and Philosophy of Science.

RUY J.G.B. DE QUEIROZ
Recife, February 2003

Contributing Authors

Gianluigi Bellin is a *Ricercatore* at the Facoltà di Scienze of the Università degli Studi di Verona, and a Lecturer at the School of Mathematics of Queen Mary and Westfield College, University of London.

Mario R. F. Benevides is a *Professor Adjunto* at the Instituto de Matemática and Coordenação de Programas de Pós-Graduação em Engenharia de Sistemas (COPPE) of the Universidade Federal do Rio de Janeiro (UFRJ).

Mads Dam is a Researcher at the Swedish Institute of Computer Science (SICS), and *Docent* at Royal Technical University, Stockholm, Sweden.

David Déharbe is a *Professor Adjunto* in the Departamento de Informática e Matemática Aplicada of the Universidade Federal do Rio Grande do Norte (UFRN) in Natal, Brazil.

Anjolina Grisi de Oliveira is a *Professor Adjunto* at the Centro de Informática (CIn) of the Universidade Federal de Pernambuco (UFPE) in Recife, Brazil.

Ruy J.G.B. de Queiroz is a *Professor Adjunto* at the Centro de Informática (CIn) of the Universidade Federal de Pernambuco (UFPE) in Recife, and a *Pesquisador IC* of the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Brazil.

Colin Stirling is a *Professor* at the School of Informatics, University of Edinburgh, Scotland.