

# Contents

<b>Valuation systems and consequence relations</b>	1
1 Introduction	2
1.1 Logics and computer science	2
1.2 Summary	8
2 Valuation systems	9
2.1 Satisfaction	9
2.2 Valuation systems	13
2.3 Modal logic and possible worlds	21
2.4 Predicate languages	25
2.5 Summary	30
3 Consequence relations and entailment relations	30
3.1 Consequence relations	31
3.2 Entailment relations	34
3.3 The systems $C$ and $S4$	35
3.4 Levels of implication	38
3.5 Consequence operator	39
3.6 Summary	40
4 Proof theory and presentations	40
4.1 Hilbert presentations	41
4.2 Natural deduction presentations	45
4.3 Natural deduction in sequent style	51
4.4 Intuitionistic logic	56
4.5 Gentzen sequent calculus for $I$	57
4.6 Gentzen sequent calculus for $C$ and $S4$	62
4.7 Properties of presentations	64
5 Some further topics	66
5.1 Valuation systems for $I$	67
5.2 Maps between logics	69
5.3 Correspondence theory	71
5.4 Consistency	74
<b>Recursion theory</b>	79
0 Introduction	80
0.1 Opening remarks	80
0.2 A taster	81
0.3 Contents of the chapter	84

1	Languages and notions of computability	84
1.1	Data types and coding	85
1.2	The imperative paradigm	87
1.3	The functional paradigm	99
1.4	Recursive functions	109
1.5	Universality	113
2	Computability and non-computability	124
2.1	Non-computability	124
2.2	Computability	127
2.3	Recursive and recursively enumerable sets	129
2.4	The $S$ - $m$ - $n$ theorem and partial evaluation	133
2.5	More undecidable problems	135
2.6	Problem reduction and r.e. completeness	136
3	Inductive definitions	139
3.1	Operators and fixed points	140
3.2	The denotational semantics of the functional language $FL$	149
3.3	Ordinals	156
3.4	The general case	169
4	Recursion theory	172
4.1	Fixed point theorems	173
4.2	Acceptable programming systems	176
4.3	Recursive operators	181
4.4	Inductive definitions and logics	183
	<b>Universal algebra</b>	189
1	Introduction	190
1.1	What is universal algebra?	190
1.2	Universal algebra in mathematics and computer science	191
1.3	Overview of the chapter	192
1.4	Historical notes	192
1.5	Acknowledgements	195
1.6	Prerequisites	196
2	Examples of algebras	196
2.1	Some basic algebras	197
2.2	Some simple constructions	207
2.3	Syntax and semantics of programs	210
2.4	Synchronous concurrent algorithms	215
2.5	Algebras and the modularisation of software	220
3	Algebras and morphisms	220
3.1	Signatures and algebras	220
3.2	Subalgebras	234

3.3	Congruences and quotient algebras	244
3.4	Homomorphisms and isomorphisms	260
3.5	Direct products	275
3.6	Abstract data types	287
4	Constructions	287
4.1	Subdirect products, residual and local properties	288
4.2	Direct and inverse limits	298
4.3	Reduced products and ultraproducts	321
4.4	Local and residual properties and approximation	332
4.5	Remarks on references	336
5	Classes of algebras	337
5.1	Free, initial and final algebras	338
5.2	Equational logic	351
5.3	Equational Horn logic	369
5.4	Specification of abstract data types	392
5.5	Remarks on references	396
6	Further reading	397
6.1	Universal algebra	398
6.2	Model theory	398

**Basic category theory** 413

1	Categories, functors and natural transformations	416
1.1	Types, composition and identities	416
1.2	Categories	418
1.3	Relating functional calculus and category theory	424
1.4	Compositionality is functorial	427
1.5	Natural transformations	433
2	On universal definitions: products, disjoint sums and higher types	437
2.1	Product types	437
2.2	Coproducts	443
2.3	Higher types	446
2.4	Reasoning by universal arguments	451
2.5	Another ‘universal definition’: primitive recursion	456
2.6	The categorical abstract machine	457
3	Universal problems and universal solutions	460
3.1	On observation and abstraction	461
3.2	A more categorical point of view	465
3.3	Universal morphisms	470
3.4	Adjunction	473
3.5	On generation	475
3.6	More examples for separation and generation	481

4	Elements and beyond	492
4.1	Variable elements, variable subsets and representable functors	492
4.2	Yoneda's heritage	498
4.3	Towards an enriched category theory	500
5	Data structures	511
5.1	Subtypes	511
5.2	Limits	516
5.3	Colimits	526
6	Universal constructions	535
6.1	The adjoint functor theorem	536
6.2	Generation as partial evaluation	541
6.3	Left Kan extensions, tensors and coends	545
6.4	Separation by testing	551
6.5	On bimodules and density	553
7	Axiomatizing programming languages	559
7.1	Relating theories of $\lambda$ -calculus	559
7.2	Type equations and recursion	566
7.3	Solving recursive equations	574
8	Algebra categorically	578
8.1	Functorial semantics	578
8.2	Enriched functorial semantics	588
8.3	Monads	593
9	On the categorical interpretation of calculi	603
9.1	Category theory as proof theory	603
9.2	Substitution as predicate transformation	605
9.3	Theories of equality	612
9.4	Type theories	619
10	A sort of conclusion	632
11	Literature	633
11.1	Textbooks	633
11.2	References	634

## Topology 641

1	Observable properties	642
2	Examples of topological spaces	646
2.1	Sierpinski space	646
2.2	Scott Topology	646
2.3	Spaces of maximal elements. Cantor space	647
2.4	Alexandroff topology	647
2.5	Stone spaces	648
2.6	Spectral spaces	649
2.7	The reals	650

3	Alternative formulations of topology	653
	3.1 Closed sets	653
	3.2 Neighbourhoods	654
	3.3 Examples	655
	3.4 Closure operators	656
	3.5 Convergence	660
4	Separation, continuity and sobriety	663
	4.1 Separation conditions	663
	4.2 Continuous functions	665
	4.3 Predicate transformers and sobriety	668
	4.4 Many-valued functions	675
5	Constructions: new spaces from old	681
	5.1 Postscript: effectiveness and representation	690
6	Metric Spaces	698
	6.1 Basic definitions	699
	6.2 Examples	702
	6.3 Completeness	706
	6.4 Topology and metric	713
	6.5 Constructions	719
	6.6 A note on uniformities	724
7	Compactness	727
	7.1 Compactness and finiteness	727
	7.2 Spectral spaces	730
	7.3 Positive and negative information: patch topology	733
	7.4 Hyperspaces	736
	7.5 Tychonoff's theorem	738
	7.6 Locally compact spaces	739
	7.7 Function spaces	744
8	Appendix	751

**Model theory and computer science:**

	<b>An appetizer</b>	763
1	Introduction	764
2	The set theoretic modelling of syntax and semantics	766
	2.1 First order structures	767
	2.2 The choice of the vocabulary	768
	2.3 Logics	769
3	Model theory and computer science	770
	3.1 Computer science	770
	3.2 The birth of model theory	771
	3.3 Definability questions	772
	3.4 Preservation theorems	773
	3.5 Disappointing ultraproducts	774

3.6	Complete theories and elimination of quantifiers	774
3.7	Spectrum problems	775
3.8	Beyond first order logic	777
3.9	The hidden method	778
3.10	0–1 Laws	779
4	Preservation theorems	780
4.1	Horn formulas	780
5	Fast growing functions	782
5.1	Non-provability results in second order arithmetic	782
5.2	Non-provability in complexity theory	782
5.3	Model theory of fast growing functions	783
6	Elimination of quantifiers	784
6.1	Computer aided theorem proving in classical mathematics	785
6.2	Tarski's theorem	787
6.3	Elementary geometry	789
6.4	Other theories with elimination of quantifiers	790
7	Computable logics over finite structures	790
7.1	Computable logics	790
7.2	Computable quantifiers	793
7.3	Computable predicate transformers	794
7.4	<i>L</i> -Reducibility	796
7.5	Logics capturing complexity classes	797
8	Ehrenfeucht–Fraïssé games	798
8.1	The games	798
8.2	Completeness of the game	801
8.3	Second order logic and its sublogics	802
8.4	More non-definability results	802
8.5	The games and pumping lemmas	805
9	Conclusions	805
	<b>Author index</b>	815
	<b>Index</b>	817