
Semantics

András Kornai

Semantics

 Springer

András Kornai
SZTAKI
Institute of Computer Science
Budapest, Hungary

ISBN 978-3-319-65644-1 ISBN 978-3-319-65645-8 (eBook)
<https://doi.org/10.1007/978-3-319-65645-8>

© Springer Nature Switzerland AG 2020

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To Ágnes

Preface

Semantics is the study of meaning. Except for the last chapter, the primary focus of the book is the meaning of linguistic expressions, typically full sentences and longer texts, as opposed to the meaning of [computer programs](#), [mathematical formulas](#), or broader [semiotic](#) concerns. In everyday use ‘semantics’ refers more to the meaning of words, in fact the [Urban Dictionary](#) defines semantics as

The study of discussing the meaning/interpretation of words or groups of words within a certain context; usually in order to win some form of argument. *Now come on, let’s not get bogged down in semantics.*

The field can be equally viewed as a chapter of linguistics, computer science, philosophy, or cognitive science, and to a certain extent the organization of this book will reflect this ambiguity by specifically marking some paragraphs on the margin as **Ling** and **Comp**, and occasionally as **Phil** or **CogSci**. Since no person can be expected to be an expert in all these fields, the prerequisites for each will be discussed separately.

Who should read this book

Our aim is to present the conceptual and formal tools required for building semantic systems capable of understanding text, both for specific tasks such as [information extraction](#) and [question answering](#) and for broad undertakings such as the [semantic web](#). Our goal here is to present the fundamental ideas that working systems rest on, and our textbook is aimed primarily at the computer science or engineering student interested in developing semantic systems. The ideal reader is a *hacker*, ‘a person who delights in having an intimate understanding of the internal workings of a system’. This means not just willingness to try and experiment with things, but also a positive attitude toward research, mathematical modeling in particular. The book is quite demanding in this respect: sentences are long, words of more than three syllables are often encountered, sidebar material and other typographical gimmicks aimed at supporting a rapid lets-find-the-keywords-and-get-it-over-with style of reading are avoided,



and there is an assumption that the reader will take the time to solve the exercises and read up on unfamiliar material as needed.

Comp



The emphasis is on the ideas, but a fair amount (at last count, some 2,500 lines) of code is also available at [GitHub](#), and the book is designed to support a very hands-on study plan that starts with the code and consults the book only as needed. Readers following this plan should be warned that only about a third of what is discussed in the book is actually accompanied by code, they have to contribute the rest themselves. Other, more traditional reading plans are suggested in Section 1.5.

Readers who study semantics because they need code that performs some semantic task will find references within the documentation of the code to the relevant landing sites in the book. Hyperlinks provide two-way crosslinking between the text and the growing body of Python code that implements the main ideas. This material is intended for the experienced software developer only — the book does not provide an introduction to Python — and should not be thought of as providing detailed documentation for the code. Readers are strongly encouraged to contribute to the main repository at <https://github.com/kornai/4lang> under a CC attribution or similar license that is weaker (more permissive) than GPL in that it must permit commercial reuse, and cannot have any viral effect on the rest of the code (GNU LGPL, BSD, and similar licenses are fine).



The computationally oriented reader, ideally a graduate student or advanced undergraduate in computer science/engineering, will find the book self-contained, except for the mathematical prerequisites summarized in Chapter 2. These will be used, and further developed, in the rest of the book with natural, rather than programming languages in mind. Because of this choice of subject, the material has surprisingly little in common with the mathematical logic prerequisites now taken for granted in programming language semantics, where the central attraction is [proofs as programs](#), the remarkable coincidence of two vocabularies, one built by logicians for the analysis of mathematical theorem proving (briefly touched in Section 2.6), and the other by computer scientists for the study of computation. For the trained functional programmer or logician the attraction of this nexus is almost irresistible, but natural language will pull us toward a considerably simpler, *zeroth order* theory, propositional calculus with some modal extensions, where the hard questions center around learnability of the concepts.



Ling

It requires a significant amount of linguistics to build systems that deal with natural language input, and paragraphs marked **Ling** are aimed at the reader who lacks these prerequisites. Within the confines of this volume we could not possibly present the technical machinery we take for granted in linguistics, and these paragraphs are intended as pointers to the linguistic literature, with the primary goal of facilitating self-study. The reader should be warned in advance that our selection of this material is strictly utilitarian, and following the pointers will not lead to a well-rounded picture of linguistic thought, not even of contemporary linguistic semantics. The timely appearance of an excellent volume devoted entirely to compositional semantics, Jacob-

son (2014), made it possible to devote more space to lexical semantics here and still keep this volume to manageable size.

Students of linguistics, especially those with a computational mindset, will likely have an easier time with the material they have to learn, though this will require considerable refactoring, and the occasional bit of unlearning, of the classical formal semantics curriculum. Those students whose computational background is weak should begin with Jurafsky and Martin (2009) or Bird, Klein, and Loper (2009). Very little is assumed from philosophy or cognitive science (see below), but there are notable mathematical prerequisites, discussed in Chapter 2.

The ideas discussed in paragraphs marked **Phil** will most often pertain to the branches of philosophy known as [philosophy of language](#) and [philosophy of science](#). References to philosophical thought will be especially frequent when we need to approach some question from the ground up, as in Chapter 3. Time and again we take the opportunity to point out connections with the ideas of philosophers, but such remarks, except perhaps for Section 9.1, do not amount to a goal-directed introduction to the relevant chapters of philosophy and are quite insufficient for self-study. Rather, they are intended for those readers who are already sensitive to philosophy, with the goal of orienting these readers towards how the positions taken here fit into the larger philosophical debate surrounding these subjects.

We offer new solutions to some well-known philosophical puzzles, in particular [The Heap](#) (sorites) and [Supererogation](#), but these are oriented toward practical goals (since the problems actually come up in system design) and not intended as a fully exhaustive philosophical treatment. In general, students of philosophy will learn a fair amount about some of the big questions: What is meaning? What is knowledge? What is truth? but no systematic philosophical treatment of any of these large subjects is given here. What is offered instead is a highly technical apparatus capable of *modeling* meaning, knowledge, and to a lesser extent, truth, both by computational and by mathematical (more algebraic than logic-based) means.

Neither the computational nor the mathematical prerequisites are normally covered in (philosophical) logic, but we will offer a reading plan geared toward philosophy in Section 1.5. In reading this book, one thing the philosopher has to unlearn, or at least strongly control, is the urge toward a highly technical language. When we analyze *right*, what we provide is an analysis of the everyday notion, not some refined theory of rights. We will of course distinguish *right*₁ ‘dextra’ from *right*₂ ‘bonus’ and *right*₃ ‘ius’, but our definition of the last one is simply ‘law’ and *law* in turn is defined as rule, system, society/2285 HAS, official, ’ ACCEPT, ABOUT can/1246[person[=T0]] (see Section 6.5 for the formal theory of these definitions). In rough paraphrase, laws are systems of rules that societies have, they have official status, people accept them, and laws are about what people can do. But wait, aren’t there unjust laws, ones that people don’t accept? Aren’t there rights that transcend society? We hold that these questions, valuable as they are, cannot be very fruitfully approached through the study of everyday language. To quote from Kornai (2008):

Phil



Since almost all social activity ultimately rests on linguistic communication, there is a great deal of temptation to reduce problems from other fields of inquiry to purely linguistic problems. Instead of understanding schizophrenia, perhaps we should first ponder what the phrase *multiple personality* means. Mathematics already provides a reasonable notion of ‘multiple’, but what is ‘personality’, and how can there be more than one per person? Can a proper understanding of the suffixes *-al* and *-ity* be the key?

CogSci



Originally, understanding systems were built by researchers like [Allen Newell](#) and [Herbert Simon](#) working on [Artificial Intelligence](#) who attempted to model human cognition, or at least borrow design ideas from what was known about the organization of mind at the time. Some of these systems, such as [SOAR](#) or [ACT-R](#), are still in use, while others have been abandoned in favor of newer cognitive architectures like [OpenCog](#). With the emergence of [functional MRI](#) techniques the field has grown enormously, but there is very little in this book that connects to this already vast, and still rapidly growing, literature. The reason, besides the obvious limitations of the author, is that trying to borrow ideas from nature turned out to be a dead end in natural language processing.

Major semantic systems like IBM’s [Watson](#) are not giant electronic brains, in fact they borrow very little from our understanding of biological systems. There may be [neural networks](#) used in various components, but more often than not there are other statistical learners like [support vector machines](#) which do away with the biological metaphor entirely. Within cognitive science there is a renewed effort towards biologically inspired models, led by the [BICA Society](#), but so far these have gained very little traction over the problems central to semantics.

As algorithms increasingly perform in a human-like fashion, the basic architecture dictated by the needs of natural language understanding may be of some interest to the philosopher and the cognitive scientist as well, and Chapters 3 and 9 contain much pertinent material. Needless to say, these disciplines have many broader concerns that are out of scope here, and the philosophy student is strongly advised to consult at least Chapter 16 of Boden (2006). The cognitive science student should of course read the entire two-volume set, and the more recent Gordon and Hobbs (2017), not as prerequisites to this book, but for gaining depth. By providing a rather detailed introduction to the technical machinery of contemporary semantics from the ground up, this book is largely complementary to Boden’s, and covers a fair amount that she could not take into account for the simple reason that it has been published since her book was written.

The book is highly sympathetic to the central claims of [embodied cognition](#), but nevertheless approaches matters from a formal symbol-manipulation direction, because the focus is on building algorithms capable of performing semantic tasks such as schematic inferencing (see Section 7.1), even if this is done at the expense of cognitive realism. The book is for those interested in building flying machines, no matter how birds actually fly, and the only consolation for the cognitive science student is that

some of the technical apparatus, the aerodynamics of understanding so to speak, will be of necessity shared between the two. The reading plan presented for the cognitive scientist in Section 1.5 emphasizes this shared aspect.

Typesetting conventions

The book is primarily designed to be read on a computer. We make heavy use of inline references, typeset in blue, particularly to [Wikipedia](#) (WP), [PlanetMath](#), and the [Stanford Encyclopedia of Philosophy](#) (SEP), especially for concepts and ideas that we feel the reader will already know but may want to refresh. Because following these links greatly improves the reading experience, readers of the paper version are advised to have a cellphone on hand so that they can scan the hyperlinks which are also rendered as QR codes on the margin.

As a novel feature, the book comes with an external index starting at page 293 and also accessible at <http://hlt.bme.hu/semantics/external> that collects a frozen copy of the external references to protect the reader against dead links. A traditional index, with several hundred index terms, is still provided, but the reader is encouraged to search the file if a term is missing there. In some cases, a term may be used informally (with or without an inline reference) before we give a more formal definition. The notational conventions used in these diverse sources may not always coincide with the ones used in the book: for example we use $\langle a, b \rangle$ to denote the [ordered pair](#) that Wikipedia would denote by (a, b) .

The diversity of the technical material presented in the book will be somewhat mitigated by a unified methodological outlook. In philosophy and logic it is quite common to approach the matter normatively, simply condemning those forms of usage that the author sees as ‘illogical’, and devising an ideal language that supports only consistent logical use. To a great extent, the normative outlook also pervades computer science, where one is at liberty to define a [formal language](#) by a [formal grammar](#) and attach compositional semantics to it by means of standard software tools such as [yacc](#). Here we are interested in building a workable semantics for natural language expressions (by ‘workable’ we mean simply that it can be used as the basis of writing computer programs) and take actual usage as the primary empirical testing ground of the theory.

The book contains many exercises, mostly rather simple (under level 30 in the system of Knuth, 1971), but often with surprisingly deep implications, like [Schur’s Lemma](#). In many cases, the solutions can be found quite trivially on the web, or even by just reading a few more pages, but readers interested in developing an active knowledge of this field are strongly advised to attack the problem on their own. The goal of some exercises, marked with a raised $^\circ$, is to check the understanding the reader has developed, and the best reading plan is to solve these problems *immediately* as the reader encounters them in the text, rather than waiting until the end of the section or chapter is reached. Other exercises, marked with a raised \rightarrow , point to material that could not be covered in the book, and often rely on additional knowledge, or presuppositions, that render the answer evasive. Still, the reader is best served by trying their hand at



these both before and after consulting the hints collected after at the end of the volume. Harder exercises are marked with a raised * and will generally be solved in the text not long after they are posed. In later chapters, we will increasingly mark exercises with a raised †, meaning that there is no unique ‘good’ solution, but the reader should experiment with the problem, primarily by building a formal or computational model that exhibits the desired properties. The numbering of definitions and exercises is absolute (includes chapter number), to facilitate cross-referencing and checking for hints at the end, but the numbering of tables, figures, and equations restarts in each chapter.

Linguistic examples are normally given in *italics*, and if a meaning (paraphrase) is provided, this appears in single quotes. Italics are also used for technical terms appearing the first time and for emphasis. The 41ang computational system contains a concept dictionary, which initially had bindings in four languages, representative samples of the major language families spoken in Europe, Germanic (English), Slavic (Polish), Romance (Latin), and Finno-Ugric (Hungarian). Today, bindings exist in over 40 languages (Ács, Pajkossy, and Kornai, 2013). The English printnames of entries in this dictionary, as well as other computationally pertinent material, will be given in typewriter font.

Each chapter ends with a section on further reading. Generally, we recommend those papers and books that presented the idea for the first time. Since many of the issues discussed here have decades, and sometimes centuries, of research behind them, this policy may make the book look far more dated than the opposite policy of citing only the latest research would. We think our policy is justified not just by the need to give proper credit, but also because the early works often provide perspective and insight that later discussions take for granted. (A systematic exception is made for monographs and textbooks that have been republished in revised form: these are cited in their latest edition, since these are often better and always easier to acquire.) However, these works are often available only in paper, and fewer and fewer students or scholars are willing to make a trip to the library. Since this trend is clearly irreversible, we make an effort to provide online references, as clickable links, avoiding password-protected portions of repositories like [Project MUSE](#) and [JSTOR](#) as much as possible.

Acknowledgments

Some of the material presented in the early chapters appeared first in three papers, Kornai (2010, 2010a, 2012), and the help of those who commented on some versions of these papers, in particular Tibor Beke [UMass Lowell](#), Zoltán Szabó [Yale](#), Terry Langendoen [UArizona](#), Donca Steriade [MIT](#), Károly Varasdi [Henrich Heine University](#), is gratefully acknowledged. Chapter 8 is largely based on work first presented in Kornai 2014c and Kornai 2014d. We are grateful to Tibor Beke and Peter Vida [Mannheim](#) for trenchant remarks on these.

The work was partially supported by OTKA grant #77476 and by the European Union and the European Social Fund through project FuturICT.hu (grant # TAMOP-



4.2.2.C-11/1/KONV-2012-0013). Some of the writing was done at Harvard's [IQSS](#), BU's [CS department](#) and [Hariri Institute](#), at the [Algebra department](#) of the Budapest University of Technology and Economics (BUTE), and the [Research Institute for Linguistics](#) of the Hungarian Academy of Sciences (HAS RIL), but the bulk of the work was done at the [Computer Science Institute](#) (HAS CS).

Special thanks to readers of the early versions of this book, who caught many typos and stylistic infelicities, suggested excellent references, exercises, and hints for exercises, and offered penetrating advice on many points evident only to someone with their expertise: Judit Ács [BUTE AUT](#), Eric Bach [Wisconsin-Madison](#), Michael Covington [University of Georgia](#), Gérard Huet [INRIA](#), Paul Kay [Berkeley](#), Marcus Kracht [Bielefeld](#), Márton Makrai (HAS RIL), András Máté [ELTE Logic](#), Imre Orthmayr [ELTE Philosophy](#), Katalin Pajkossy (BUTE), Gábor Recski (BUTE AUT), András Simonyi [PPKE](#), Ferenc Takó [ELTE Philosophy](#), Madeleine Thompson [Empirical](#), and Attila Zséder [Lensa](#). Needless to say, they do not agree with everything in the book, the views expressed here are not those of the funding agencies, and all errors and omissions remain my own.

I would like to single out the Springer proofreader, Douglas Meekison, who managed to transform the sometimes over-excited tone of the manuscript to one of scholarly decorum by a more precise placement of commas and by spreading prepositions to each conjunct, and liberated the printed page from the typographical poverty of `ascii`. My editor, Ronan Nugent, is also to be thanked for making the otherwise often painful publication process an enjoyable experience.

The main developers are Gábor Borbély (BUTE), Márton Makrai (HAS RIL), Dávid Nemeskey, (HAS CS), Gábor Recski (BUTE AUT), and Attila Zséder. The material was taught at BUTE twice, Spring 2010 and Fall 2014, and I am grateful to many of the students, including Kara Greenfield (now with [MIT Lincoln Labs](#)), Sarah Judd (now with [Girls Who Code](#)), and Dániel Vásárhelyi (HAS RIL).

Contents

Preface	vii
1 Introduction	1
1.1 Compositionality and contextuality	1
1.2 Selecting the subject matter	3
1.3 Information content	6
1.4 Plan of the book	7
1.5 Suggested reading plans	10
1.6 Further reading	14
2 Linear Spaces, Boolean Algebras, and First Order Logic	17
2.1 Algebras, Boolean algebras	17
2.2 Universal algebra	20
2.3 Filters, ultrafilters, ultraproducts	22
2.4 The propositional calculus	25
2.5 First order formulas	30
2.6 Proof theory	33
2.7 Multivariate statistics	36
2.8 Further reading	46
3 Prolepsis	49
3.1 Understanding	51
3.2 The minimal theory	54
3.3 Space and time	56
3.4 Psychology	61
3.5 Rules	64
3.6 Regularities	68
3.7 The standard theory	73
3.8 Desiderata	77
3.9 Continuous vector space models	81

3.10 Further reading	87
4 Graphs and Machines	91
4.1 Abstract finite computation	92
4.2 Formal syntax	98
4.3 The smallest machines	106
4.4 Graph and machine operations	109
4.5 Lexemes	111
4.6 Inner syntax	117
4.7 Further reading	123
4.8 Appendix: defining words	124
5 Phenogrammar	127
5.1 Hierarchical structure	128
5.2 Morphology	131
5.3 Syntax	138
5.4 Dependencies	148
5.5 Representing knowledge and meaning	154
5.6 Thoughts in the head	158
5.7 Pragmatics	162
5.8 Valuation	170
5.9 Further reading	174
6 Lexemes	177
6.1 Lexical entries	178
6.2 Concepts	181
6.3 Lexical categories	184
6.4 Word meaning	188
6.5 The formal model	197
6.6 The semantics of lexemes	200
6.7 Further reading	203
7 Models	205
7.1 Schematic inferences	206
7.2 External models	210
7.3 Modalities	214
7.4 Quantification	222
7.5 Further reading	225
8 Embodiment	227
8.1 Perception	228
8.2 Action	235
8.3 Adverbials	242

8.4 Further reading	244
9 The meaning of life	247
9.1 Moral philosophy	248
9.2 The empirical basis of moral law	252
9.3 Metatheoretical considerations	256
9.4 A formal model	259
9.5 Summary and conclusions	262
9.6 Further reading	264
Hints for selected exercises	267
Solutions for selected exercises	269
References	271
Index	289
External index	293