CHAPTER 1

INTRODUCTION

One of the most striking aspects of human language is the complexity of the meanings that it conveys. No other animal possesses a mode of expression that allows it to articulate intricate emotions, describe distant times and places, study molecules and galaxies, or discuss the production of sophisticated tools, weapons and cures. The complex meanings of natural language make it an efficient, general-purpose instrument of human thought and communication. But what are meanings? And how does language convey them?

To illustrate one aspect of the problem, let us consider a phrase in one of Bob Dylan’s famous love songs. The phrase opens the song’s refrain by describing a woman, whose identity is not disclosed. It goes like this:

(1.1) sad-eyed lady of the lowlands, where the sad-eyed prophet says that no man comes

If we want to restate the meaning of this phrase in simpler terms, we can do it as follows:

(1.2) There’s a lady. That lady has sad eyes. She is from the lowlands. Some prophet also has sad eyes. That prophet says “no man comes to the lowlands”.

Without doubt, this way of paraphrasing Dylan’s verse robs it of much of its poetic value. But at the same time it also highlights a remarkable property of meaning in natural language. When we hear a long expression like (1.1), we immediately draw from it all sorts of simple conclusions. This happens even in cases where we miss information that is important for understanding the “true meaning” of what is being said. Dylan’s song only gives vague clues about the identity of the lady. Yet upon hearing the refrain we unfailingly draw
from (1.1) the conclusions in (1.2). The converse is true as well: when Dylan invented his description of the sad-eyed lady, he must have implicitly assumed the statements in (1.2) as part of its meaning. This kind of back-and-forth reasoning occurs whenever we think and converse. When we hear, utter or think of an expression, we instinctively relate it to other phrases that we consider obvious conclusions. Drawing such trivial-looking inferences using our language is one of the abilities that characterize us as linguistic creatures. No other animal has this linguistic ability, and no current technology can accurately mimic it.

Our effortless manipulation of meaning is highly systematic, and relies on an ingrained ability to recognize structure in language. When we hear the phrase in (1.1), we mentally tack its words into short collocations like sad-eyed and the lowlands. Further, short expressions are tacked together into longer expressions such as sad-eyed lady from the lowlands. These syntactic dependencies between words and expressions lead to a complex hierarchical structure. In the case of (1.1), some main elements of this structure are represented below.

\[
\text{(1.3) } \begin{array}{c}
\text{[ [ sad-eyed ] lady ] [ of [ [ the lowlands ], [ where [ [ the [ [ sad-eyed ] prophet ]] [ says [ that [ [ no man ] comes ]]]]]])]
\end{array}
\]

The bracketed expressions in (1.3) represent constituents: sub-parts of the description in (1.1) that act as syntactic units – noun phrases, verb phrases, clauses etc. As the representation in (1.3) illustrates, constituents are often embedded within one another. For instance, the short sentence no man comes is embedded in the verb phrase says that no man comes, which is itself embedded within the sentence the sad-eyed prophet says that no man comes. In total, the expression in (1.1) has no fewer than seven levels of constituents that are embedded within each other. This complexity does not impair our ability to make sense of the description. Furthermore, it is part and parcel of our ability to understand it. In (1.1), the highly organized way in which the constituents are embedded makes it possible for us to immediately grasp Dylan’s complex vision as paraphrased in (1.2). In the case of complicated phrases like (1.1), it is clear that we would not be able to extract even the basic paraphrase in (1.2) if language did not support well-organized hierarchical structures. Furthermore, syntactic hierarchies help us to extract meaning from most other
linguistic expressions, including ones that are much more ordinary than Dylan’s verse.

The subfield of linguistics known as formal semantics studies how linguistic structure helps speakers to manipulate meaning. The word ‘formal’ stresses the centrality of linguistic forms in the enterprise. At the same time, the token ‘formal’ also expresses a motivation to account systematically for language meanings by using precise mathematical methods. Formal semanticists have benefited from the many breakthroughs in logic and computer science, two disciplines that constantly develop new artificial languages and address challenging questions about their meanings and forms. The dazzling achievements that logicians and computer scientists achieved in the twentieth century were based on a rich tradition of research in philosophy of language and the foundations of mathematics. It is only natural that in the 1960s, when semanticists started to systematically address questions about meaning and form in natural language, they turned to these neighboring disciplines in search of guiding principles. As a result, formal semantics relies on the mathematical foundations that were laid in major works on logic, philosophy of language and theoretical computer science.

The mathematical foundations of formal semantics give us precise tools for studying natural languages. Mathematical semantic models help us see what meanings are, and, more importantly, why they can be shared by different expressions. By examining meanings under the powerful microscope of mathematical theories, formal semantics has obtained effective methods for uncovering systematic regularities in the everyday use of language expressions.

The scientific value of this linguistic endeavor is further enhanced by recent developments in other branches of cognitive science that study natural language. In the emerging field of cognitive neuroscience, mathematical principles are becoming increasingly important for harnessing recent advances in brain imaging. As a leading cognitive neuroscientist puts it: “only mathematical theory can explain how the mental reduces to the neural. Neuroscience needs a series of bridging laws […] that connect one domain to the other” (Dehaene 2014, p. 163). These laws are also needed in order to understand how the brain enables the semantic dexterity of language speakers. Mental semantic faculties are profitably described by mathematical laws. Recent works in natural language semantics have supported many of these
laws by statistically analyzing experimental data. As neuroscience brings more experimental data on the workings of the brain, it is becoming increasingly important to connect statistical generalizations about this data with models of our mental semantic abilities.

Similar procedures of mathematical theorizing are equally critical in current work in *artificial intelligence*. Recent advances in statistical machine learning make it possible to exploit formal semantic principles to enhance algorithms and computing technologies. In a recent state-of-the-art review, the authors describe this new direction, stating that “the distinction between logical and statistical approaches is rapidly disappearing with the development of models that can learn the conventional aspects of natural language meaning from corpora and databases” (Liang and Potts 2015, p. 356). In the new domain of computational semantics, mathematical and logical principles of formal semantics are increasingly employed together with statistical algorithms that deal with the parametrization of abstract semantic models by studying distributions of various linguistic phenomena in ordinary language.

Although these recent developments are not the focus of the current book, they do highlight new motivations for using precise principles and techniques in the study of natural language semantics. The achievements of formal semantics have formed a lively area of research, where new ideas, techniques, experimental results and computer systems appear every day. This book introduces you to some of the most important mathematical foundations of this field.

**AIMS AND ORGANIZATION OF THIS BOOK**

The two senses of the word ‘formal’ have a key role in this textbook. The book is a systematic introduction to the study of form and meaning in natural language. At the same time, it capitalizes on the precise mathematical principles and techniques that underlie their analysis. The aim is to help the reader acquire the tools that would allow her to do further semantic work, or engage in interdisciplinary research that relies on principles of formal semantics. Because of that, the book does not attempt to single out any of the current versions of formal semantic theory. Rather, it covers five topics that are of utmost importance to all of them.
Chapter 2 is a general overview of the major goals and techniques in formal semantics. It focuses on the principles of natural language semantics that support meaning relations as in (1) and (2). These semantic relations are called \textit{entailments}. They are described by abstract mathematical \textit{models}, and general principles of compositionality that connect forms with model-theoretical meanings.

Chapter 3 introduces \textit{semantic types} as a means of systematizing the use of models. Typed meanings are derived from simpler ones by a uniform semantic operation of function application. A convenient notation of \textit{lambda-terms} is introduced for describing semantic functions. This notation is illustrated for a couple of modification and coordination phenomena.

Chapter 4 uses the principles and tools of the two previous chapters for treating \textit{quantification}. By focusing on the semantics of noun phrases that involve counting and other statements about quantities, Chapter 4 directly introduces one of the best-known parts of formal semantics: the theory of generalized quantifiers.

Chapter 5 extends the framework of the preceding chapters for treating meaning relations between expressions that appear a certain distance from each other. A principle of \textit{hypothetical reasoning} is added to the system of Chapter 3. This principle works in duality with function application, and complements its operation. The two principles apply within a system of linguistic \textit{signs}, which controls the interactions between forms and meanings.

Chapter 6 treats \textit{intensional expressions}: expressions that refer to attitudes, beliefs or possibilities. Such expressions are treated in semantic models containing entities that represent \textit{possible worlds}. Possible world semantics is introduced as a systematic generalization of the system developed in previous chapters.

Part of the material in Chapters 3 and 6 was covered by early textbooks on “Montague Grammar” (see further reading at the end of this chapter). Here, this material is introduced in a more general setting that takes recent findings into account and capitalizes on the mathematical architecture of type-theoretical grammars. Chapter 4 is unique in being a detailed textbook-level introduction to the central problem of quantification in natural language, which is fully based on the type-theoretical framework of Chapter 3. The treatment of long-distance dependencies in Chapter 5 is the first textbook-level
introduction of a general theoretical configuration known as *Abstract Categorial Grammar*.

At the end of each chapter there are exercises (see below) and references for suggested further reading. Further materials can be found through the website of Edinburgh University Press, at the following link:

edinburghuniversitypress.com/book/9780748640430

**ON THE EXERCISES IN THIS BOOK**

At the end of each chapter you will find some exercises, with model solutions to many of them. Acquiring the ability to solve these exercises constitutes an integral part of studying the material in this book. You will be referred to exercises at various points of the text, and further developments in the book often rely on the exercises in previous chapters. There are two kinds of exercise:

- Technical exercises, which should be solvable by using only the methods explained in the body of the textbook.
- More advanced exercises, which are specified at the beginning of each exercise section. Some of these advanced exercises introduce new notions that were not addressed in the text. These more “notional” advanced exercises are listed in **boldface** at the beginning of the exercises, and are especially recommended among the advanced exercises.

Upon finishing a chapter, and before moving on to the next chapter, it is advisable to make sure that you can correctly solve all of the technical exercises.

**WHO IS THIS BOOK FOR?**

The book is meant for any reader who is interested in human language and its mathematical modeling. For readers whose main interest is linguistic theory, the book serves as an introduction to some of the most useful tools and concepts in formal semantics, with numerous exercises to help grasp them. Readers who are mainly interested in mathematical models of language will find in the book an introduction to natural language semantics that emphasizes its empirical and methodological motivations.
The book is especially suitable for the following audiences:

- general readers with the necessary mathematical background (see below)
- students and teachers of undergraduate linguistics courses on natural language semantics, which put sufficient emphasis on its set-theoretical background (see below)
- students and teachers of relevant undergraduate courses in artificial intelligence, computer science, cognitive science and philosophy
- researchers and advanced students in linguistics

PRESUPPOSED BACKGROUND

To be able to benefit from this book you should have some basic background in naive set theory. At the end of this chapter, you will find some suggestions for further reading, as well as some standard notation, exercises and solutions. By solving the exercises, you will be able to practice some basic set theory at the required level before you start reading. The book does not presuppose any prior knowledge in logic or theoretical linguistics. However, some general familiarity with these disciplines may be useful. Some suggestions for textbooks that introduce this background are given in the suggestions for further reading at the end of this chapter.

FOR THE INSTRUCTOR

The material in this book has been used for teaching undergraduate and graduate courses in linguistics, computer science and artificial intelligence programs. Different kinds of audiences may benefit from different complementary materials. For linguistics students, the most important additions should include more semantic and pragmatic theories of phenomena like anaphora, plurals, events, ellipsis, presupposition or implicature. In most linguistics programs, a short introduction to basic set-theoretical notions would be necessary in order to allow students to grasp the materials in this book (for materials see the further reading section below). For computer science and AI students, additional material on computational semantics may be useful, especially if it is accompanied by programming assignments. The type-theoretical semantics in this book is especially easy to adapt
for programming in strongly typed functional languages like Haskell. Some remarks about recommended literature are made at the end of the further reading section below.

**FURTHER READING**

**Background material on linguistics, set theory and logic:** For a general introduction to linguistics, see Fromkin et al. (2014). For a classical introduction to naive set theory, see Halmos (1960). Linguistics students may find the introduction in Partee et al. (1990, chs.1–3) more accessible. For a useful open-source introduction and exercises, see ST (2015). Two classical textbooks on logic are Suppes (1957); Barker-Plummer et al. (2011).

**On the history of formal semantics:** For a book-length overview, see Partee (2015). For article-length overviews, see Abbott (1999); Partee (1996).

**Other introductions to formal semantics:** Chapters 3 and 6 overlap in some critical aspects with the early textbooks Dowty et al. (1981) and Gamut (1982), which introduced formal semantics as developed in Montague (1973). Zimmermann and Sternefeld (2013) is a friendly introduction to basic topics in formal semantics. For some of the topics covered in the present book, there are also more advanced textbooks that may be consulted. Carpenter (1997) and Jacobson (2014) are detailed introductions to compositional type-theoretical semantics. Jacobson’s book also contains an elaborate linguistic discussion. For introductions to formal semantics as it is often used in generative grammar, see Chierchia and McConnel-Ginet (1990); Heim and Kratzer (1997). For an introduction to formal semantics in the framework of Discourse Representation Theory, see Kamp and Reyle (1993). Readers who are interested in general perspectives on meaning besides formal semantics may consult Elbourne (2011); Saeed (1997).

**For the instructor:** On further important topics in formal semantics that are not covered in this textbook, see Chapter 7. For a textbook that uses the Haskell programming language to illustrate some of the core problems in formal semantics, see Van Eijck and Unger (2010).
INTRODUCTION

Concepts and notation from set theory
\( x \in A \)  
\( x \) is an element of the set \( A = x \) is a member of \( A \)

\( x \not\in A \)  
\( x \) is not an element of \( A \)

\( \emptyset \)  
the empty set \( = \) the set that has no members

\( A \subseteq B \)  
the set \( A \) is a subset of the set \( B = B \) is a superset of \( A = every \) element of \( A \) is an element of \( B \)

\( A \nsubseteq B \)  
\( A \) is not a subset of \( B \)

\( \wp(A) \)  
the powerset of \( A = \) the set of all subsets of \( A \). Example:
\( \wp(\{a, b\}) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\} \)

\( A \cap B \)  
the intersection of \( A \) and \( B = \) the set of elements that are in both \( A \) and \( B \)

\( A \cup B \)  
the union of \( A \) and \( B = \) the set of elements that are in \( A \) or \( B \) (or both)

\( A - B \)  
the difference between \( A \) and \( B = \) the set of elements in \( A \) that are not in \( B \)

\( \overline{A} \)  
the complement of \( A \) (in \( E = E - A \), where \( E \) is a given superset of \( A \))

\(|A|\)  
the cardinality of \( A = \) for finite sets: the number of elements in \( A \)

\( \{x : x \in A \} : S\)  
the set of elements in \( A \) s.t. the statement \( S \) holds

Example: \( \{x \in \{a, b\} : x \in \{b, c\}\} = \{a, b\} \cap \{b, c\} = \{b\} \)

\( \{A \subseteq B : S\} \)  
the set of subsets of \( B \) s.t. the statement \( S \) holds. Example:
\( \{A \subseteq \{a, b\} : |A| = 1\} = \{\{a\}, \{b\}\} \)

\( \langle x, y \rangle \)  
an ordered pair of items \( x \) and \( y \)

\( A \times B \)  
the cartesian product of \( A \) and \( B = \) the set of ordered pairs \( \langle x, y \rangle \) s.t. \( x \in A \) and \( y \in B \)

Example: \( \{a, b\} \times \{1, 2\} = \{(a, 1), (a, 2), (b, 1), (b, 2)\} \)

A binary relation between \( A \) and \( B \) is a subset of the cartesian product \( A \times B \).

A function \( f \) from \( A \) to \( B \) is a binary relation between \( A \) and \( B \) that satisfies: for every \( x \in A \), there is a unique \( y \in B \) s.t. \( \langle x, y \rangle \in f \). If \( f \) is a function where \( \langle x, y \rangle \in f \), we say that \( f \) maps \( x \) to \( y \), and write \( f : x \mapsto y \) or \( f(x) = y \).

Example: the binary relation \( f = \{(a, 1), (b, 2)\} \) is a function from \( \{a, b\} \) to \( \{1, 2\} \), which is equivalently specified \( [a \mapsto 1, b \mapsto 2] \) or by indicating that \( f(a) = 1 \) and \( f(b) = 2 \).
$B^A$ is the set of functions from $A$ to $B$.

Example: $\{1, 2\}^\{a,b\} = \text{the functions from } \{a, b\} \text{ to } \{1, 2\}$

$= \{[a \mapsto 1, b \mapsto 1], [a \mapsto 1, b \mapsto 2], [a \mapsto 2, b \mapsto 1], [a \mapsto 2, b \mapsto 2]\}$

EXERCISES

1. Which of the following statements are true?
   (i) $a \in \{a, b\}$   (ii) $\{a\} \in \{a, b\}$   (iii) $\{a\} \subseteq \{a, b\}$
   (iv) $a \subseteq \{a, b\}$   (v) $\{a\} \cap \{a\}$   (vi) $\{a\} \subseteq \{a\}$
   (vii) $\{\{a, b, c\}\} \subseteq \wp(\{a, b, c\})$   (viii) $\{\{a, b, c\}\} \in \wp(\{a, b, c\})$
   (ix) $\emptyset \in \{\{a\}, \{b\}, \{c\}\}$

2. Write down explicitly the following sets by enumerating their members, e.g. $\wp(\{a\}) = \emptyset, \{a\}$.
   (i) $\wp(\{a, b, c\})$   (ii) $\{a\} \cap \wp(\{a\})$   (iii) $\{\{a\}\} \cap \wp(\{a, b\})$
   (iv) $\wp(\{a, b\}) \cap \wp(\{b, c\})$   (v) $(\wp(\{a\}) \cup \wp(\{b\})) \cap \wp(\{a, b\})$
   (vi) $\wp(\emptyset)$

3. Write down explicitly the following sets by enumerating their members.
   (i) $(\{a, b\} \times \{c\}) \cap (\{a\} \times \{b, c\})$   (ii) $\wp(\emptyset) \times \wp(\{a, b\})$
   (iii) $\wp(\{a, b\}) \times \wp(\{b, c\})$

4. Which of the following binary relations are functions from $\{a, b\}$ to $\{1, 2\}$?
   (i) $\{(a, 1), (a, 1), (b, 2)\}$   (ii) $\{(a, 1), (a, 2)\}$
   (iii) $\{(a, 1), (b, 1)\}$   (iv) $\{(a, 1), (a, 2), (b, 1)\}$

5. How many binary relations are there between $\{a, b\}$ and $\{1, 2\}$?
   How many of them are functions?

6. Write down the functions in $\{\text{no, yes}\}^{\{a,b,c\}}$. For each such function show a member of the powerset $\wp(\{a, b, c\})$ that intuitively corresponds to it.

7. Write down the functions in $\{a, b, c\}^{\{\text{left}, \text{right}\}}$. For each such function show a member of the cartesian product $\{a, b, c\} \times \{a, b, c\}$ that intuitively corresponds to it.

8. Write down explicitly the following sets of functions:
   (i) $\wp(\{a\})^{\wp(\{b\})}$   (ii) $\{1, 2\}^{\{a,b\} \times \{c\}}$   (iii) $\{(1, 2)^{\{c\}}\}^{\{a,b\}}$

9. Consider the following function $f$ in $\{1, 2\}^{\{a,b\} \times \{c,d\}}$:
   $[\langle a, c \rangle \mapsto 1, \langle a, d \rangle \mapsto 1, \langle b, c \rangle \mapsto 2, \langle b, d \rangle \mapsto 1]$.
   Write down the function $g$ in $\{(1, 2)^{\{c,d\}}\}^{\{a,b\}}$ that satisfies for every $x$ in $\{a, b\}$, for every $y$ in $\{c, d\}$: $g(x)(y) = f(\langle x, y \rangle)$.
10. Write down explicitly the members of the following sets:
   (i) \( \{ f \in [a, b]^{[b, c]} : f(b) = b \} \)
   (ii) \( \{ A \subseteq \{ a, b, c, d \} : |A| \geq 3 \} \)
   (iii) \( \{(x, y) \in [a, b, c] \times \{ b, c, d \} : x \neq y \} \)

**SOLUTIONS TO EXERCISES**

1. i, iii, v, vi, vii, x
2. (i) \( \emptyset, \{ a \}, \{ b \}, \{ c \}, \{ a, b \}, \{ a, c \}, \{ b, c \}, \{ a, b, c \} \) (ii) \( \emptyset \)
   (iii) \( \{ \{ a \} \} \) (iv) \( \emptyset, \{ b \} \) (v) \( \emptyset, \{ a \}, \{ b \} \) (vi) \( \emptyset, \{ \emptyset \} \)
3. (i) \( \{ (a, c) \} \) (ii) \( \{ (\emptyset, \emptyset), (\emptyset, \{ a \}), (\emptyset, \{ b \}), (\emptyset, \{ a, b \}), (\{ \emptyset \}, \emptyset), (\{ \emptyset \}, \{ a \}), (\{ \emptyset \}, \{ b \}), (\{ \emptyset \}, \{ a, b \}) \} \) (iii) \( \{ (b, c), \{ a, c \}, \{ b, c \} \} \)
4. ii, iv
5. 16; 4
6. \( \{ a \mapsto \text{no}, \ b \mapsto \text{no}, \ c \mapsto \text{no} \} : \emptyset \)
   \( \{ a \mapsto \text{yes}, \ b \mapsto \text{no}, \ c \mapsto \text{no} \} : \{ a \} \)
   \( \{ a \mapsto \text{no}, \ b \mapsto \text{yes}, \ c \mapsto \text{no} \} : \{ b \} \)
   \( \{ a \mapsto \text{no}, \ b \mapsto \text{no}, \ c \mapsto \text{yes} \} : \{ c \} \)
   \( \{ a \mapsto \text{yes}, \ b \mapsto \text{yes}, \ c \mapsto \text{no} \} : \{ a, b \} \)
   \( \{ a \mapsto \text{yes}, \ b \mapsto \text{no}, \ c \mapsto \text{yes} \} : \{ a, c \} \)
   \( \{ a \mapsto \text{no}, \ b \mapsto \text{yes}, \ c \mapsto \text{yes} \} : \{ b, c \} \)
   \( \{ a \mapsto \text{yes}, \ b \mapsto \text{yes}, \ c \mapsto \text{yes} \} : \{ a, b, c \} \)
7. \( \{ \text{left} \mapsto a, \ \text{right} \mapsto a \} : \langle a, a \rangle \)
   \( \{ \text{left} \mapsto a, \ \text{right} \mapsto b \} : \langle a, b \rangle \)
   \( \{ \text{left} \mapsto b, \ \text{right} \mapsto c \} : \langle a, c \rangle \)
   \( \{ \text{left} \mapsto b, \ \text{right} \mapsto a \} : \langle b, a \rangle \)
   \( \{ \text{left} \mapsto b, \ \text{right} \mapsto b \} : \langle b, b \rangle \)
   \( \{ \text{left} \mapsto c, \ \text{right} \mapsto c \} : \langle c, c \rangle \)
   \( \{ \text{left} \mapsto c, \ \text{right} \mapsto a \} : \langle c, a \rangle \)
   \( \{ \text{left} \mapsto c, \ \text{right} \mapsto b \} : \langle c, b \rangle \)
8. (i) \( \{ [\emptyset \mapsto \emptyset, \ {b} \mapsto \emptyset], \ [\emptyset \mapsto \emptyset, \ {b} \mapsto \{ a \}], \ [\emptyset \mapsto \{ a \}, \ {b} \mapsto \emptyset], \ [\emptyset \mapsto \{ a \}, \ {b} \mapsto \{ a \}] \} \)
   (ii) \( \{ [(a, c) \mapsto 1, \ (b, c) \mapsto 1], \ [(a, c) \mapsto 1, \ (b, c) \mapsto 2], \ [(a, c) \mapsto 2, \ (b, c) \mapsto 1], \ [(a, c) \mapsto 2, \ (b, c) \mapsto 2] \} \)
   (iii) \( \{ [(a \mapsto c \mapsto 1), \ (b \mapsto c \mapsto 1)], \ [(a \mapsto c \mapsto 1), \ (b \mapsto c \mapsto 2)], \ [(a \mapsto c \mapsto 2), \ (b \mapsto c \mapsto 1)], \ [(a \mapsto c \mapsto 2), \ (b \mapsto c \mapsto 2)] \} \)
9. \( \{ a \mapsto c \mapsto 1, \ d \mapsto 1 \}, \ b \mapsto c \mapsto 2, \ d \mapsto 1 \} \)
10. (i) \( \{ b \mapsto b, \ c \mapsto a \}, \ b \mapsto b, \ c \mapsto b \)
    (ii) \( \{ b, c, d \}, \ {a, c, d}, \ {a, b, d}, \ {a, b, c}, \ {a, b, c, d} \)
    (iii) \( \{ a, b \}, \ {a, c}, \ {a, d}, \ {b, c}, \ {b, d}, \ {c, b}, \ {c, d} \)