

The Mathematics of GIS

Wolfgang Kainz

Contents

0. Preface	XI
1. The Structure of Mathematics	1
1.1 Brief History of Mathematics	2
1.2 Sub-disciplines of Mathematics	2
2. Propositional Logic	5
2.1 Assertion and Proposition	6
2.2 Logical Operators	7
2.3 Types of Propositional Forms	9
2.4 Applications in GIS	11
2.5 Exercises	11
3. Predicate Logic	13
3.1 Predicates	14
3.2 Quantifiers	15
3.3 Quantifiers and Logical Operators	16
3.4 Compact Notation	17
3.5 Applications in GIS	18
3.6 Exercises	18
4. Logical Inference	19
4.1 Logical Arguments	20
4.2 Proving Arguments Valid in Propositional Logic	20
4.2.1 Proving Arguments Valid with Truth Tables	21
4.2.2 Proving Arguments Valid with Rules of Inference	21
4.3 Proving Arguments Valid in Predicate Logic	22
4.4 Applications in GIS	23
4.5 Exercises	23
5. Set Theory	25
5.1 Sets and Elements	26
5.2 Relations between Sets	27
5.3 Operations on Sets	27
5.4 Applications in GIS	29
5.5 Exercises	30
6. Relations and Functions	31
6.1 Cartesian Product	32
6.2 Binary Relations	32
6.2.1 Relations and Predicates	33
6.2.2 Graphic Representation of Binary Relations	33

6.2.3 <i>Special Properties of Relations</i>	33
6.2.3.1 <i>Equivalence Relation</i>	34
6.2.3.2 <i>Order Relation</i>	35
6.2.4 <i>Composition of Relations</i>	35
6.3 <i>Functions</i>	36
6.3.1 <i>Composition of Functions</i>	37
6.3.2 <i>Classes of Functions</i>	37
6.4 <i>Applications in GIS</i>	40
6.5 <i>Exercises</i>	42
7. <i>Coordinate Systems and Transformations</i>	43
7.1 <i>Coordinate Systems</i>	44
7.1.1 <i>Cartesian Coordinate Systems</i>	44
7.1.2 <i>Polar Coordinate Systems</i>	45
7.1.3 <i>Transformations between Cartesian and Polar Coordinate Systems</i>	45
7.1.4 <i>Geographic Coordinate System</i>	47
7.2 <i>Vectors and Matrices</i>	47
7.2.1 <i>Vectors</i>	47
7.2.2 <i>Matrices</i>	51
7.3 <i>Transformations</i>	52
7.3.1 <i>Geometric Transformations</i>	52
7.3.1.1 <i>Translation</i>	52
7.3.1.2 <i>Rotation</i>	53
7.3.1.3 <i>Scaling</i>	53
7.3.2 <i>Combination of Transformations</i>	54
7.3.3 <i>Homogeneous Coordinates</i>	55
7.3.4 <i>Transformation between Coordinate Systems</i>	56
7.4 <i>Applications in GIS</i>	57
7.5 <i>Exercises</i>	58
8. <i>Algebraic Structures</i>	59
8.1 <i>Components of an Algebra</i>	60
8.1.1 <i>Signature and Variety</i>	60
8.1.2 <i>Identity and Zero Elements</i>	61
8.2 <i>Varieties of Algebras</i>	61
8.2.1 <i>Group</i>	62
8.2.2 <i>Field</i>	62
8.2.3 <i>Boolean Algebra</i>	63
8.2.4 <i>Vector Space</i>	63
8.3 <i>Homomorphism</i>	64
8.4 <i>Applications in GIS</i>	65
8.5 <i>Exercises</i>	66
9. <i>Topology</i>	67
9.1 <i>Topological Spaces</i>	68
9.1.1 <i>Metric Spaces and Neighborhoods</i>	68
9.1.2 <i>Topology and Open Sets</i>	69
9.1.3 <i>Continuous Functions and Homeomorphisms</i>	71
9.1.4 <i>Alternate Definition of a Topological Space</i>	72
9.2 <i>Base, Interior, Closure, Boundary, and Exterior</i>	74
9.3 <i>Classification of Topological Spaces</i>	76
9.3.1 <i>Separation Axioms</i>	76

9.3.2 Compactness.....	78
9.3.3 Size.....	80
9.3.4 Connectedness.....	81
9.4 Simplicial Complexes and Cell Complexes.....	82
9.4.1 Simplexes and Polyhedra	82
9.4.2 Cells and Cell Complexes	83
9.5 Applications in GIS.....	86
9.5.1 Spatial Data Sets.....	87
9.5.2 Topological Transformations.....	88
9.5.3 Topological Consistency	88
9.5.4 Spatial Relations	90
9.6 Exercises	91
10. Ordered Sets	93
10.1 Posets	94
10.1.1 Order Diagrams.....	94
10.1.2 Upper and Lower Bounds.....	95
10.2 Lattices.....	96
10.3 Normal Completion	97
10.3.1 Special Elements	98
10.3.2 Normal Completion Algorithm.....	99
10.4 Application in GIS	101
10.5 Exercises	101
11. Graph Theory	103
11.1 Introducing Graphs	104
11.1.1 Basic Concepts.....	105
11.1.2 Path, Circuit, Connectivity.....	106
11.2 Important Classes of Graphs.....	107
11.2.1 Directed Graph.....	107
11.2.2 Planar Graph.....	107
11.3 Representation of Graphs	108
11.4 Eulerian and Hamiltonian Tours, Shortest Path Problem	110
11.4.1 Eulerian Graphs.....	110
11.4.2 Hamiltonian Tours.....	111
11.4.3 Shortest Path Problem.....	111
11.5 Applications in GIS.....	111
11.6 Exercises	112
12. Fuzzy Logic and GIS	113
12.1 Fuzziness	114
12.1.1 Motivation.....	114
12.1.2 Fuzziness versus Probability.....	114
12.2 Crisp Sets and Fuzzy Sets.....	115
12.3 Membership Functions	116
12.4 Operations on Fuzzy Sets	118
12.5 Alpha-Cuts.....	122
12.6 Linguistic Variables and Hedges.....	122
12.7 Fuzzy Inference	123

12.7.1 MAMDANI's Direct Method.....	124
12.7.2 Simplified Method.....	127
12.8 Applications in GIS.....	129
12.8.1 Objective.....	129
12.8.2 Fuzzy Concepts.....	129
12.8.3 Software Approach.....	129
12.8.3.1 ArcInfo GRID.....	130
12.8.3.2 ArcMap Spatial Analyst.....	130
12.8.3.3 ArcView 3.x.....	130
12.8.3.4 ArcGIS 9 Script.....	131
12.8.4 Result.....	132
12.9 Exercises.....	132
13. Spatial Modeling.....	135
13.1 Real World Phenomena And Their Abstractions.....	136
13.1.1 Spatial Data And Information.....	136
13.2 Concepts Of Space And Time.....	137
13.2.1 Pre-Newtonian Concepts Of Space And Time.....	137
13.2.2 Classical Concepts Of Space And Time.....	138
13.2.3 Contemporary Concepts Of Space And Time.....	139
13.2.4 Concepts Of Space And Time In Spatial Information Systems.....	139
13.3 The Real World And Its Models.....	140
13.3.1 Maps.....	140
13.3.2 Databases.....	141
13.3.3 Space And Time In Real World Models.....	142
13.4 Real World Models And Their Representation.....	142
13.4.1 Database Design.....	143
13.4.2 Spatial Data Models.....	144
13.4.2.1 Field-based Models.....	145
13.4.2.2 Object-based Models.....	146
13.4.3 Spatiotemporal Data Models.....	146
13.4.3.1 Space-Time Cube Model.....	147
13.4.3.2 Snapshot Model.....	147
13.4.3.3 Space-Time Composite Model.....	147
13.4.3.4 Event-based Model.....	148
13.4.3.5 Spatiotemporal Object Model.....	148
13.5 Summary.....	148
14. Solutions of Exercises.....	149
15. References and Bibliography.....	153
16. Index.....	155

List of Figures

Figure 1. Sub-disciplines of mathematics and their relationships	3
Figure 2. Raster Calculator with logical connectors	11
Figure 3. VENN diagram.....	26
Figure 4. Non-commutativity of the Cartesian product	32
Figure 5. Sample relations.....	34
Figure 6. Composition of relations	36
Figure 7. Functions and relations	37
Figure 8. Topological relations	40
Figure 9. Spatial relations derived from topological invariants.....	41
Figure 10. Map projections with singularities.....	41
Figure 11. Cartesian coordinate system in the plane.....	44
Figure 12. Cartesian coordinate system in 3-D space.....	44
Figure 13. Polar coordinate system in the plane	45
Figure 14. Spherical coordinate system.....	45
Figure 15. Conversion between Cartesian and polar coordinates in the plane	45
Figure 16. Conversion between Cartesian coordinates and spherical coordinates	46
Figure 17. Geographic coordinate system.....	47
Figure 18. Point vector	48
Figure 19. Cross product of two vectors.....	50
Figure 20. Scalar triple product.....	50
Figure 21. Translation.....	52
Figure 22. Rotation	53
Figure 23. Scaling	54
Figure 24. Manual digitizing setup	58
Figure 25. Raster calculator interface	65
Figure 26. Open disk in \mathbb{R}^2	69
Figure 27. Neighborhood axioms	70
Figure 28. Continuous function	71
Figure 29. Example of a homeomorphic function	72
Figure 30. Equivalent approaches to the definition of a topological space, open sets and neighborhoods and related theorems	73
Figure 31. Interior (upper left), boundary (upper right), closure (lower left) and exterior (lower right) of an open set	76
Figure 32. Separation axioms T_0 , T_1 , and T_2	77
Figure 33. Separation axioms T_3 and T_4	78
Figure 34. Relationship between separation characteristics of topological spaces	78

Figure 35. Simplexes of dimension 0, 1, 2, and 3	82
Figure 36. Valid simplicial complex (left) and invalid simplicial complex (right)	83
Figure 37. Unit balls and cells.....	84
Figure 38. Cell decomposition and skeletons.....	85
Figure 39. Construction of a CW complex.....	86
Figure 40. Two-dimensional spatial data set as cell complex	87
Figure 41. Topological mapping.....	88
Figure 42. Closed polygon boundary check.....	89
Figure 43. Node consistency check	90
Figure 44. Spatial relationships between two simple regions based on the 9-intersection	91
Figure 45. Poset and corresponding diagram	95
Figure 46. Lower bounds	96
Figure 47. Normal completion lattice	100
Figure 48. Normal completion	101
Figure 49. Geometric interpretation of new lattice elements.....	101
Figure 50. The seven bridges of Königsberg	104
Figure 51. Graph of the Königsberg bridge problem.....	104
Figure 52. Complete graphs	105
Figure 53. Isomorphic graphs	106
Figure 54. Connected (G) and disconnected (H) graph.....	107
Figure 55. Directed graphs	107
Figure 56. Planar graph	108
Figure 57. Dual graph.....	108
Figure 58. Undirected and directed graph	109
Figure 59. Membership functions for “short”, “average”, and “tall”	116
Figure 60. Linear membership function	117
Figure 61. Sinusoidal membership function.....	117
Figure 62. Gaussian membership function.....	118
Figure 63. Set inclusion.....	119
Figure 64. Fuzzy set union operators.....	120
Figure 65. Fuzzy set intersection	120
Figure 66. Fuzzy set and its complement	121
Figure 67. Law of the excluded middle and law of contradiction for fuzzy set Average.	121
Figure 68. Membership functions for Tall, Very Tall, and Very Very Tall	123
Figure 69. Membership function for Tall and Not Very Tall.....	123
Figure 70. Membership function for Tall and Slightly Tall	123
Figure 71. Inference rule in MAMDANI’s direct method.....	124

Figure 72. Fuzzy sets of the rules.....	126
Figure 73. Fuzzy inference step 2	126
Figure 74. Fuzzy inference final result	127
Figure 75. Simplified Method.....	127
Figure 76. Membership functions for flat and steep slope.....	128
Figure 77. Membership functions for favorable and unfavorable aspect.	128
Figure 78. Membership function for “high elevation”	129
Figure 79. Analysis with a fuzzy logic approach (left) and a crisp approach (right) ...	132
Figure 80. Platonic solids as building blocks of matter	138
Figure 81. Spatial modeling is a structure preserving mapping from the real world to a spatial model.	142
Figure 82. Data modeling from the real world to a database (digital landscape model), and from there to digital cartographic models and analogue products for visualization	143
Figure 83. Two data layers in a field-based model	145
Figure 84. Layers in an object based model	146

List of Tables

Table 1. Logical Connectors	7
Table 2. Truth tables for logical operators.....	8
Table 3. Logical identities	10
Table 4. Logical implications.....	11
Table 5. Logical relationships involving quantifiers.....	17
Table 6. Rules of inference.....	21
Table 7. Rules of inference involving predicates and quantifiers	22
Table 8. Rules for set operations.....	28
Table 9. ArcInfo overlay commands	29
Table 10. Properties of the Cartesian product	32
Table 11. Properties of interior, closure, and boundary of a set	75
Table 12. Arc table for the arc-node structure.....	88
Table 13. Special elements and the closure operator in the normal completion.....	99
Table 14. Normal completion	99
Table 15. Characteristic function for height classes.....	116
Table 16. Membership values for the height classes.....	116
Table 17. Rules for set operations valid for crisp and fuzzy sets	121
Table 18. Rules valid only for crisp sets.....	121
Table 19. Operators for hedges	122
Table 20. Hedges and their models.....	122
Table 21. Fuzzy inference step 1	126
Table 22. Data models and schemas in database design (the ANSI/SPARC architecture)	144

*“Though this be madness,
Yet there’s method in it.”*

The Binary Bible of Saint Silicon

Logic and set theory can be regarded as the foundation of mathematics. Mathematics is written in the language of logic, and set theory is the very fundament on which all mathematical theories are built. Logic is not only the language of mathematics. It appears also in programming languages as syntactic constructs to express propositions, predicates, and to infer conclusions from given or assumed facts.

The purpose of this book is to provide the reader with the mathematical knowledge needed when they have to deal with spatial information systems. Readers are expected to have a general knowledge of high school mathematics. The use of computers and software for the handling and processing of spatial data requires new contents such as discrete mathematics and topology.

The book is structured into 13 chapters

Chapter 1 gives a brief overview of the structure of mathematics and how the different mathematical disciplines are built on top of more fundamental ones. The next three chapters deal with mathematical logic, the language and foundation of mathematics. Propositional and predicate logic are presented as well as logical inference, the methods of drawing logical conclusions from given facts.

Chapter 5 and 6 are an introduction into the basic notions of sets, set operations, relations, and mappings. These two chapters together with the three chapters on logic represent the foundation for the subsequent chapters dealing with mathematical structures.

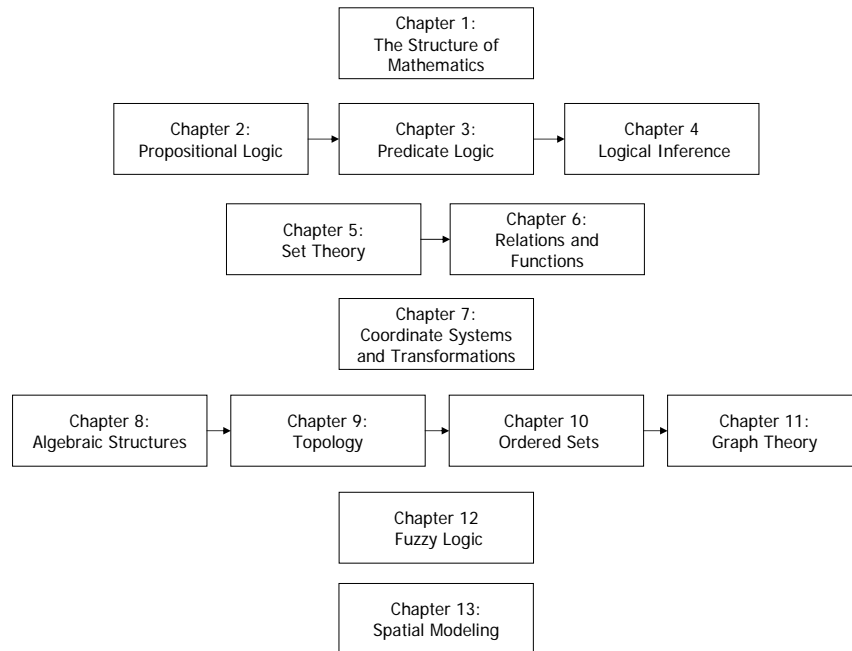
The next chapter on coordinate systems and transformations builds the bridge between the foundation and the more advanced chapters on mathematical structures. Much of chapter 7 would normally be considered to belong either to (analytical) geometry or to linear algebra.

Chapters 8 to 11 present the highly relevant subjects of algebra, topology, ordered sets, and graph theory. These chapters address the mathematical core of many GIS functions from data storage, consistency to spatial analysis.

Uncertainty plays an increasingly important role in GIS. Chapter 12 addresses fuzzy logic and its applications in GIS. It shows how vague concepts can be formalized in mathematical language and how they are applied to spatial decision making.

Chapter 13 is a synthesis of the previous chapters together with some philosophical considerations about a space and time. It shows that spatial modeling is built on solid mathematics as well as that there are challenging and interesting philosophical questions as to how to represent models of spatial features.

The book can be read in several ways as illustrated by the horizontal blocks in the following diagram.



Readers interested in the logical and set theoretic foundations might want to read chapters 2 to 4 and chapters 5 and 6, respectively. For someone with a particular interest in more advanced structures chapters 8 to 11 will be of interest.


Chapters 7, 12, and 13 can be read individually without losing too much of the context. The best way, of course, is to read all the text from chapter 1 to 13.

This book is work in progress and not every chapter or section is complete. The author appreciates any comments and hints that might help to improve the text or its appearance.

*Wolfgang Kainz
Vienna, August 2010*

The Structure of Mathematics

Mathematics is an activity that has been performed by humans since thousands of years. The understanding of what mathematics is has changed over the centuries. In the beginning, mathematics was mainly devoted to practical calculations related to trade and land surveying. Over the centuries, mathematics has become a scientific discipline with many applications in all domains of life. This chapter gives a brief history of mathematics and explains how the different theories and branches of mathematics are rooted in logic and set theory.



1.1 Brief History of Mathematics

The first known cultures that actively performed mathematical calculations in ancient history were the Sumerians, Babylonians, Egyptians, and the Chinese. In the beginning, mathematics was always related to practical problems of commerce, trading and surveying. This is the reason why the ancient cultures mainly developed practical solutions for arithmetic and geometric problems.

In the fifth century before Christ, the ancient Greeks started to do mathematics for its own sake, and to focus the scientific attention to mathematics as a science. The concept of axioms and logical deduction was developed then. The first great example of this approach is *The Elements* of EUCLID, the first textbook on geometry, which was valid until the 19th century.

The Indians and Arabs further developed the number concept and trigonometry. In the 17th and 18th century, the concepts of calculus and analytical geometry were developed as a consequence of the intensive studies in physics and natural sciences.

In the 19th century, mathematicians began to establish an axiomatic foundation of mathematical theories. Starting from a minimal set of axioms statements (theorems) can be derived whose validity can be formally established (proof). This axiomatic approach has been applied since then to formalize mathematics. Logic and set theory play an important role as the language and foundation principle, respectively.

1.2 Sub-disciplines of Mathematics

Logic is a formal language in which mathematical statements are written. It defines rules how to derive new statements from existing ones, and provides methods to prove their validity.

Set theory deals with sets, the fundamental building block of mathematical structures, and the operations defined on them. The notation of set theory is the basic tool to describe structures and operations in mathematical disciplines.

Relations define relationships among elements of a set or several sets. These relationships allow for instance the classification of elements into equivalence classes or the comparison of elements with regard to certain attributes. *Functions* (or mappings) are a special kind of relations.

Sets whose elements are in certain relationships to each other or follow certain operations are mathematical *structures*. We distinguish between three major structures in mathematics, *algebraic*, *order*, and *topologic* structures. In sets with an algebraic structure we can do arithmetic, sets with an order structure allow the comparison of elements, and sets with a topologic structure allow to introduce concepts of convergence and continuity. Calculus is based on topology.

Often, sets carry more than one structure. The real numbers, for instance, carry an algebraic, an order, and a topologic structure. Results from algebraic topology are used in the theory of geographic information systems (GIS).

Figure 1 shows the sub-disciplines and their position in a general concept of mathematics and the fundamental building blocks.

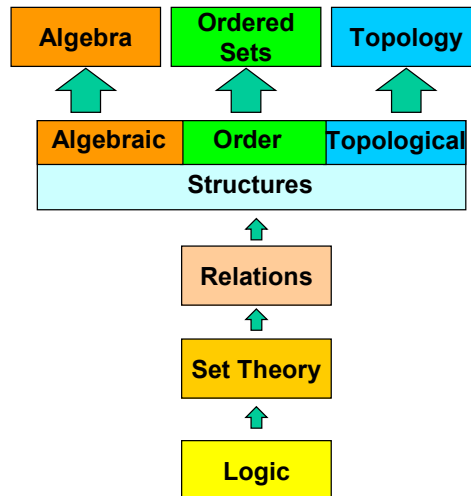



Figure 1. Sub-disciplines of mathematics and their relationships

On top of the different structures and mixed structures, we find the many mathematical disciplines such as calculus, algebra, and (analytical) geometry. The classical theories of great importance in spatial data handling are (analytical) geometry, linear algebra, and calculus. With the introduction of digital technologies of GIS other branches of mathematics became equally important, such as topology, graph theory, and the investigation of non-continuous discrete sets and their operations. The latter two fall under the domain that is usually called *finite* or *discrete mathematics* that plays an important role in computer science and its applications.

Propositional Logic

Propositional logic deals with assertions or statements that are either true or false and operators that are used to combine them. Such statements are called propositions. Any other statements for which we cannot establish whether they are true or false are not the subject of logic. This chapter explains the principles of propositional logic by introducing the concepts of proposition, propositional variable, propositional form, and logical operators. The translation of natural language into propositions and the establishment of their truth-values with the help of truth tables are shown as well.



2.1 Assertion and Proposition

Propositional logic deals with statements that are either true or false. Here, we will only deal with a two-valued logic. This is the logic on which most of the mathematical disciplines are built, and which is used in computing (a bit can only assume two states, on or off, one or zero).

Definition 1 (Assertion and Proposition). An *assertion* is a statement. If an assertion is either true or false, but not both¹, we call it a *proposition*. If a proposition is true, it has a truth-value of *true*; if it is false, it has a truth-value of *false*. Truth-values are usually written as true, false, or T, F, or 1, 0. In the following sections, we will use the 1-0 notation for truth-values.

Example 1. The following statements illustrate the concept of assertion, proposition and truth-values. The following are propositions:

- (1) “It rains.”
- (2) “I pass the exam.”
- (3) “ $3 + 4 = 8$ ”
- (4) “3 is an odd number and 7 is a prime number.”

Assertion (1) and (2) can be true or false. Proposition (3) is false, and (4) is true. The following statements are not propositions:

- (5) “Are you at home?”
- (6) “Use the elevator!”
- (7) “ $x + y < 12$ ”
- (8) “ $x = 6$ ”

(5) and (6) are not assertions (they are a question and a command, respectively), and therefore they cannot be propositions. (7) and (8) are assertions, but no propositions. Their truth-value depends on the value of the variables x and y . Only when we replace the variables with some values, the assertion becomes a proposition.

Often we have to be more general in writing down assertions. For this, we use propositional variables and propositional forms.

Definition 2 (Propositional Variable). A *propositional variable* is an arbitrary proposition whose truth-value is unspecified. We use upper case letters P, Q, R, \dots for propositional variables.

We can combine propositions and propositional variables to form new assertions. For the combination we use words such as “and”, “or”, and “not”.

Example 2. “Beer is good and water has no taste” is a combination of the two propositions “Beer is good” and “Water has no taste” using the connector “and”. “P or not Q.” is a combination of the propositional variables P and Q using the connectors “or” and “not”.

¹ We call a logic in which assertions are either true or false a *two-valued logic*. The *law of the excluded middle* characterized a two-valued logic.

2.2 Logical Operators

In the example above P and Q are called *operands*, the words “and”, “or”, and “not” are *logical operators*, or *logical connectives*. An operator such as “not” that operates only on one operand is called *unary operator*; those that operate on two operands such as “and” and “or” are called *binary operators*.

Definition 3 (Propositional Form). A *propositional form* is an assertion that contains at least one propositional variable. We use upper case Greek letters to denote propositional forms, $\Phi(P, Q, \dots)$.

When we substitute propositions for the propositional variables of a propositional form, we get a proposition. When we use logical connectives to derive new propositions from old ones, the truth-value of the new proposition depends on the logical connective and the truth-values of the old propositions.

Example 3. When P stands for “Vienna is the capital of Austria” and Q stands for “Two is an odd number” then the propositional form in Example 2 “ P or not Q ” becomes the proposition “Vienna is the capital of Austria or two is an even number”.

Logical operators are used to combine propositions or propositional variables. Table 1 shows the most common operators.

Table 1. Logical Connectors

Logical Connector	Symbol	Read or written as
Conjunction	\wedge	and
Disjunction	\vee	or
Exclusive or	\oplus	either ... or but not both
Negation	\neg	not
Implication	\Rightarrow	implies, if...then...
Equivalence	\Leftrightarrow	equivalent, ...if and only if..., iff ²

To determine the truth-value for a combined statement we need to look at every possible combination of truth-values for the operands. This is done using *truth tables* that are defined for every operand. Table 2 shows the truth tables for the most common logical operators. We use the symbols “0” for *false* and “1” for *true*.

Negation is a *unary operator*, i.e., it applies to one variable, and changes the truth-value of a proposition. The other operators apply to two operands. The conjunction (or logical and) is only true if both operands are true. The disjunction (or inclusive or) is true whenever at least one of the operands is true. The exclusive or is only true if either one or the other operand is true, but never both.

When we use the English term “or” we do not make explicit whether we mean the inclusive or exclusive or. It usually follows from the context what we mean. In mathematics, we cannot operate in this way. Therefore, we must make a distinction between inclusive and exclusive or.

In the statement “I go to work or I am tired” the operator indicates an inclusive or. I can go to work and I can be tired at the same time. However, when we say that “I am alive or I am dead” we clearly mean an exclusive or. A person cannot be alive and dead at the same time.³

² The term “iff” meaning “if and only if” is used only in written text.

³ We exclude here the possibility of being a zombie, a state of existence (the living dead) that appears frequently in horror movies.

Table 2. Truth tables for logical operators

Conjunction			Disjunction			Exclusive or		
P	Q	$P \wedge Q$	P	Q	$P \vee Q$	P	Q	$P \oplus Q$
0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	0	1	1
1	0	0	1	0	1	1	0	1
1	1	1	1	1	1	1	1	0

Negation		Implication			Equivalence		
P	$\neg P$	P	Q	$P \Rightarrow Q$	P	Q	$P \Leftrightarrow Q$
0	1	0	0	1	0	0	1
1	0	0	1	1	0	1	0
		1	0	0	1	0	0
		1	1	1	1	1	1

In the implication $P \Rightarrow Q$ we call P the *premise*, *hypothesis*, or *antecedent*, and Q the *conclusion* or *consequence*. The implication can be read in many different ways:

- “If P , then Q ”
- “ P only if Q ”
- “ P implies Q ”
- “ P is a sufficient condition for Q ”
- “ Q if P ”
- “ Q follows from P ”
- “ Q provided P ”
- “ Q is a logical consequence of P ”
- “ Q whenever P ”

If $P \Rightarrow Q$ is an implication then $Q \Rightarrow P$ is called the *converse* and $\neg Q \Rightarrow \neg P$ is called the *contrapositive*.

Example 4. Let us consider the implication “If it rains, then I get wet”. The converse of this implication reads as “If I get wet, then it rains”, and the contrapositive is “If I do not get wet, then it does not rain”.

In natural language, the implication expresses a causal or inherent relationship between a premise and a conclusion. The statement “If I take a shower, then I will get wet” clearly states a causal relationship between taking a shower and getting wet. The statement “If this is an airplane, then it has wings” expresses a property of airplanes.

In propositional logic, there need not be any relationship between the premise and the conclusion of an implication. We have to keep this in mind in order not to get confused by some propositions.

Example 5. Let us take P to be “the moon is larger than the earth” and Q as “the sun is hot”. The implication “If the moon is larger than the earth, then the sun is hot” is true, although there is no relationship whatsoever between the two propositions. The implication is true because P is false and Q is true. According to the truth table for implications, anything (either a true or a false statement) can follow from a false proposition.

Two propositions that have the same truth-values are said to be logically equivalent. $P \Leftrightarrow Q$ can be read in different ways:

- “ P is equivalent to Q ”

“ P is a necessary and sufficient condition for Q ”

“ P if and only if Q ”

“ P iff Q ”

The truth tables for logical operators are used to determine the truth-values of arbitrary propositional forms. Whenever there are n propositional variables involved in a propositional form, we have 2^n possible combinations of true and false to investigate.

Example 6. The truth table for the proposition $\neg(P \wedge \neg Q)$ is constructed as:

P	Q	$\neg Q$	$P \wedge \neg Q$	$\neg(P \wedge \neg Q)$
0	0	1	0	1
0	1	0	0	1
1	0	1	1	0
1	1	0	0	1

We see that for two variables we have to investigate four different cases.

2.3 Types of Propositional Forms

In propositional logic, we distinguish certain propositional forms that are either always true or always false, regardless of the truth-values of the propositional variables.

Definition 4 (Tautology, Contradiction, Contingency). A propositional form whose truth-value is true for all possible truth-values of its propositional variables is called a *tautology*. A *contradiction* (or *absurdity*) is a propositional form that is always false. A *contingency* is a propositional form that is neither a tautology nor a contradiction.

The following examples illustrate the concepts of tautology, contradiction, and contingency.

Example 7. The propositional form $(P \wedge Q) \Rightarrow P$ is a tautology.

P	Q	$P \wedge Q$	$(P \wedge Q) \Rightarrow P$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	1

Example 8. The propositional form $P \wedge \neg P$ is a contradiction.

P	$\neg P$	$P \wedge \neg P$
0	1	0
1	0	0

This propositional form corresponds to the law of the excluded middle (also called “tertium non datur” with its Latin name) stating that something cannot be and not be at the same time.

Example 9. The propositional form $(P \vee \neg Q) \Rightarrow Q$ is a contingency.

P	Q	$\neg Q$	$P \vee \neg Q$	$(P \vee \neg Q) \Rightarrow Q$
0	0	1	1	0
0	1	0	0	1
1	0	1	1	0
1	1	0	1	1

Definition 5 (Logical Identity). Two propositional forms $\Phi(P, Q, R, \dots)$ and $\Psi(P, Q, R, \dots)$ are said to be *logically equivalent* when their truth tables are identical, or when the equivalence $\Phi(P, Q, R, \dots) \Leftrightarrow \Psi(P, Q, R, \dots)$ is a tautology. Such equivalence is also called a *logical identity*.

We can replace one propositional form with its equivalent form. This helps often to simplify logical expressions. Table 3 lists the most important logical identities.

Table 3. Logical identities

1.	$P \Leftrightarrow (P \vee P)$	idempotence of \vee
2.	$P \Leftrightarrow (P \wedge P)$	idempotence of \wedge
3.	$(P \vee Q) \Leftrightarrow (Q \vee P)$	commutativity of \vee
4.	$(P \wedge Q) \Leftrightarrow (Q \wedge P)$	commutativity of \wedge
5.	$[(P \vee Q) \vee R] \Leftrightarrow [P \vee (Q \vee R)]$	associativity of \vee
6.	$[(P \wedge Q) \wedge R] \Leftrightarrow [P \wedge (Q \wedge R)]$	associativity of \wedge
7.	$\neg(P \vee Q) \Leftrightarrow (\neg P \wedge \neg Q)$	DE MORGAN's Laws
8.	$\neg(P \wedge Q) \Leftrightarrow (\neg P \vee \neg Q)$	
9.	$[P \wedge (Q \vee R)] \Leftrightarrow [(P \wedge Q) \vee (P \wedge R)]$	distributivity of \wedge over \vee
10.	$[P \vee (Q \wedge R)] \Leftrightarrow [(P \vee Q) \wedge (P \vee R)]$	distributivity of \vee over \wedge
11.	$(P \vee \mathbf{1}) \Leftrightarrow \mathbf{1}$	
12.	$(P \wedge \mathbf{1}) \Leftrightarrow P$	
13.	$(P \vee \mathbf{0}) \Leftrightarrow P$	
14.	$(P \wedge \mathbf{0}) \Leftrightarrow \mathbf{0}$	
15.	$(P \vee \neg P) \Leftrightarrow \mathbf{1}$	law of the excluded middle
16.	$(P \wedge \neg P) \Leftrightarrow \mathbf{0}$	
17.	$P \Leftrightarrow \neg(\neg P)$	double negation
18.	$(P \Rightarrow Q) \Leftrightarrow (\neg P \vee Q)$	implication
19.	$(P \Leftrightarrow Q) \Leftrightarrow [(P \Rightarrow Q) \wedge (Q \Rightarrow P)]$	equivalence
20.	$[(P \wedge Q) \Rightarrow R] \Leftrightarrow [P \Rightarrow (Q \Rightarrow R)]$	exportation
21.	$[(P \Rightarrow Q) \wedge (P \Rightarrow \neg Q)] \Leftrightarrow \neg P$	absurdity
22.	$(P \Rightarrow Q) \Leftrightarrow (\neg Q \Rightarrow \neg P)$	contrapositive

In the table, **1** and **0** denote propositions that are always true and false, respectively. Identity 18 allows us to replace the implication by negation and disjunction. The equivalence can be replaced by implications through identity 19. Identities 7 and 8 (DE MORGAN's laws) allow the replacement of conjunction by disjunction and vice versa. All the identities can be proven by constructing their truth tables using the truth tables of the logical operators established in Table 1 on page 7.

Example 10. Simplify the following propositional form: $\neg(\neg P \Rightarrow \neg Q)$.

The numbers on the right indicate the identities that have been applied to simplify the propositional form

$$\begin{aligned}
 & \neg(\neg P \Rightarrow \neg Q) & (22) \\
 & \neg(Q \Rightarrow P) & (18) \\
 & \neg(\neg Q \vee P) & (7) \\
 & \neg\neg Q \wedge \neg P & (17) \\
 & Q \wedge \neg P & (4) \\
 & \neg P \wedge Q
 \end{aligned}$$

Many useful tautologies are implications. Table 4 lists the most important of them.

Table 4. Logical implications

1.	$P \Rightarrow (P \vee Q)$	addition
2.	$(P \wedge Q) \Rightarrow P$	simplification
3.	$[P \wedge (P \Rightarrow Q)] \Rightarrow Q$	<i>modus ponens</i>
4.	$[(P \Rightarrow Q) \wedge \neg Q] \Rightarrow \neg P$	<i>modus tollens</i>
5.	$[\neg P \wedge (P \vee Q)] \Rightarrow Q$	disjunctive syllogism
6.	$[(P \Rightarrow Q) \wedge (Q \Rightarrow R)] \Rightarrow (P \Rightarrow R)$	hypothetical syllogism
7.	$(P \Rightarrow Q) \Rightarrow [(Q \Rightarrow R) \Rightarrow (P \Rightarrow R)]$	
8.	$[(P \Rightarrow Q) \wedge (R \Rightarrow S)] \Rightarrow [(P \wedge R) \Rightarrow (Q \wedge S)]$	
9.	$[(P \Leftrightarrow Q) \wedge (Q \Leftrightarrow R)] \Rightarrow (P \Leftrightarrow R)$	

Some of these implications correspond to rules of inference that will be discussed later.

2.4 Applications in GIS

In GIS applications, we find logical operators mainly in spatial analysis and database queries. Figure 2 shows the Raster Calculator of ArcMap Spatial Analyst with its logical connectors. In this example, all raster cells with an elevation between 1,000 and 1,500 will be selected. The logical “and” connector is represented by the character “&”.

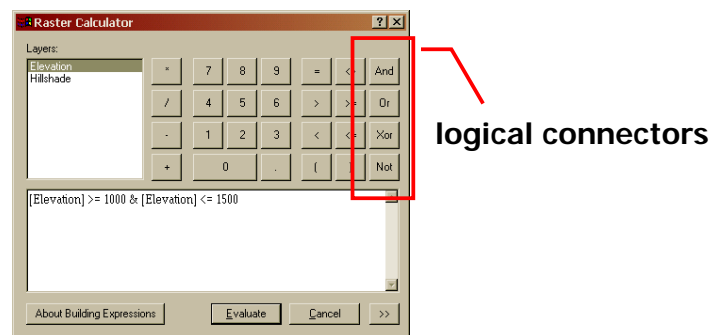


Figure 2. Raster Calculator with logical connectors

The logical implication can be found in every programming language in the form of the *if-statement*, which takes the general form

```
if <condition> then <statement> else <statement>
```

The condition contains an expression that can be evaluated as either true or false (proposition). Logical connectors or comparison operators are often part of the condition. The following AML program prompts the user for a coverage name and deletes it if it exists.

```
&sv covername = [response 'Enter a coverage name']
&if [exists %covername%] &then
    KILL %covername% ALL
&else
    &type Coverage %covername% does not exist!
```

2.5 Exercises


Exercise 1 Construct the truth table of the propositional form $[(P \Leftrightarrow Q) \wedge Q] \Rightarrow P$.

- Exercise 2* Show that $(P \wedge Q) \Rightarrow (P \vee Q)$ is a tautology.
- Exercise 3* Simplify the propositional form $\neg(P \vee Q) \vee (\neg P \wedge Q)$.
- Exercise 4* Let P be the proposition "It is raining." Let Q be the proposition "I will get wet." Let R be the proposition "I am sick."
- Write the following propositions in symbolic notation:
 - If it is raining, then I get wet and I am sick.
 - I am sick if it is raining.
 - I will not get wet.
 - It is raining and I am not sick.
 - Write a sentence in English that corresponds to the following propositions:
 - $R \wedge Q$
 - $(P \Rightarrow Q) \vee \neg R$
 - $\neg(R \vee Q)$
 - $(Q \Rightarrow R) \wedge (R \Rightarrow Q)$
- Exercise 5* Write down the converse and contrapositive of the following propositions:
- "If it rains, then I get wet."
 - "I will stay only if he leaves."
 - "I will not pass the exam, if I do not study hard."
- Exercise 6* For the following expressions, find equivalent expressions using identities. The equivalent expressions must use only \wedge and \neg and be as simple as possible.
- $P \vee Q \vee \neg R$
 - $P \vee [(\neg Q \wedge R) \Rightarrow P]$
 - $P \Rightarrow (Q \Rightarrow P)$
- Exercise 7* In a computer program you have the following statement $x \leftarrow y$ **and** $\text{FUNC}(y, z)$ where x, y are logical variables, FUNC is a logical function and z is an output variable. The value of z is determined by the execution of the function FUNC . Optimizing compilers generate code that is only executed when really needed. Assume such an optimized code has been generated for your program. Can you always rely on that a value for z is computed?

Predicate Logic

The language of propositional logic is not powerful enough to make all the assertions needed in mathematics. We frequently need to make general statements about the properties of an object or relationships between objects, such as “All humans are mortal” or the equation with two variables “ $x + y = 2$ ”.

This chapter introduces the concepts of predicates and quantifiers that enrich the language of logic and allow making assertions in a much more general way than what is possible in propositional logic. The knowledge acquired about predicates will be used to translate natural language statements into the form of predicates.



3.1 Predicates

In the language of propositions we cannot make assertions such as “ $x + y = 5$ ” or “ $x \leq y$ ”, because the truth-value of these statements depends on the values of the variables x and y . Only when we assign values to the variables, the assertions become propositions.

We also make assertions in natural language like “Ann lives in Vienna” or “All humans are mortal” that correspond to a general construct “ x lives in y ” or “all x and $M(x)$ ”. These constructs express a relationship between objects or a property of objects.

Definition 6 (Predicate). A term designating a property or relationship is called a *predicate*.

Assertions made with predicates and variables become a proposition when the variables are replaced by specific values.

Example 11. In the assertion “ x lives in y ” x and y are variables, and “lives in” is a predicate. When we replace x by “John” and y by “Vienna” it becomes the proposition “John lives in Vienna.”

Example 12. Predicates appear commonly in computer programs as control statements of high-level programming languages. The statement “**if** $x < 5$ **then** $y \leftarrow 2 * y$ ” for instance contains the predicate “ $x < 5$ ”. When the program runs the current value of x determines the truth-value of “ $x < 5$ ”.

Some predicates have a well-known notation in mathematics. Examples are “equal to” or “greater than” that are usually written as “ $=$ ” and “ $>$ ”, respectively. Otherwise, we will denote predicates with upper case letters.

Example 13. The assertion “ x is a woman” can be written as $W(x)$, “ x lives in y ” can be written as $L(x, y)$, and “ $x + y = z$ ” could be written as $S(x, y, z)$.

Definition 7 (Variables, Universe). In the expression $P(x_1, x_2, \dots, x_n)$, P is a predicate⁴, and the x_i are variables. When P has n variables we say that it has n arguments or it is an n -place predicate. Values for the variables must come from a set called the *universe of discourse*, or the *universe*. The universe is normally denoted as U and must contain at least one element.

When we take values c_1, c_2, \dots, c_n from the universe and assign them to the variables of a predicate $P(x_1, x_2, \dots, x_n)$, we get a proposition $P(c_1, c_2, \dots, c_n)$.

Definition 8. If $P(c_1, c_2, \dots, c_n)$ is true for every choice of elements from the universe, then we say that P is *valid in the universe* U . If $P(c_1, c_2, \dots, c_n)$ is true for some elements of the universe, we say that P is *satisfiable in the universe* U . The values

⁴ To be precise, we must distinguish between *predicate variables* and *predicate constants*. Whenever we use specific predicates, such as W , L or S in Example 13, we actually deal with predicate constants, whereas an expression like P with no immediate interpretation of the predicate denotes a predicate variable.

c_1, c_2, \dots, c_n that make $P(c_1, c_2, \dots, c_n)$ true are said to *satisfy* P . If $P(c_1, c_2, \dots, c_n)$ is false for every choice of values of the universe we say that P is *unsatisfiable* in U .

3.2 Quantifiers

We have seen that a predicate can become a proposition by substituting values for the arguments. We say that the variables are *bound*. There are two ways of binding variables of predicates.

Definition 9 (Binding of Variables). Variables of predicates can be bound by assigning a value to them, or by quantifying them. We know two *quantifiers*, the *universal* and the *existential* quantifier.

If $P(x)$ is a predicate then the assertion “for all x , $P(x)$ ” (which means, “for all values of x , the assertion $P(x)$ is true”) is a statement in which the variable x is *universally quantified*. The universal quantifier “for all” is written as \forall , and can be read as “for all”, “for every”, “for any”, “for arbitrary”, or “for each.” The statement “for all x , $P(x)$ ” becomes “ $\forall x P(x)$ ”. We say that $\forall x P(x)$ is *true* if and only if $P(x)$ is valid in U ; otherwise, it is false.

If $P(x)$ is a predicate then the assertion “for some x , $P(x)$ ” (which means, “there exists at least one value of x for which the assertion $P(x)$ is true”) is a statement in which the variable x is *existentially quantified*. The existential quantifier “there exists” is written as \exists , and can be read as “there exists”, “for some” or “for at least one”. The statement “for some x , $P(x)$ ” becomes “ $\exists x P(x)$ ”. We say that $\exists x P(x)$ is *true* if and only if $P(x)$ is satisfiable in U ; otherwise, it is false.

There is also a variation of the existential quantifier to assert that there is one and only one element in the universe, which makes a predicate true. This quantifier is read as “there is one and only one x such that...”, “there is exactly one x such that...” or “there is a unique x such that...”. It is written as $\exists!$.

Example 14. Let us assume the universe to be all integers \mathbb{Z} and the following propositions formed by quantification:

- (1) $\forall x[x - 1 < x]$
- (2) $\forall x[x = 5]$
- (3) $\forall x \forall y[x + y > x]$
- (4) $\exists x[x < x + 1]$
- (5) $\exists x[x = 5]$
- (6) $\exists x[x = x + 1]$
- (7) $\exists! x[x = 5]$

Propositions (1), (4), (5), and (7) are true. Propositions (3) is false in the integers; however, it would be true in the positive integers \mathbb{Z}^+ . Propositions (2) and (6) are false.

As we have seen above variables can be bound by assigning values to them. We can also express quantified assertions with propositions by assigning all elements of the universe to the variables and combining them with logical operators.

Definition 10. If the universe U consists of the elements c_1, c_2, c_3, \dots , then the propositions $\forall x P(x)$ and $\exists x P(x)$ can be written as $P(c_1) \wedge P(c_2) \wedge P(c_3) \wedge \dots$ and $P(c_1) \vee P(c_2) \vee P(c_3) \vee \dots$, respectively.

All variables must be bound to transform a predicate into a proposition. If in an n -place predicate m variables are bound, we say that the predicate has $n - m$ free variables.

Example 15. The predicate $P(x, y, z)$ representing “ $x + y < z$ ” has three variables. If we bind one variable, e.g., x is assigned the value 2, then we get the predicate $P(2, y, z)$ with two free variables, representing “ $2 + y < z$ ”.

The order in which the variables are bound is the same as the order in the quantifier list when more than one quantifier is applied to a predicate. Therefore $\forall x \forall y P(x, y)$ has to be evaluated as $\forall x [\forall y P(x, y)]$. The order of the quantifiers is not arbitrary. It affects the meaning of an assertion. $\forall x \exists y$ has not the same meaning as $\exists y \forall x$. The only exception is that we can always replace $\forall x \forall y$ by $\forall y \forall x$, and $\exists x \exists y$ by $\exists y \exists x$.

Example 16. If $P(x, y)$ denotes the predicate “ x is child of y ” in the universe of all persons. Then the proposition $\forall x \exists y P(x, y)$ means, “Everyone is the child of someone”, whereas $\exists y \forall x P(x, y)$ means, “There is a person so that everyone is the child of this person”.

3.3 Quantifiers and Logical Operators

When we express mathematical or natural language statements, we generally need quantifiers, predicates and logical operators. These statements can take on a variety of different forms.

Example 17. Let the universe be the integers and $E(x)$ denote “ x is an even number”, $O(x)$ “ x is an odd number”, $N(x)$ “ x is a non-negative integer”, and $P(x)$ “ x is a prime number.” The following examples show how assertions can be expressed in the language of predicate logic.

- | | |
|--|---|
| (a) There exists an odd integer. | $\exists x O(x)$ |
| (b) Every integer is even or odd. | $\forall x [E(x) \vee O(x)]$ |
| (c) All prime numbers are non-negative. | $\forall x [P(x) \Rightarrow N(x)]$ |
| (d) The only even prime number is two. | $\forall x [(E(x) \wedge P(x)) \Rightarrow x = 2]$ |
| (e) There is only one even prime number. | $\exists! x [E(x) \wedge P(x)]$ |
| (f) Not all prime numbers are odd. | $\neg \forall x [P(x) \Rightarrow O(x)], \text{ or } \exists x [P(x) \wedge \neg O(x)]$ |
| (g) If an integer is not even, then it is odd. | $\forall x [\neg E(x) \Rightarrow O(x)]$ |

In analogy to tautologies, contradictions and contingencies in propositional logic we can also establish types of assertions involving predicate variables⁵.

Definition 11 (Validity of assertions with predicate variables). An assertion involving predicate variables is *valid* if it is true for every universe. An assertion is *satisfiable* if there exist a universe and some interpretations of the predicate variable that make it true. It is *unsatisfiable* if there is no universe and no interpretation that make the assertion true. Two assertions A_1 and A_2 are *logically equivalent* if for every universe

⁵ For the notion of predicate variable, see footnote 4 on page 14.

and every interpretation of the predicate variables $A_1 \Leftrightarrow A_2$, i.e., A_1 is true iff A_2 is true.

The *scope* of a quantifier is the part of the assertion for which variables are bound by this quantifier.

Example 18. In the assertion $\forall x[P(x) \wedge Q(x)]$ the scope of the universal quantifier is $P(x) \wedge Q(x)$. In the assertion $[\exists x P(x)] \Rightarrow [\forall x Q(x)]$ the scope of \exists is $P(x)$ and the scope of \forall is $Q(x)$.

Table 5 shows a list of logical equivalencies and other relationships between assertions involving quantifiers.

Table 5. Logical relationships involving quantifiers

1.	$\forall x P(x) \Rightarrow P(c)$, where c is an arbitrary element of the universe
2.	$P(c) \Rightarrow \exists x P(x)$, where c is an arbitrary element of the universe
3.	$\forall x \neg P(x) \Leftrightarrow \neg \exists x P(x)$
4.	$\forall x P(x) \Rightarrow \exists x P(x)$
5.	$\exists x \neg P(x) \Leftrightarrow \neg \forall x P(x)$
6.	$[\forall x P(x) \wedge Q] \Leftrightarrow \forall x [P(x) \wedge Q]$
7.	$[\forall x P(x) \vee Q] \Leftrightarrow \forall x [P(x) \vee Q]$
8.	$[\exists x P(x) \wedge Q] \Leftrightarrow \exists x [P(x) \wedge Q]$
9.	$[\exists x P(x) \vee Q] \Leftrightarrow \exists x [P(x) \vee Q]$
10.	$[\forall x P(x) \wedge \forall x Q(x)] \Leftrightarrow \forall x [P(x) \wedge Q(x)]$
11.	$[\forall x P(x) \vee \forall x Q(x)] \Rightarrow \forall x [P(x) \vee Q(x)]$
12.	$\exists x [P(x) \wedge Q(x)] \Rightarrow [\exists x P(x) \wedge \exists x Q(x)]$
13.	$[\exists x P(x) \vee \exists x Q(x)] \Leftrightarrow \exists x [P(x) \vee Q(x)]$

The logical equivalencies (3) and (5) can be used to propagate negation signs through a sequence of quantifiers.

Equivalencies (6), (7), (8) and (9) tell us that whenever a proposition occurs within the scope of a quantifier, it can be removed from the scope of the quantifier. Predicates whose variables are not bound by a quantifier can also be removed from the scope of this quantifier.

Statements (10) and (12) show that the universal quantifier *distributes* over the conjunction, but the existential quantifier does not. (13) and (11) show that the existential quantifier distributes over the disjunction, but the universal quantifier does not.

3.4 Compact Notation

The form of logical notation as presented here is often too complex to express relatively simple assertions in mathematical language. Therefore, a compact form of logical notation is used.

For the assertion “for every x such that $x \geq 0$, $P(x)$ is true” we would have to write $\forall x[(x \geq 0) \Rightarrow P(x)]$. Instead we can write in compact notation $\forall x_{x \geq 0} P(x)$. In the same way we write for the assertion “there exists an x such that $x \neq 5$ and $P(x)$ is true” $\exists x[(x \neq 5) \wedge P(x)]$ in the long notation and $\exists x_{x \neq 5} P(x)$ in the compact notation. This notation allows also to propagate the negation sign through quantifiers as mentioned in logical equivalencies (3) and (5) of Table 5 above.

3.5 Applications in GIS

In relational database technology, we use the select operator to select a subset of tuples t (or records) in a relation that satisfies a given selection condition. In general, we can denote the select operator as $\sigma_{\text{selection condition}}(\text{relation name})$ or $\sigma_{\varphi(t)}(R)$ when we substitute selection condition with $\varphi(t)$ and R for the relation name. The selection condition is a predicate, i.e., it designates a property of the tuples, and we can thus write the general selection as a predicative set expression $\{t \in R \mid \varphi(t)\}$.

Let $\text{ARC}(\text{ID}, \text{StartNode}, \text{EndNode}, \text{LPoly}, \text{RPoly})$ be a relation schema describing arcs in a topologically structured data set. The selection operator $\sigma_{(\text{LPoly} = 'A' \text{ OR } \text{RPoly} = 'A')}(\text{ARC})$ results in all arcs that form the boundary of polygon A. A translation of this selection into standard SQL reads as

```
SELECT * FROM ARC WHERE LPoly = 'A' or RPoly = 'A';
```


3.6 Exercises

- Exercise 8* Translate the following assertions into the notation of predicate logic (the universe is given in parentheses):
- (a) If three is odd, some numbers are odd. (integers)
 - (b) Some cats are blue. (animals)
 - (c) All cats are blue. (animals)
 - (d) There are areas, lines, and points. (geometric figures)
 - (e) If x is greater than y and y is greater than z , then x is greater than z . (integers)
 - (f) When it is night all cats are black. (animals)
 - (g) When it is daylight some cats are black. (animals)
 - (h) All students of this course are happy if they pass the mathematics exam. (university students)

Logical Inference

This chapter introduces the concept of a logical argument. Starting from a set of premises (or hypotheses) a conclusion is drawn. If the conclusion follows logically from the premises, the argument is valid. If this is not the case then the conclusion cannot be drawn from the hypotheses.

In a formal mathematical system, we assume a set of axioms that are a set of given unquestioned true statements. From these axioms, we derive assertions that can be shown to be true. These assertions are called theorems. A proof is an argument, which established the truth of a theorem.



4.1 Logical Arguments

Often we assume that certain assumptions are true, and we draw a conclusion from these assumptions. If, for instance, we assume that the two statements “Lisa is beautiful” and “If Lisa is beautiful, all men will adore Lisa” are true, then we can conclude that “All men will adore Lisa”.

Definition 12 (Logical Argument). A *logical argument* consists of a set of *hypotheses* (or *premises*) that are assumed true. The *conclusion* follows from the premises. *Rules of inference* specify which conclusions can be drawn from assertions known or assumed to be true. An argument is said to be *valid* (or *correct*) when the conclusion follows logically from the premises.

Logical arguments are usually written in the form of

$$\begin{array}{c} P_1 \\ P_2 \\ \vdots \\ \frac{P_n}{\therefore Q} \end{array}$$

where the P_i are the premises and Q is the conclusion.

Example 19. The argument presented above is written as

P_1 :	Lisa is beautiful
P_2 :	<u>If Lisa is beautiful, all men will adore Lisa.</u>
Conclusion:	All men will adore Lisa.

The rule of inference applied is of the form

$$\begin{array}{c} P \\ \frac{P \Rightarrow Q}{\therefore Q} \end{array}$$

4.2 Proving Arguments Valid in Propositional Logic

In general, arguments can be proven valid in two ways, using truth tables or using rules of inference. In the first case, an argument has to be translated into its equivalent tautological form. The procedure is straightforward:

- (i.) Identify all propositions.
- (ii.) Assign propositional variables to the propositions.
- (iii.) Write the argument in its tautological form using the propositional variables.
- (iv.) Evaluate the tautological form using a truth table.

Note, that the more propositions are involved the more tedious the procedure becomes. In the case of applying rules of inference, the trick is to find the right rules and apply them properly.

4.2.1 Proving Arguments Valid with Truth Tables

Every logical argument with n premises P_1, P_2, \dots, P_n and the conclusion Q can be written as a propositional form $(P_1 \wedge P_2 \wedge \dots \wedge P_n) \Rightarrow Q$. If this propositional form is a tautology, the argument is correct.

Example 20. The argument in Example 19 above contains the propositions P “Lisa is beautiful” and Q “all men adore Lisa”. It has the tautological form $[P \wedge (P \Rightarrow Q)] \Rightarrow Q$. The proof that this is a tautology is left to the reader.

4.2.2 Proving Arguments Valid with Rules of Inference

A second way to prove an argument valid is to apply rules of inference. They are applied to the premises until the conclusion follows (argument valid) or the conclusion cannot be reached (argument invalid). Table 6 shows the most important rules of inference, their tautological form and the name that was given to them by logicians.

Table 6. Rules of inference

Rule of inference	Tautological form	Name
$\frac{P}{\therefore P \vee Q}$	$P \Rightarrow (P \vee Q)$	addition
$\frac{P \wedge Q}{\therefore P}$	$(P \wedge Q) \Rightarrow P$	simplification
$\frac{P \quad P \Rightarrow Q}{\therefore Q}$	$[P \wedge (P \Rightarrow Q)] \Rightarrow Q$	<i>modus ponens</i>
$\frac{\neg Q \quad P \Rightarrow Q}{\therefore \neg P}$	$[\neg Q \wedge (P \Rightarrow Q)] \Rightarrow \neg P$	<i>modus tollens</i>
$\frac{P \vee Q \quad \neg P}{\therefore Q}$	$[(P \vee Q) \wedge \neg P] \Rightarrow Q$	disjunctive syllogism
$\frac{P \Rightarrow Q \quad Q \Rightarrow R}{\therefore P \Rightarrow R}$	$[(P \Rightarrow Q) \wedge (Q \Rightarrow R)] \Rightarrow [P \Rightarrow R]$	hypothetical syllogism
$\frac{P \quad Q}{\therefore P \wedge Q}$	$(P \wedge Q) \Rightarrow (P \wedge Q)$	conjunction
$\frac{(P \Rightarrow Q) \wedge (R \Rightarrow S) \quad P \vee R}{\therefore Q \vee S}$	$[(P \Rightarrow Q) \wedge (R \Rightarrow S) \wedge (P \vee R)] \Rightarrow [Q \vee S]$	constructive dilemma
$\frac{(P \Rightarrow Q) \wedge (R \Rightarrow S) \quad \neg Q \vee \neg S}{\therefore \neg P \vee \neg R}$	$[(P \Rightarrow Q) \wedge (R \Rightarrow S) \wedge (\neg Q \vee \neg S)] \Rightarrow [\neg P \vee \neg R]$	destructive dilemma

Some of these rules of inference are evident. The disjunctive syllogism, for instance, simply says that if you have two options and you know that one is not available, then you choose the other one⁶.

Example 21. The argument presented in Example 19 above is a straightforward application of the *modus ponens*.

⁶ Most people would agree that even dogs or cats know this.

Example 22. In the same way as above the argument that “Women do not run after me”, and “If I were attractive all women would run after me”, therefore “I am not attractive” is a straightforward application of the *modus tollens*.

4.3 Proving Arguments Valid in Predicate Logic

When we want to prove the validity of an argument that contains predicates and quantifiers, we need more rules. Table 7 shows some of the rules of inference involving predicates and quantifiers.

Table 7. Rules of inference involving predicates and quantifiers

Rule of inference	Name
$\frac{\forall xP(x)}{\therefore P(c)}$	universal instantiation
$\frac{P(x)}{\therefore \forall xP(x)}$	universal generalization
$\frac{\exists xP(x)}{\therefore P(c)}$	existential instantiation
$\frac{P(c)}{\therefore \exists xP(x)}$	existential generalization

The universal instantiation allows us to conclude from the fact that if a predicate is valid in a given universe, then it is also valid for one individual from that universe. The universal generalization permits us to conclude that if we can prove that a predicate is valid for every element of the given universe, then the universally quantified assertion holds.

The existential instantiation concludes from the truth that there is at least one element of the universe for which the predicate is true, that there is one element c for which $P(c)$ is true. The existential generalization allows us to conclude from the truth that a predicate is true for one particular element of the universe, that the existentially quantified assertion $\exists xP(x)$ is true.

Example 23. Let us consider the following argument:

Every man has a brain.
John Williams is a man.
 Therefore, John Williams has a brain.

Let $M(x)$ denote the assertion “ x is a man”, $B(x)$ denote the assertion “ x has a brain”, and W denote John Williams. Then the logical argument can be expressed as:

1. $\forall x[M(x) \Rightarrow B(x)]$
2. $M(W)$
3. $\therefore B(W)$

A formal proof of the argument is as follows:

Assertion	Reasons
1. $\forall x[M(x) \Rightarrow B(x)]$	Hypothesis 1
2. $M(W) \Rightarrow B(W)$	Step 1 and universal instantiation
3. $M(W)$	Hypothesis 2
4. $B(W)$	Steps 2 and 3 and <i>modus ponens</i>

We do not go deeper into the theory of proving arguments in general predicate logic.

4.4 Applications in GIS

Rule-based systems apply rules to data provided using an inference engine. These systems are also called expert systems, and are widely applied in the geosciences. Spatial decision support systems (SDSS) are rule-based systems that are designed and tuned for spatial data.

Rules are stored as implications in the form “if <premise> then <consequence>”. The inference engine examines the given data in the database and determines if they match a given premise. If this is the case, the consequence is applied accordingly. This is a straightforward application of the *modus ponens*.

4.5 Exercises

Exercise 9 Translate the following argument into a symbolic notation and check if it is correct:

P1: If I study well, I will not fail in the mathematics exam.

P2: If I do not play soccer, I study.

P3: I failed the mathematics exam.

Conclusion: I played soccer.

Exercise 10 Translate the following argument into a symbolic notation and check if it is correct using a truth table:

P1: If the Earth is a disk then I do not reach the USA.

P2: If I travel west then I reach the USA.

P3: I do not travel west and I reach the USA.

Conclusion: The Earth is not a disk.

Exercise 11 Translate the following argument into a symbolic notation and check if it is correct:

P1: If 6 is not even, then 5 is no prime number⁷.

P2: 6 is even.


Conclusion: 5 is a prime number.

⁷ A *prime number* is any natural number n that can only be divided by 1 and n .

Set Theory

Sets are the very fundamental building block of many mathematical theories. They are intuitively perceived as a collection of well-distinguished objects. A formal definition and axiomatic foundation of set theory is more complicated and will not be discussed here.

Starting from an intuitive definition of sets, we explore relations between sets and operations on sets. The fundamental principles of subset and set equality as well as set union, intersection, and difference are explained.



5.1 Sets and Elements

Set theory was developed by GEORG CANTOR (1845-1918) as what we call today naïve set theory. This is a more intuitive approach than the axiomatic set theory. However, in naïve set theory there is the possibility for logical contradictions (or paradoxes), something that should not occur in a formal system.

Definition 13 (Set). A set is a collection of well-distinguished objects. Any object of the collection is called an element or a member of the set. An element x of a set S is written as $x \in S$. If x is not a member of S we write $x \notin S$. If a set has a finite number of elements we call it a *finite* set. A set with no elements is called the *empty*, *null* or *void* set and is denoted as $\{\}$ or \emptyset .

There are many ways to specify a set. A finite set can be specified explicitly by listing all its elements. The set A consisting of the natural numbers smaller than 10 can be written as $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, or we can describe the set implicitly by means of a predicate and a free variable, $A = \{x \mid x \in \mathbb{N} \wedge x < 10\}$. We can also draw a set with a VENN diagram (Figure 3).

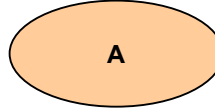


Figure 3. VENN diagram

In order to indicate the “size” of a set we need a measure. This is defined as the number of elements (or cardinality).

Definition 14 (Cardinality). The *cardinality* of a set S is the number of its elements, written as $|S|$.

Example 24. The set $A = \{x \mid x \text{ is a character of the English alphabet}\}$ has cardinality $|A| = 26$.

Example 25. The natural numbers \mathbb{N} are an infinite set. Their cardinality is denoted as \aleph_0 (pronounced as aleph zero⁸). Every set S that has the same cardinality as the natural numbers is called *countably infinite*.⁹ The proof is usually established by finding a one-to-one function that maps the natural numbers to S .

Example 26. The cardinality of the integers \mathbb{Z} and the rational numbers \mathbb{Q} is \aleph_0 . The rational numbers \mathbb{Q} are all fractions of the form $\frac{a}{b}$ where $a, b \in \mathbb{Z}$.

Example 27. The cardinality of the real numbers \mathbb{R} is denoted as c (the continuum). They are said to be uncountable infinite. There are more real numbers than rational numbers. And there are more rational numbers than integers.

⁸ Aleph is a character in the Hebrew alphabet.

⁹ A set is *finite* if there exists a one to one correspondence between its elements and a subset of the natural numbers $\{1, 2, \dots, n\}$ for some n (including $n = 0$ for the empty set). A set is *countable* if it is either finite or countably infinite.

Example 28. The cardinality of the set $A = \{1, 1, 2, 2, 2, 3\}$ is 3, because the elements of a set must be distinguishable. In A , element 1 appears twice, 2 appears three times, and 3 appears once. Since it does not matter how often an element is repeated, the number of elements is three.

Example 29. The set $A = \{\emptyset\}$ has one element, the empty set. Therefore, its cardinality is 1. Although the empty set has cardinality zero, here it appears as an element of set A .

5.2 Relations between Sets

We know two relations between sets, subset and equality. The subset relation refers to the containment of one set in another.

Definition 15 (Subset). If each element of a set A is an element of a set B then A is *subset* of B , written as $A \subseteq B$. B is called *superset* of A , written as $B \supseteq A$. We call a set A a *proper subset* of B when $A \subset B$ and $A \neq B$.

Two sets A and B are equal written as $A = B$ if and only if $A \subset B$ and $B \subset A$.

The following statements can be derived from the definitions of sets and their relationships:

- (i.) If U is the universe of discourse, then $A \subset U$.
- (ii.) For any set A , $A \subset A$.
- (iii.) If $A \subset B$ and $B \subset C$, then $A \subset C$.
- (iv.) The empty set is subset of every set, or for any set A , $\emptyset \subset A$.

5.3 Operations on Sets

In the following, we consider operations on sets that use given sets (*operands*) to produce a new set (*resultant*).

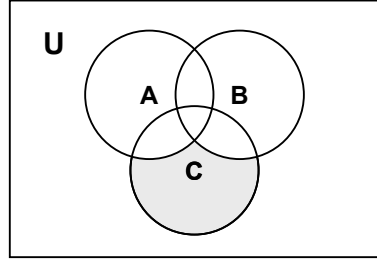
Definition 16 (Union). The *union* of two sets A and B , written as $A \cup B$ is the set $A \cup B = \{x \mid x \in A \vee x \in B\}$.

Definition 17 (Intersection). The *intersection* of two sets A and B , written as $A \cap B$ is the set $A \cap B = \{x \mid x \in A \wedge x \in B\}$. If $A \cap B = \emptyset$, we say that the two sets are *disjoint*.

Definition 18 (Difference). The *difference* of two sets A and B , written as $A - B$ (or $A \setminus B$) is the set $A - B = \{x \mid x \in A \wedge x \notin B\}$.

Definition 19 (Complement). The *complement* of a set A , written as \overline{A} , is the set $\overline{A} = U - A = \{x \mid x \notin A\}$, where U is the universe of discourse.

Example 30. The following Venn diagram illustrates the operations $\overline{A \cup B} \cap C$.



Union and intersection can generally be defined for more than two sets. Let I be an arbitrary finite or infinite index set. Every element $i \in I$ has assigned a set A_i , then the *union* of the A_i is defined as $\bigcup_{i \in I} A_i = \{x \mid \exists i[i \in I \wedge x \in A_i]\}$. In the same way we define the *intersection* of the A_i as $\bigcap_{i \in I} A_i = \{x \mid \forall i[i \in I \Rightarrow x \in A_i]\}$.

Table 8 summarizes some of the most important rules for set operations. They can be easily proven by translating them into their equivalent form in the language of logic.

Table 8. Rules for set operations

1.	$A \cup A = A$	
2.	$A \cap A = A$	
3.	$(A \cup B) \cup C = A \cup (B \cup C)$	
4.	$(A \cap B) \cap C = A \cap (B \cap C)$	associativity
5.	$A \cup B = B \cup A$	
6.	$A \cap B = B \cap A$	commutativity
7.	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	
8.	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	distributivity
9.	$\overline{A \cup B} = \overline{A} \cap \overline{B}$	
10.	$\overline{A \cap B} = \overline{A} \cup \overline{B}$	DE MORGAN's law
11.	$A \cup \emptyset = A$	
12.	$A \cap U = A$	
13.	$A \cup U = U$	
14.	$A \cap \emptyset = \emptyset$	
15.	$A \cup \overline{A} = U$	
16.	$A \cap \overline{A} = \emptyset$	
17.	$\overline{\overline{A}} = A$	
18.	$\overline{U} = \emptyset$	
19.	$\overline{\emptyset} = U$	
20.	$A - B \subset A$	
21.	If $A \subset B$ and $C \subset D$ then $(A \cup C) \subset (B \cup D)$	
22.	If $A \subset B$ and $C \subset D$ then $(A \cap C) \subset (B \cap D)$	
23.	$A \subset A \cup B$	
24.	$A \cap B \subset A$	
25.	If $A \subset B$ then $A \cup B = B$	
26.	If $A \subset B$ then $A \cap B = A$	
27.	$A - \emptyset = A$	
28.	$A \cap (B - A) = \emptyset$	
29.	$A \cup (B - A) = A \cup B$	
30.	$A - (B \cup C) = (A - B) \cap (A - C)$	
31.	$A - (B \cap C) = (A - B) \cup (A - C)$	

Another important concept in set theory is to look at the subsets of a given set. This leads to the definition of the power set.

Definition 20 (Power Set). The set of all subsets of a set A is the *power set of A* , denoted as $\wp(A)$.

If a set is finite, the power set is finite; if a set is infinite, the power set is infinite. The power set of a set with n elements has 2^n elements.

Example 31. The power set of $A = \{1, 2, 3\}$ with three elements has $2^3 = 8$ elements and is written as $\wp(A) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$. Note that the empty set and the set itself are always elements of the power set.

5.4 Applications in GIS

Overlay operations are among the most common functions that a GIS provides for spatial analysis. Since spatial features such as points, arcs and polygons can be regarded as sets, overlay operations correspond to set intersection, union, difference, and complement. Table 9 shows the basic ArcInfo overlay commands and the corresponding set operations in mathematical notation. Other ArcInfo functions such as CLIP, UPDATE, and IDENTITY are based on combinations of overlay and graphical clip operations.

Table 9. ArcInfo overlay commands

Command	A	B	Set Operation
ERASE	in cover	erase cover	$A - B$
INTERSECT	in cover	intersect cover	$A \cap B$
UNION	in cover	union cover	$A \cup B$

Normally, it does not matter in which sequence we apply overlay operations of the same type. The associative and commutative laws for set operations allow the application of intersection and union in arbitrary order.

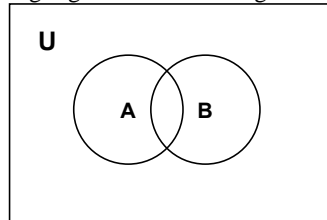
The distributive laws can be used to simplify spatial overlay operations by reducing the number of operations. For example, if we have three data sets A , B and C . We need the intersection of A and B and the intersection of A and C , and finally, compute the union of the results. These operations amount to the following set operations $(A \cap B) \cup (A \cap C)$. This would need three overlay operations. However, the distributive law of set operations allows us to reduce the number of operations to two as $A \cap (B \cup C)$.

When we deal with polygon features in a GIS, we always have an embedding polygon that contains all features of our data set (or coverage). Often it is called world polygon. In set theoretic terms, this corresponds to the universe of discourse¹⁰.

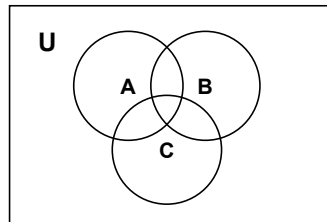
¹⁰ In Chapter 9 we will see that also for topological reasons we need an embedding space for the cell complex of spatial features.

5.5 Exercises

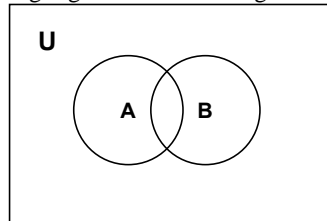
Exercise 12 Highlight in the following VENN diagram $\overline{A \cap B}$.



Exercise 13 Highlight in the following VENN diagram $A \cap (B \cup C)$.



Exercise 14 Highlight in the following VENN diagram $\overline{A \cup B}$.



Exercise 15 Specify the power set for each of the following sets:


- (a) $\{a, b, c\}$
- (b) $\{\{a, b\}, \{c\}\}$
- (c) $\{\emptyset\}$

Exercise 16 Let $U = \{a, b, c, d, e, f, g\}$ be the universe, $A = \{a, b, c, d, e\}$, $B = \{a, c, e, g\}$ and $C = \{b, e, f, g\}$ are sets. Compute the following:

- (i) $\overline{B \cup C}$
- (ii) $\overline{C} \cap A$
- (iii) $B - C$
- (iv) The power set of $B - C$.

Relations and Functions

Relations are a very important concept in mathematics. Based on the fundamental principle of the Cartesian product we will introduce relations as the foundation of mappings and functions. Relations are based on a common understanding of relationships among objects. These relationships may refer to a comparison between objects of the same set, or they involve elements of different sets. Two special types of relations, the equivalence relation and the order relation, play an important role in mathematics. The first is used to classify objects; the latter one is the basis for the theory of ordered sets. In this chapter, we deal only with binary relations only. They are relations between two sets.



6.1 Cartesian Product

Definition 21 (Cartesian Product). The *Cartesian product* (or *cross product*) of two sets A and B , denoted as $A \times B$, is the set of all pairs $\{ \langle a, b \rangle \mid a \in A \wedge b \in B \}$.

Example 32. Let $A = \{1, 2\}$, $B = \{a, b\}$ and $C = \emptyset$. Then

- (a) $A \times B = \{ \langle 1, a \rangle, \langle 1, b \rangle, \langle 2, a \rangle, \langle 2, b \rangle \}$
- (b) $A \times C = \emptyset$

Example 33. Consider the sets $A = \{\text{Vienna, Amsterdam}\}$ and $B = \{\text{Austria, Netherlands, France}\}$. The Cartesian product $A \times B$ is the set of six elements $\{ \langle \text{Vienna, Austria} \rangle, \langle \text{Vienna, Netherlands} \rangle, \langle \text{Vienna, France} \rangle, \langle \text{Amsterdam, Austria} \rangle, \langle \text{Amsterdam, Netherlands} \rangle, \langle \text{Amsterdam, France} \rangle \}$.

The Cartesian product is not commutative, i.e., $A \times B \neq B \times A$. This can easily be seen in the example above.

We can also graphically represent the Cartesian product. Assume we have two sets $A = \{x \mid 2 \leq x \leq 3\}$ and $B = \{y \mid 0 \leq y \leq 1\}$. Then the cross products $A \times B = \{ \langle x, y \rangle \mid 2 \leq x \leq 3 \wedge 0 \leq y \leq 1 \}$ and $B \times A = \{ \langle y, x \rangle \mid 2 \leq x \leq 3 \wedge 0 \leq y \leq 1 \}$ can be graphically represented as in Figure 4.

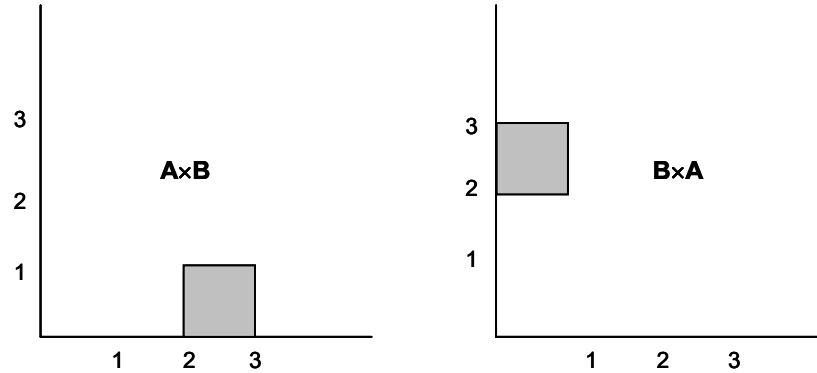


Figure 4. Non-commutativity of the Cartesian product

Some properties of the Cartesian product are listed in Table 10.

Table 10. Properties of the Cartesian product

1.	$A \times (B \cup C) = (A \times B) \cup (A \times C)$
2.	$A \times (B \cap C) = (A \times B) \cap (A \times C)$
3.	$(A \cup B) \times C = (A \times C) \cup (B \times C)$
4.	$(A \cap B) \times C = (A \times C) \cap (B \times C)$

6.2 Binary Relations

Although relations are generally defined with more than two sets, we restrict ourselves here to binary relations between two sets.

Definition 22 (Binary Relation). A binary relation R over $A \times B$ is a subset of $A \times B$. The set A is called the *domain* of R ; B is the *codomain*. We write $\langle a, b \rangle \in R$ also as aRb , and $\langle a, b \rangle \notin R$ is written as $a \not R b$. If the relation is defined over $A \times A$, we call it a relation *on* A .

Example 34. Consider the set $A = \{\text{Vienna, Amsterdam}\}$ and the set $B = \{\text{Austria, Netherlands, France}\}$. The set $C = \{\langle \text{Vienna, Austria} \rangle, \langle \text{Amsterdam, Netherlands} \rangle\}$ is a relation over $A \times B$ that can be read as “is capital of.”

Definition 23 (Inverse Relation). Let R be a relation over $A \times B$. The *inverse relation* (or *inverse*) R^{-1} is defined as the relation over $B \times A$ such that $R^{-1} = \{\langle b, a \rangle \mid \langle a, b \rangle \in R\}$.

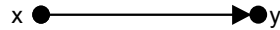
Example 35. Let $A = \{\text{John, Ann, Frank}\}$ and $B = \{\text{Mercedes, BMW}\}$ be two sets of persons and cars. $R = \{\langle \text{John, Mercedes} \rangle, \langle \text{Ann, BMW} \rangle, \langle \text{Frank, BMW} \rangle\}$ is a relation “drives a.” Then $R^{-1} = \{\langle \text{Mercedes, John} \rangle, \langle \text{BMW, Ann} \rangle, \langle \text{BMW, Frank} \rangle\}$ is the inverse relation that can be read as “is driven by.”

6.2.1 Relations and Predicates

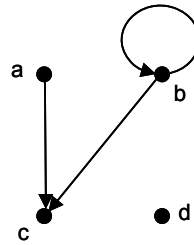
Every binary relation R on a set A corresponds to a predicate with two variables and A as the universe of discourse. If the relation is given, the predicate can be defined as $P(a_1, a_2)$ is true if and only if $\langle a_1, a_2 \rangle \in R$. Likewise, if a predicate P is given we can define a relation R such that $R = \{\langle a_1, a_2 \rangle \mid P(a_1, a_2) \text{ is true}\}$.

6.2.2 Graphic Representation of Binary Relations

It is often convenient to represent relations graphically. For this purpose, we will use *directed graphs* (or *digraphs*¹¹). If there is a relation between the two elements x and y , i.e., xRy , we use the following digraph representation



Example 36. Let $A = \{a, b, c, d\}$ be a set and $R = \{\langle a, c \rangle, \langle b, b \rangle, \langle b, c \rangle\}$ a relation on A . The digraph is represented by the following diagram.



6.2.3 Special Properties of Relations

Some properties of binary relations are so important that they must be discussed in more detail. The following list defines these properties.

¹¹ A graph is defined by two sets V and E , the set of *nodes* (*points* or *vertices*) and the set of *edges* (*arcs* or *lines*), and an incidence relation on $V \times V$ that describes which nodes are connected by edges. If the arcs are directed, we call the graph a *directed graph*.

Definition 24 (Properties of Relations). Let R be a binary relation on a set A . We say that

- (i.) R is *reflexive* if xRx for every x in A .
- (ii.) R is *irreflexive* if xRx for no x in A .
- (iii.) R is *symmetric* if xRy implies yRx for every x, y in A .
- (iv.) R is *antisymmetric* if xRy and yRx together imply $x = y$ for every x, y in A .
- (v.) R is *transitive* if xRy and yRz together imply xRz for every x, y, z in A .

These properties are reflected in certain characteristics of a digraph representation of relations. The digraph of a reflexive relation has a loop on every node of the graph. The graph of an irreflexive relation has no loop on any node. A relation can be neither reflexive nor irreflexive. In this case, it simply has loops on some nodes, but not on all.

The graph of a symmetric relation has either two or no arcs between any two distinct nodes of the graph. For an antisymmetric relation the graph has either one arc or no arc between any two distinct nodes of the graph. Loops may, but need not, occur in the graphs of symmetric and antisymmetric relations.

If in the graph of a transitive relation, there is an arc from x to y and from y to z , then there must also be an arc from x to z .

Example 37. Consider the set of three elements $\{1, 2, 3\}$ and the relations represented in Figure 5.

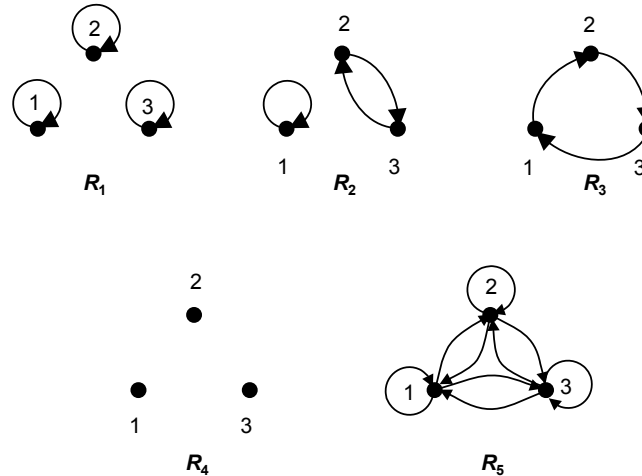


Figure 5. Sample relations

- (a) R_1 is reflexive, symmetric, antisymmetric, and transitive. It is the equality relation on the set. It is not irreflexive.
- (b) R_2 is symmetric but not reflexive, irreflexive, antisymmetric, or transitive.
- (c) R_3 is irreflexive and antisymmetric, but not reflexive, symmetric or transitive.
- (d) R_4 is irreflexive, symmetric, antisymmetric and transitive. It is not reflexive. It is the empty relation on the set.
- (e) R_5 is the universal relation on the set. It is reflexive, symmetric and transitive, but not irreflexive or antisymmetric.

6.2.3.1 Equivalence Relation

A reflexive, symmetric and transitive relation is called *equivalence relation*. An equivalence relation divides a set S into non-empty mutually disjoint sets or *equivalence*

classes $[a] = \{x \mid \langle a, x \rangle \in R\}$ where a is an element of S and R is an equivalence relation. The set of all equivalence classes of S (written as S/R) is called the *quotient set* of S under R , i.e., $S/R = \{[a] \mid a \in S\}$. An element $y \in [a]$ is called a *representative* of the class $[a]$.

Example 38. Let R be the relation \parallel (parallel) on the set of all lines in the plane. This relation is an equivalence relation, because (i) for every line l we have $l \parallel l$ (reflexive), (ii) for every two lines l_1, l_2 we have that if $l_1 \parallel l_2 \Rightarrow l_2 \parallel l_1$ (symmetric), and (iii) if $l_1 \parallel l_2$ and $l_2 \parallel l_3$ then $l_1 \parallel l_3$ (transitivity). The relation classifies the set of lines into the equivalence classes of parallel lines. Every element of one class is a representative of this class.

6.2.3.2 Order Relation

A reflexive, antisymmetric and transitive relation is called *order relation*. Order relations allow the comparison of elements of a set.

Example 39. The subset relation between two sets is an order relation, because (i) for all sets we have $A \subseteq A$ (reflexive), (ii) if $A \subseteq B$ and $B \subseteq A$ then it follows $A = B$ (antisymmetric), and (iii) if $A \subseteq B$ and $B \subseteq C$ then $A \subseteq C$ (transitive).

6.2.4 Composition of Relations

We can generate new relations by composing a sequence of relations. Formally, we define the composition of relations as follows.

Definition 25 (Composition of Relations). Let R_1 be a relation from A to B , and R_2 be a relation from B to C . The *composite relation* from A to C , written as $R_1 R_2$ is defined as

$$R_1 R_2 = \{\langle a, c \rangle \mid a \in A \wedge c \in C \wedge \exists b [b \in B \wedge \langle a, b \rangle \in R_1 \wedge \langle b, c \rangle \in R_2]\}.$$

The composition of relations is not commutative, but associative.

A relation R on a set A can be composed with itself any number of times to form a new relation on the set A . For RR we often write R^2 , for RRR we write R^3 , and so on.

Example 40. If R is the relation “is father of”, then RR is the relation “is paternal grandfather of.”

Let $A = \{a, b, c, d\}$ be a set and consider $R_1 = \{\langle a, a \rangle, \langle a, b \rangle, \langle b, d \rangle\}$ and $R_2 = \{\langle a, d \rangle, \langle b, c \rangle, \langle b, d \rangle, \langle c, b \rangle\}$ to be two relations on A . Then $R_1 R_2 = \{\langle a, c \rangle, \langle a, d \rangle\}$, $R_2 R_1 = \{\langle c, d \rangle\}$, $R_1^2 = \{\langle a, a \rangle, \langle a, b \rangle, \langle a, d \rangle\}$, and $R_2^3 = \{\langle b, c \rangle, \langle c, b \rangle, \langle b, d \rangle\}$.

The composition of relations can be illustrated with a digraph as displayed in Figure 6.

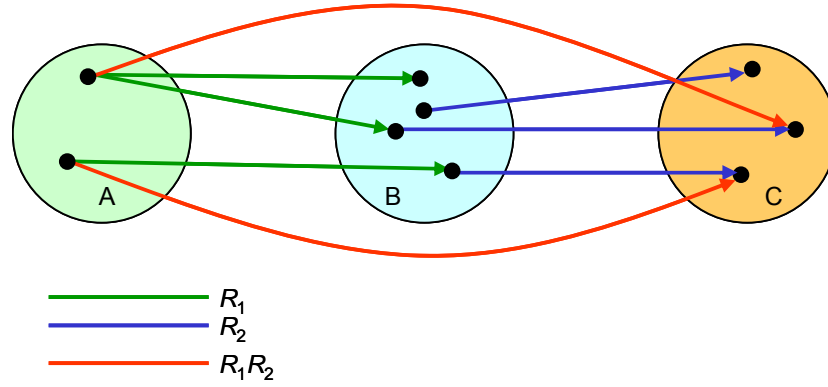


Figure 6. Composition of relations

Let R_1 be a relation from A to B , R_2 and R_3 be a relation from B to C , and R_4 be a relation from C to D . Then the following statements are true:

- (i.) $R_1(R_2 \cup R_3) = R_1R_2 \cup R_1R_3$
- (ii.) $R_1(R_2 \cap R_3) \subset R_1R_2 \cap R_1R_3$
- (iii.) $(R_2 \cup R_3)R_4 = R_2R_4 \cup R_3R_4$
- (iv.) $(R_2 \cap R_3)R_4 \subset R_2R_4 \cap R_3R_4$

6.3 Functions

Functions are a special kind of binary relations. They are used throughout mathematics.

Definition 26 (Function). A function (map, mapping or transformation) f from A to B , written as $f : A \rightarrow B$, is a binary relation from A to B such that for every $a \in A$, there exists a unique $b \in B$ such that $\langle a, b \rangle \in f$. We write $f(a) = b$ and we call A the domain and B the codomain of f . a is the argument and b the value of the function for the argument a .

To correctly specify a function we must indicate the domain, codomain and the value $f(x)$ for every argument x . Note that the important difference between a relation and a function is that for a function it is not possible that an argument has more than one value, and a value must exist for all elements of the domain.

Example 41. Consider the function from the natural numbers to the natural numbers $f : \mathbb{N} \rightarrow \mathbb{N}$ where $f(x) = 2x - 1$. This function maps all natural numbers to the odd numbers. One is mapped to one, two to three, three to five, etc.

Example 42. Consider the sets $A = \{1, 2\}$ and $B = \{a, b, c\}$. When the domain and codomain are finite, we can represent functions as digraphs. In the following Figure 7 (a) and (b) are functions; (c) and (d) are no functions. (c) is not a function because not for every element of the domain we have a value. (d) is not a function because the argument 1 has more than one value assigned.

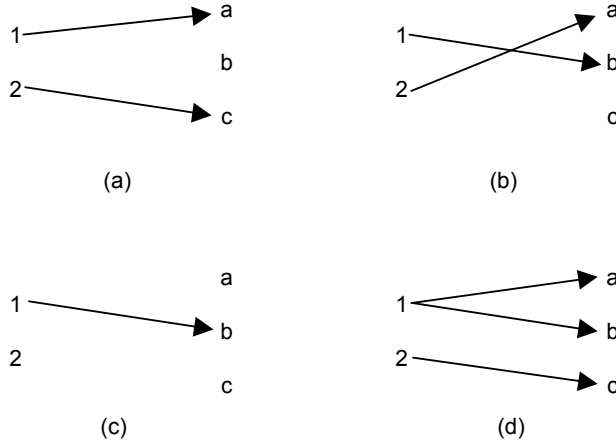


Figure 7. Functions and relations

6.3.1 Composition of Functions

In the same way as with relations, we can generate new functions by composing a sequence of functions.

Definition 27 (Composition of Functions). Let $g : A \rightarrow B$ and $f : B \rightarrow C$ be two functions. The *composite function* $f \circ g$ is a function from A to C and $(f \circ g)(x) = f(g(x))$ for all x in A . The composition of functions is not commutative, but it is associative.

Note that a composite function is only defined when the codomain of the first function g is equal to the domain of the second function f .

Example 43. Let $g : \mathbb{N} \rightarrow \mathbb{N}$ with $g(x) = 2x$ and $f : \mathbb{N} \rightarrow \mathbb{N}$ with $f(x) = x + 1$. The composite function $f(g(x)) = 2x + 1$ and the composite function $g(f(x)) = 2x + 2$.

6.3.2 Classes of Functions

Certain characteristics of functions are so important that a special terminology has been developed for them.

Definition 28 (Surjection). A function f from A to B is called *surjective* (*onto* or *surjection*) if the image of the codomain is the image of the domain, or $f(A) = B$.

Definition 29 (Injection). A function f from A to B is called *injective* (*one-to-one* or *injection*) if distinct arguments have distinct values, or if $a \neq a'$ then $f(a) \neq f(a')$.

Definition 30 (Bijection). A function f from A to B is *bijective* (*one-to-one and onto*, or *bijection*) if it is surjective and injective.

Example 44. Let $f : \mathbb{Z} \rightarrow \{0, 1\}$ be a function from the integers to the set $\{0, 1\}$ defined with $f(x) = \begin{cases} 0 & \text{for } x \text{ is even} \\ 1 & \text{for } x \text{ is odd} \end{cases}$. This function is surjective, but not injective.

Example 45. Consider the function $f : \mathbb{Z} \rightarrow \mathbb{Z}$ in the integers with $f(x) = 2x - 1$. This function is injective, but not surjective.

Example 46. Consider the function $f : \mathbb{Z} \rightarrow \mathbb{Z}$ in the integers with $f(x) = x + 1$. This function is bijective.

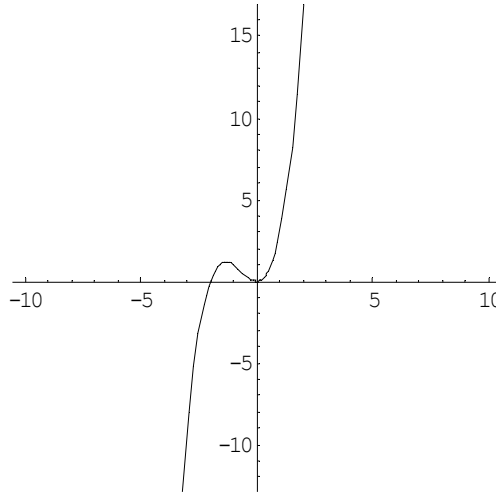
In the case of functions from the real numbers to the real numbers, we can interpret the properties of being surjective, injective or bijective in terms of the graph of the function:

Surjectivity: Every horizontal line intersects the graph of the function at least once.

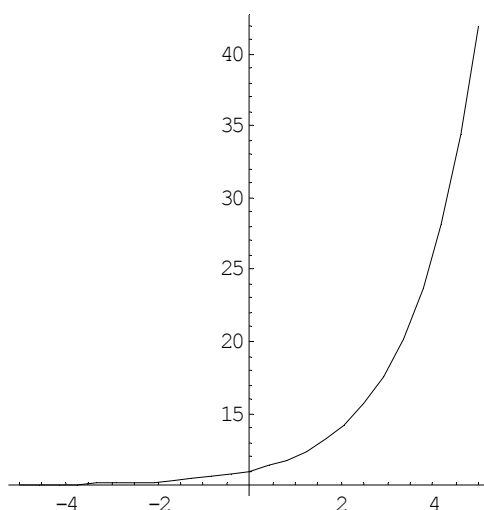
Injectivity: No horizontal line intersects the graph of the function more than once.

Bijectivity: Every horizontal line intersects the graph of the function exactly once.

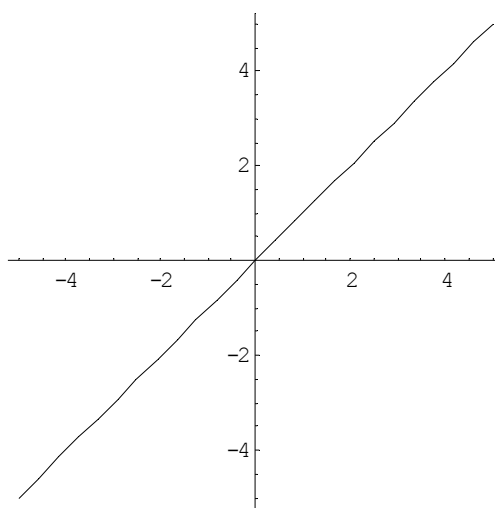
Example 47. Consider the function $f : \mathbb{R} \rightarrow \mathbb{R}$ in the real numbers with $f(x) = x^3 + 2x^2$. Every horizontal line intersects the graph of the function at least once. Therefore, the function is surjective. The function is not injective, because there are lines (e.g., $y = 0$) that intersect the graph more than once.



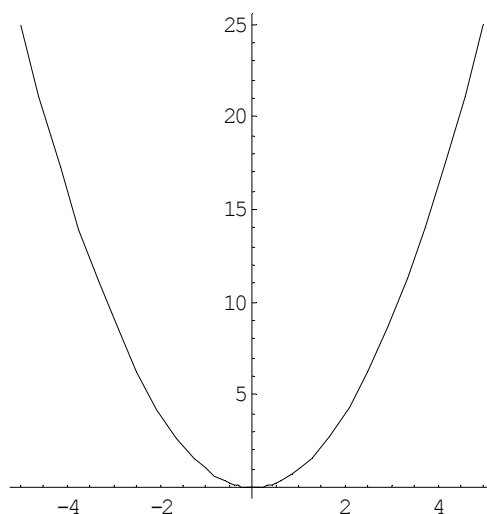
Example 48. Consider the function $f : \mathbb{R} \rightarrow \mathbb{R}$ with $f(x) = 2^x + 10$. No horizontal line intersects the graph more than once. Therefore, the function is injective. It is not surjective, because there are lines that do not intersect the graph at all.



Example 49. Consider the function $f : \mathbb{R} \rightarrow \mathbb{R}$ with $f(x) = x$. Every horizontal line intersects the graph of the function exactly once. Therefore, the function is bijective.



Example 50. Consider the function $f : \mathbb{R} \rightarrow \mathbb{R}$ with $f(x) = x^2$. The function is neither surjective nor injective.



These special properties of functions also propagate through composite functions. If $f \circ g$ is a composite function, then

- (i.) If f and g are surjective, then $f \circ g$ is surjective.
- (ii.) If f and g are injective, then $f \circ g$ is injective.
- (iii.) If f and g are bijective, then $f \circ g$ is bijective.

The converse of these statements is not true. However, we can establish the following:

- (i.) If $f \circ g$ is surjective, then f is surjective.
- (ii.) If $f \circ g$ is injective, then g is injective.
- (iii.) If $f \circ g$ is bijective, then f is surjective and g is injective.

Definition 31 (Inverse Function). Let $f : A \rightarrow B$ be a bijection from A to B . The converse relation of f is called the *inverse function* of f , written as f^{-1} .

The inverse function is only defined when the function is a bijection. The inverse function then is also a bijection.

6.4 Applications in GIS

Relations play an important role in GIS. The best-known examples of relations in GIS are the spatial or *topological relations* between the building blocks of feature data sets. These building blocks correspond to nodes, arcs and polygons in a two-dimensional setting.

Formally, we distinguish between the following relations among the elements of the set of nodes, arcs, and polygons. Every arc has a relation with two nodes (the start node and the end node relation); every arc has a relation with two polygons (the left polygon and the right polygon relation). Figure 8 shows a two-dimensional data set and the topological relations between nodes, arcs and polygons.

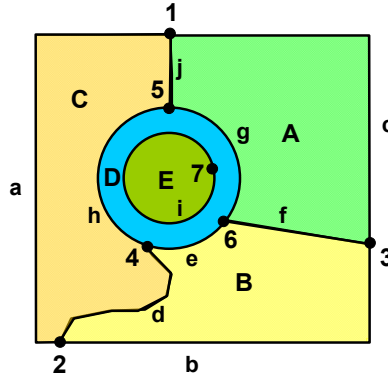


Figure 8. Topological relations

In this example we have the set of nodes, arcs, and polygons defined as $N = \{1, 2, 3, 4, 5, 6\}$, $A = \{a, b, c, d, e, f, g, h, i, j\}$, and $P = \{A, B, C, D, E\}$. The start node – end node relation is defined as

$$AN = \{ \langle a, 1 \rangle, \langle a, 2 \rangle, \langle b, 2 \rangle, \langle b, 3 \rangle, \langle c, 3 \rangle, \langle c, 1 \rangle, \langle d, 2 \rangle, \langle d, 4 \rangle, \langle e, 4 \rangle, \langle e, 6 \rangle, \langle f, 3 \rangle, \langle f, 6 \rangle, \langle g, 6 \rangle, \langle g, 5 \rangle, \langle h, 4 \rangle, \langle h, 5 \rangle, \langle i, 7 \rangle, \langle j, 5 \rangle, \langle j, 1 \rangle \}$$

whereas the left polygon – right polygon relation can be written as

$$AP = \{ \langle a, C \rangle, \langle a, 0 \rangle, \langle b, B \rangle, \langle b, 0 \rangle, \langle c, A \rangle, \langle c, 0 \rangle, \langle d, C \rangle, \langle d, B \rangle, \langle e, D \rangle, \langle e, B \rangle, \\ \langle f, B \rangle, \langle f, A \rangle, \langle g, D \rangle, \langle g, A \rangle, \langle h, C \rangle, \langle h, D \rangle, \langle i, D \rangle, \langle i, E \rangle, \langle j, C \rangle, \langle j, A \rangle \}$$

Other types of relations are those among spatial features in a data set. The best-known examples are the eight relations between simple spatial regions that can be derived from topological invariants of boundary and interior (see chapter 9). Figure 9 shows these relations.

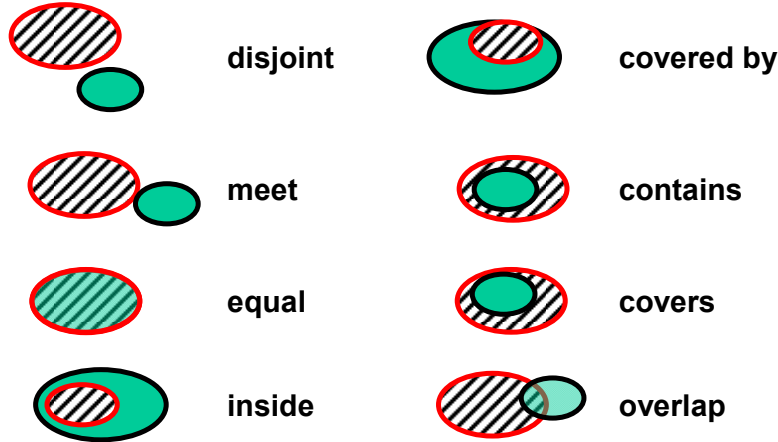


Figure 9. Spatial relations derived from topological invariants

Functions appear in many different forms in GIS. One typical application is map projections. Here, a point on the surface of the earth whose location is given as latitude φ and longitude λ is mapped to a point on a plane by a set of two mapping rules for the easting and northing, respectively, as

$$\text{easting} = f_1(\varphi, \lambda)$$

$$\text{northing} = f_2(\varphi, \lambda)$$

Not every map projection is a function in the mathematical sense. Many map projections, for instance, map the pole to a line, which means that there is more than one value for a given argument. These cases, where a point on the earth is mapped to a line or cannot be mapped at all, are called singularities. Figure 10 shows two projections where the poles are mapped to a line (a) and to a point (b). In the second projection, the South Pole cannot be mapped at all.

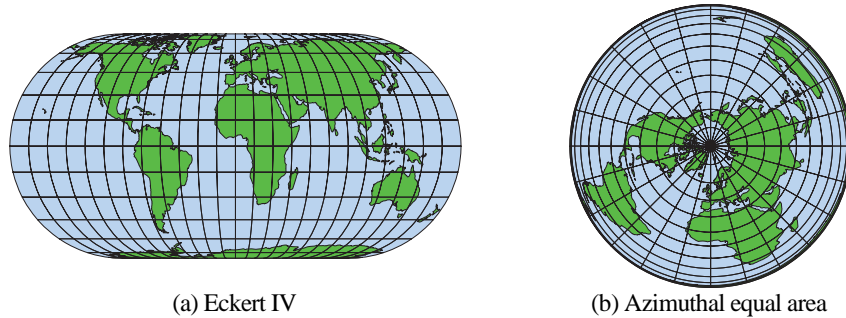


Figure 10. Map projections with singularities

6.5 Exercises

Exercise 17 Let $A = \{a, b\}$, $B = \{2, 3\}$ and $C = \{3, 4\}$. Compute:

- (i) $A \times (B \cup C)$
- (ii) $(A \times B) \cup (A \times C)$
- (iii) $A \times (B \cap C)$
- (iv) $(A \times B) \cap (A \times C)$

Exercise 18 Let $W = \{1, 2, 3, 4\}$ and consider the following relations on W :

- $R_1 = \{<1, 1>, <1, 2>\}$
- $R_2 = \{<1, 1>, <2, 3>, <4, 1>\}$
- $R_3 = \{<1, 3>, <2, 4>\}$
- $R_4 = \{<1, 1>, <2, 2>, <3, 3>\}$

Check these relations whether they are reflexive, irreflexive, symmetric, antisymmetric, or transitive.

Exercise 19 Let $X = \{1, 2, 3, 4\}$. Which one of the following relations is symmetric and which one is transitive? In case the relation is not symmetric or transitive explain why.

- $f = \{<2, 3>, <1, 4>, <2, 1>, <3, 2>, <4, 4>\}$
- $g = \{<3, 1>, <4, 2>, <1, 1>\}$
- $h = \{<2, 1>, <3, 4>, <1, 4>, <2, 1>, <4, 4>\}$

Exercise 20 Let $X = \{1, 2, 3, 4\}$. Which one of the following relations is a function from X to X ? In case the relation is not a function explain why.

- $f = \{<2, 3>, <1, 4>, <2, 1>, <3, 2>, <4, 4>\}$
- $g = \{<3, 1>, <4, 2>, <1, 1>\}$
- $h = \{<2, 1>, <3, 4>, <1, 4>, <2, 1>, <4, 4>\}$

Exercise 21 Let R_1 and R_2 be relations on a set $A = \{a, b, c, d\}$ where $R_1 = \{<a, a>, <a, c>, <c, d>\}$ and $R_2 = \{<a, d>, <b, c>, <b, b>, <c, d>\}$. Find $R_1 R_2$, $R_2 R_1$, R_1^2 , and R_2^3 .

Exercise 22 Let $f: \mathbb{R} \rightarrow \mathbb{R}$ be defined as $f(x) = x^2 - 3x + 2$. Compute $\frac{f(x+h) - f(x)}{h}$.

Exercise 23 Let f and g be functions from $X = \{1, 2, 3, 4, 5\}$ in X defined as:

- $f = \{<1, 3>, <2, 5>, <3, 3>, <4, 1>, <5, 2>\}$
- $g = \{<1, 4>, <2, 1>, <3, 1>, <4, 2>, <5, 3>\}$

Determine (i) the codomain of f and g . (ii) Determine $g \circ f$ and $f \circ g$.

Coordinate Systems and Transformations

All points in space can be uniquely referenced by their coordinates. Depending on the type of space, we distinguish between different coordinate systems such as Cartesian coordinate systems for Euclidean spaces or spherical coordinate systems for the surface of a sphere and elliptical coordinate systems for the surface of an ellipsoid. The sphere and the ellipsoid are geometric bodies used to approximate the shape of the earth.

Spatial features such as points, arcs and polygons as well as raster cells are spatially referenced through their coordinates. Often, it is necessary to apply transformations to these coordinates in order to shift, rotate, scale or warp the features. In this chapter, we will discuss frequently used coordinate systems and transformations applied to geometric features in a Euclidean space.

7.1 Coordinate Systems

A coordinate system functions to assign any point in space a pair or triple of real numbers, its coordinates. The most common coordinate systems are rectangular or Cartesian coordinate systems and polar coordinate systems. In this chapter, we deal with a two- or three-dimensional real space (also called the *Euclidean space*) where every point has real-valued coordinates.

7.1.1 Cartesian Coordinate Systems

In the real plane \mathbb{R}^2 every point P has a unique pair of real numbers (x, y) with $x, y \in \mathbb{R}$ assigned. On the other hand, every pair of real numbers (x, y) defines uniquely a point in the real plane. We define a single point O , the *origin*, and two perpendicular lines through that point, the *axes*. The horizontal axis is called x -axis, the vertical one is the y -axis. Every point P is uniquely defined by its *Cartesian coordinates* $P(x, y)$. Figure 11 illustrates these Cartesian coordinates.

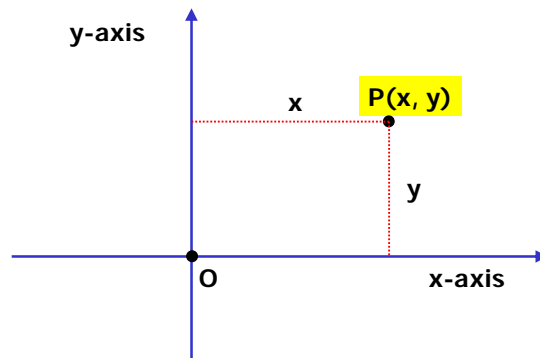


Figure 11. Cartesian coordinate system in the plane

We can easily extend the 2-dimensional coordinates in the plane to 3-dimensional coordinates in space by defining a Cartesian coordinate system in \mathbb{R}^3 . Every point P is then clearly defined by the triple (x, y, z) of Cartesian coordinates (Figure 12).

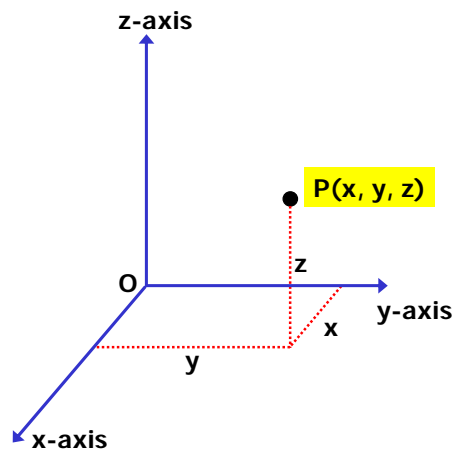


Figure 12. Cartesian coordinate system in 3-D space

7.1.2 Polar Coordinate Systems

A different way of assigning a point in the plane unique coordinates is the use of polar coordinates. They are defined in a polar coordinate system which is given by a fixed point O , the *origin* or *pole*, and a line through the pole, the *polar axis*. Every point in the plane is then determined by its distance from the pole, the radius r , and the angle φ between the radius and the polar axis (Figure 13).

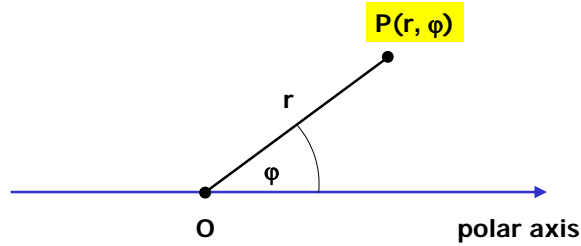


Figure 13. Polar coordinate system in the plane

In a three-dimensional polar coordinate system (or *spherical coordinate system*) a point P is defined by the radius r from the origin to the point and two angles: the angle φ between the projection of \overline{OP} onto the x, y -plane, and the angle θ between \overline{OP} and the positive z -axis (Figure 14).

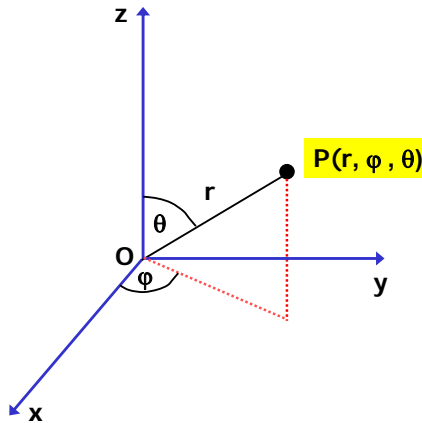


Figure 14. Spherical coordinate system

7.1.3 Transformations between Cartesian and Polar Coordinate Systems

The relationships between x and y and r and φ are illustrated in Figure 15 and can be expressed by the following correspondences:

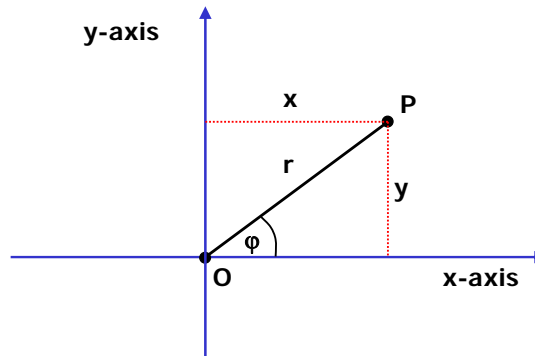


Figure 15. Conversion between Cartesian and polar coordinates in the plane

$$\begin{aligned}
x &= r \cos \varphi \\
y &= r \sin \varphi \\
r &= \sqrt{x^2 + y^2} \\
\tan \varphi &= \frac{y}{x} \text{ with } \varphi \in [0, 2\pi] \setminus \left\{ (2k+1)\frac{\pi}{2} \mid k \in \mathbb{Z} \right\}
\end{aligned}$$

Example 51. Given the Cartesian coordinates of the point $P(3,4)$ we can compute its polar coordinates as $r = \sqrt{3^2 + 4^2} = \sqrt{9+16} = \sqrt{25} = 5$ and $\tan \varphi = \frac{4}{3} = 1.3333$, i.e., $\varphi = 53.13^\circ$. The point thus has the polar coordinates $P(5, 53.13)$.

The conversion between three-dimensional Cartesian coordinates and spherical coordinates can be performed using the following formulas (see also Figure 16):

$$\begin{aligned}
x &= r \sin \theta \cos \varphi \\
y &= r \sin \theta \sin \varphi \\
z &= r \cos \theta \\
r &= \sqrt{x^2 + y^2 + z^2} \\
\sin \varphi &= \frac{y}{\sqrt{x^2 + y^2}} \\
\cos \varphi &= \frac{x}{\sqrt{x^2 + y^2}} \\
\cos \theta &= \frac{z}{r} \\
\tan \theta &= \frac{\sqrt{x^2 + y^2}}{z} \text{ with } \theta \in [0, \pi] \\
\tan \varphi &= \frac{y}{x} \text{ with } \varphi \in [0, 2\pi] \setminus \left\{ (2k+1)\frac{\pi}{2} \mid k \in \mathbb{Z} \right\}
\end{aligned}$$

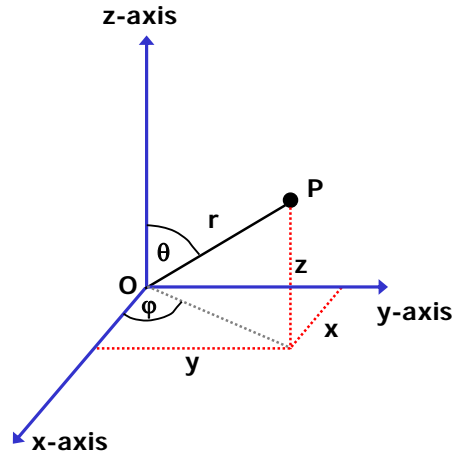


Figure 16. Conversion between Cartesian coordinates and spherical coordinates

Example 52. Given a point $P(2,3,4)$ in the three-dimensional Euclidean space \mathbb{R}^3 we can compute its spherical coordinates as $r = \sqrt{2^2 + 3^2 + 4^2} = \sqrt{4+9+16} = \sqrt{29} = 5.385$, $\cos \theta = \frac{4}{5.385} = 0.743$ and $\tan \varphi = \frac{3}{2} = 1.5$. From this we get $\theta = 42.03^\circ$ and $\varphi = 56.31^\circ$.

7.1.4 Geographic Coordinate System

A special case of a spherical coordinate system is the *geographic coordinate system*, which is used to identify locations on the surface of the earth (Figure 17). The origin M of the geographic coordinate system is the center of the earth. The equator E lies in the plane defined by the x - and y -axes. The circle G defined by the intersection of the x, z -plane with the earth is the zero meridian through Greenwich. Every point P on the surface of the earth is uniquely defined by its latitude φ and longitude λ , denoted as $P(\varphi, \lambda)$.

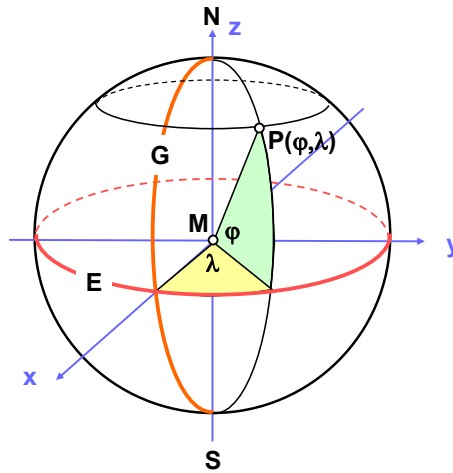


Figure 17. Geographic coordinate system

The latitude is measured as the angle between the equatorial plane and the radius from the origin to the point towards north (positive latitude) or south (negative latitude). A latitude on the northern hemisphere ranges from 0 to 90, and from 0 to -90 on the southern hemisphere. Note that this is different from the way how the angle θ is defined for spherical coordinates.

The longitude is the angle between the planes through the zero meridian and the origin, and the plane through the point P and the origin. Longitudes are positive from 0 to 180 towards east and negative from 0 to 180 west.

Every circle through the poles is called a *meridian*; every circle parallel to the equatorial plane is called a *parallel*. For practical calculations, the radius R of the earth is assumed 6,370 kilometers.

Example 53. The airport of Vienna, Austria, has a latitude of $48^{\circ}07'$ North and a longitude of $16^{\circ}34'$ East, or $VIE(48.1167, 16.5667)$.

7.2 Vectors and Matrices

Vectors and matrices play an important role in the analytical treatment of geometric figures. We can represent points in space by their respective point vectors, and we can apply many calculations related to the characteristics of geometric figures using vector representations.

7.2.1 Vectors

In section 8.2.4 we have defined the algebraic structure of a vector space. The elements of a vector space are called vectors several axioms for operations among vectors and

vectors with scalars are defined. Here, we define vectors as a class of arrows in two- or three-dimensional real space.

Definition 32 (Vector). A *vector* is a class of parallel, directed arrows of the same length in space. A single arrow is called a *representative* of the vector.

For simplicity, we will not make a difference between a vector and a representative, and will simply call a representative a vector. The tail of a vector is called the *initial point*, and the head of the arrow is called the *terminal point*.

Every point $P(x, y, z)$ in \mathbb{R}^3 can be represented by its point vector $\vec{P} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ as shown in

Figure 18¹². In \mathbb{R}^2 the point vector components reduce to two for the x - and y -coordinate components.

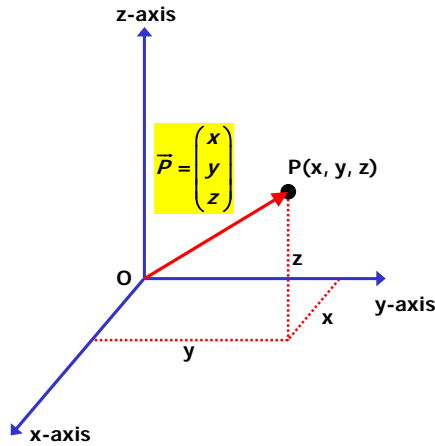


Figure 18. Point vector

The *length* of a vector is defined as $|\vec{P}| = \sqrt{x^2 + y^2 + z^2}$ in \mathbb{R}^3 and $|\vec{P}| = \sqrt{x^2 + y^2}$ in \mathbb{R}^2 . A vector of length 1 is called a *unit vector*.

From section 8.2.4 we know that we can define operations of addition and multiplication with a scalar for vectors:

$$\vec{a} + \vec{b} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} + \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = \begin{pmatrix} a_x + b_x \\ a_y + b_y \\ a_z + b_z \end{pmatrix} \text{ and } \lambda \vec{a} = \lambda \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \begin{pmatrix} \lambda a_x \\ \lambda a_y \\ \lambda a_z \end{pmatrix}$$

Example 54. The sum of the two three-dimensional vectors (1,2,3) and (4,5,6) is

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} + \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix} = \begin{pmatrix} 1+4 \\ 2+5 \\ 3+6 \end{pmatrix} = \begin{pmatrix} 5 \\ 7 \\ 9 \end{pmatrix}.$$

Beside the addition of vectors and the multiplication of a vector with a scalar, we know three different vector products. They all have also a geometric interpretation.

¹² For the sake of a more compact vector notation we also write $\vec{P} = (x, y, z)$.

Definition 33 (Dot product). If \vec{a} and \vec{b} are two vectors the *dot product* (or *Euclidean inner product*) is defined as

$$\vec{a} \cdot \vec{b} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \cdot \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = a_x b_x + a_y b_y + a_z b_z$$

The result of the dot product is always a number (scalar). The dot product is commutative and distributive, but not associative:

$$\begin{aligned} \vec{a} \cdot \vec{b} &= \vec{b} \cdot \vec{a} \\ (\vec{a} + \vec{b}) \cdot \vec{c} &= \vec{a} \cdot \vec{c} + \vec{b} \cdot \vec{c} \end{aligned}$$

The dot product can be used to compute the angle φ between two vectors \vec{a} and \vec{b} according to the following formula:

$$\cos \varphi = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|}$$

Example 55. The angle φ between the two vectors $\vec{a} = (1, 2, 3)$ and $\vec{b} = (3, 1, 1)$ is calculated according to the formula as $\cos \varphi = \frac{1 \cdot 3 + 2 \cdot 1 + 3 \cdot 1}{\sqrt{1 + 4 + 9} \cdot \sqrt{9 + 1 + 1}} = \frac{8}{\sqrt{14} \cdot \sqrt{11}} = 0.645$. It follows that $\varphi = 49.86^\circ$.

Definition 34 (Cross product). If \vec{a} and \vec{b} are two vectors the *cross product* is defined as

$$\vec{c} = \vec{a} \times \vec{b} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \times \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = \begin{pmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{pmatrix}$$

The result of the cross product is a vector. It is distributive, but not commutative:

$$\begin{aligned} (\vec{a} + \vec{b}) \times \vec{c} &= \vec{a} \times \vec{c} + \vec{b} \times \vec{c} \\ \vec{a} \times \vec{b} &= -\vec{b} \times \vec{a} \end{aligned}$$

The cross product can be geometrically interpreted as illustrated in Figure 19:

- The product vector \vec{c} is perpendicular to the vectors \vec{a} and \vec{b} .
- \vec{a}, \vec{b} and \vec{c} form a right-handed coordinate system.
- The length of \vec{c} is equal to the area of the parallelogram spanned by \vec{a} and \vec{b} , where φ is the angle between the two vectors, according to the following formula $|\vec{c}| = |\vec{a} \times \vec{b}| = |\vec{a}| \cdot |\vec{b}| \cdot \sin \varphi$.

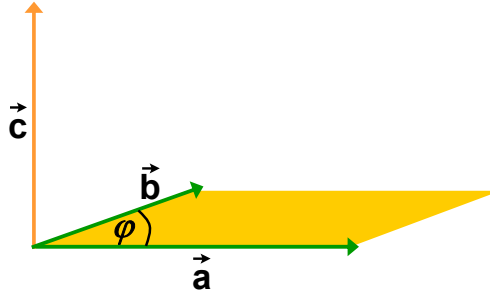


Figure 19. Cross product of two vectors

Example 56. The cross product of the two vectors $\vec{a} = (1, 0, 0)$ and $\vec{b} = (1, 1, 0)$ is equal to

$$\vec{a} \times \vec{b} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \cdot 0 - 0 \cdot 1 \\ 0 \cdot 1 - 1 \cdot 0 \\ 1 \cdot 1 - 0 \cdot 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \text{ and the length of the vector is 1. The angle between } \vec{a} \text{ and}$$

\vec{b} is 45° . Therefore we can compute the length of the cross product as $1 \cdot \sqrt{2} \cdot \frac{\sqrt{2}}{2} = 1$.

Definition 35 (Scalar triple product). If \vec{a}, \vec{b} and \vec{c} are three vectors the *scalar triple product* is defined as

$$(\vec{a}\vec{b}\vec{c}) = (\vec{a} \times \vec{b}) \cdot \vec{c} = \vec{c} \cdot (\vec{a} \times \vec{b}) = a_x(b_y c_z - b_z c_y) - a_y(b_x c_z - b_z c_x) + a_z(b_x c_y - b_y c_x)$$

If the three vectors do not lie in the same plane, then they form a parallelepiped when they are positioned with the common initial point (Figure 20). The result of the scalar triple product is a number equal to the volume of this parallelepiped and can also be computed according to the formula

$$(\vec{a}\vec{b}\vec{c}) = |\vec{a} \times \vec{b}| \cdot |\vec{c}| \cdot \cos \gamma$$

where γ is the angle between the cross product vector of $\vec{a} \times \vec{b}$ and \vec{c} .

Example 57. The volume of the parallelepiped with $\vec{a} = (2, -6, 2)$, $\vec{b} = (0, 4, -2)$, and $\vec{c} = (2, 2, -4)$ is computed by inserting into the formula given in Definition 35 as $2 \cdot [4 \cdot (-4) - (-2) \cdot 2] - (-6) \cdot [0 \cdot (-4) - (-2) \cdot 2] + 2 \cdot [0 \cdot 2 - 4 \cdot 2] = 2 \cdot (-12) - (-6) \cdot 4 + 2 \cdot (-8) = -24 + 24 - 16 = -8$

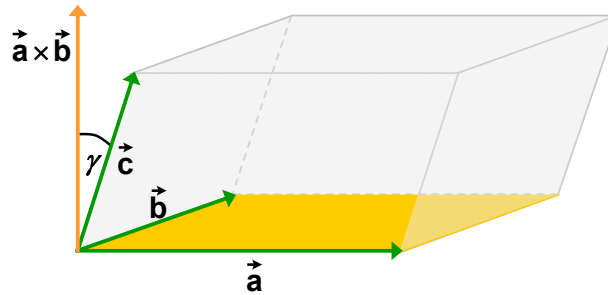


Figure 20. Scalar triple product

The following correspondences hold between the dot product, cross product and the scalar triple product.

$$\begin{aligned}\vec{a} \times (\vec{b} \times \vec{c}) &= (\vec{a} \cdot \vec{c})\vec{b} - (\vec{a} \cdot \vec{b})\vec{c} \\ (\vec{a} \times \vec{b}) \cdot (\vec{c} \times \vec{d}) &= (\vec{a} \cdot \vec{c})(\vec{b} \cdot \vec{d}) - (\vec{a} \cdot \vec{d})(\vec{b} \cdot \vec{c}) \\ (\vec{a} \times \vec{b})^2 &= \vec{a}^2 \vec{b}^2 - (\vec{a} \cdot \vec{b})^2 \\ (\vec{a} \times \vec{b}) \times (\vec{c} \times \vec{d}) &= \vec{c}(\vec{a}\vec{b}\vec{d}) - \vec{d}(\vec{a}\vec{b}\vec{c})\end{aligned}$$

7.2.2 Matrices

Rectangular arrays of real numbers occur in many contexts in mathematics and as data structure in applications of computer science.

Definition 36 (Matrix). A *matrix* is a rectangular array of real numbers. The numbers in the array are called the entries in the matrix.

A matrix M is a rectangular array of numbers with m rows and n columns represented as:

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} & \cdots & m_{1n} \\ m_{21} & m_{22} & m_{23} & \cdots & m_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{m1} & m_{m2} & m_{m3} & \cdots & m_{mn} \end{pmatrix}$$

We call M a $m \times n$ -matrix. We also know that the matrices form a vector space and that we can multiply matrices with matrices and matrices with vectors according to the following rules:

Let A and B be two matrices. Their *sum* is defined as

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} + \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{pmatrix} = \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \cdots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & \cdots & a_{2n} + b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} & \cdots & a_{mn} + b_{mn} \end{pmatrix}$$

Note that the sum has the same number of rows and columns of the input matrices, and that both matrices must have the same number of rows and columns. If the number of rows and columns does not match, the sum is not defined.

The *multiplication of a matrix A with a scalar s* is defined as

$$sA = s \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} = \begin{pmatrix} sa_{11} & sa_{12} & \cdots & sa_{1n} \\ sa_{21} & sa_{22} & \cdots & sa_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ sa_{m1} & sa_{m2} & \cdots & sa_{mn} \end{pmatrix}$$

The *product of two matrices A and B* is only defined if the number of columns of A is equal the number of rows of B . Given a $m \times p$ -matrix A and a $p \times n$ -matrix B the product of $C = AB$ of A and B is a $m \times n$ -matrix where every element c_{ij} of the product is calculated according to the following schema

$$\begin{pmatrix} a_{11} & \cdots & a_{1p} \\ \vdots & \ddots & \vdots \\ a_{i1} & \cdots & a_{ip} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mp} \end{pmatrix} \begin{pmatrix} b_{11} & \cdots & b_{1j} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{p1} & \cdots & b_{pj} & \cdots & b_{pn} \end{pmatrix} = \begin{pmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ \vdots & c_{ij} & \vdots \\ \vdots & \ddots & \vdots \\ c_{m1} & \cdots & c_{mn} \end{pmatrix}$$

$$\text{and } c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{ip}b_{pj} = \sum_{k=1}^p a_{ik}b_{kj}.$$

Example 58. The product of the two matrices $\begin{pmatrix} r & s \\ t & u \end{pmatrix}$ and $\begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{pmatrix}$ is computed as

$$\begin{pmatrix} r & s \\ t & u \end{pmatrix} \begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{pmatrix} = \begin{pmatrix} ra_1 + sb_1 & ra_2 + sb_2 & ra_3 + sb_3 \\ ta_1 + ub_1 & ta_2 + ub_2 & ta_3 + ub_3 \end{pmatrix}.$$

Example 59. The product of $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ and $\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}$ is calculated as

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 1 \cdot 5 + 2 \cdot 7 & 1 \cdot 6 + 2 \cdot 8 \\ 3 \cdot 5 + 4 \cdot 7 & 3 \cdot 6 + 4 \cdot 8 \end{pmatrix} = \begin{pmatrix} 19 & 22 \\ 43 & 50 \end{pmatrix}.$$

7.3 Transformations

When we rotate, shift or scale geometric figures we apply geometric transformations. Here, we will focus on plane coordinate systems and their transformations. Another problem related to transformations is to determine the parameters of a transformation between two plane coordinate systems that compensates for scaling, rotation, skew and translation. We will discuss the HELMERT (or similarity) transformation and the affine transformation that both provide solutions to this problem.

7.3.1 Geometric Transformations

In the following sections, we will discuss geometric transformations in the plane using Cartesian coordinates.

7.3.1.1 Translation

The shift of a geometric figure in horizontal and vertical direction results in a translation operation (Figure 21). The translation factor in x -direction is t_x , the factor in the y -direction is t_y . They need not be the same.

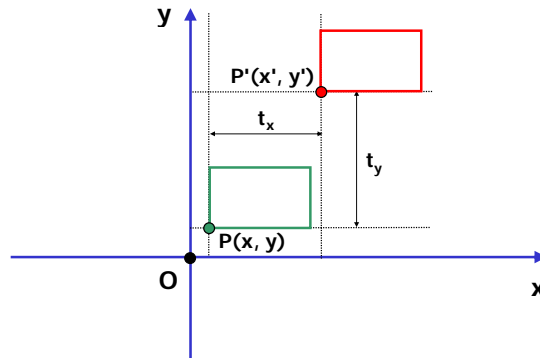


Figure 21. Translation

Given the coordinates of a point $P(x, y)$, the coordinates of the new point $P'(x', y')$ are calculated according to the following formula:

$$\begin{aligned}x' &= x + t_x \\y' &= y + t_y\end{aligned}$$

In matrix notation, we can write the translation of a point defined by its vector $\vec{P} = \begin{pmatrix} x \\ y \end{pmatrix}$

with a translation vector $\vec{t} = \begin{pmatrix} t_x \\ t_y \end{pmatrix}$ resulting in a new point $\vec{P}' = \begin{pmatrix} x' \\ y' \end{pmatrix}$ as $\vec{P}' = \vec{P} + \vec{t}$ or

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}.$$

7.3.1.2 Rotation

The rotation of a geometric figure in a two-dimensional coordinate system with the angle φ is shown in Figure 22.

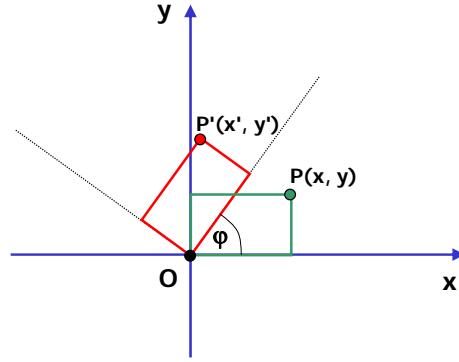


Figure 22. Rotation

Given the coordinates of a point $P(x, y)$, the coordinates of the rotated point $P'(x', y')$ are calculated according to the following formula:

$$\begin{aligned}x' &= x \cos \varphi - y \sin \varphi \\y' &= x \sin \varphi + y \cos \varphi\end{aligned}$$

In matrix notation the rotation of a point $\vec{P} = \begin{pmatrix} x \\ y \end{pmatrix}$ with an angle φ can be denoted as

$$\vec{P}' = R\vec{P} \text{ with the rotation matrix } R = \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} \text{ or } \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

7.3.1.3 Scaling

The scaling of a geometric figure can be described by the application of a multiplication factor (or scaling factor) to the coordinates in a given coordinate system (Figure 23).

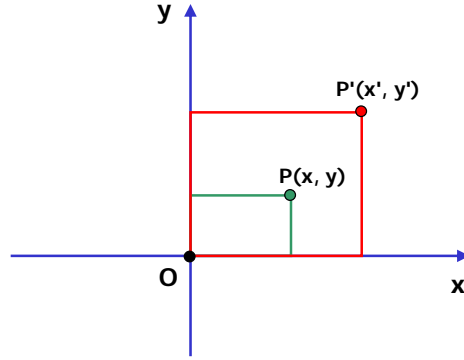


Figure 23. Scaling

Given the coordinates of a point $P(x, y)$, the coordinates of the scaled point $P'(x', y')$ are calculated according to the following formula:

$$\begin{aligned}x' &= s_x \cdot x \\y' &= s_y \cdot y\end{aligned}$$

The factors s_x and s_y are the scale factors in the x - and y -directions. They need not be the same.

In matrix notation the scaling of a point $\vec{P} = \begin{pmatrix} x \\ y \end{pmatrix}$ with scale factors s_x and s_y for x and

y , respectively, can be written as $\vec{P}' = S\vec{P}$ with the scaling matrix $S = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix}$ or

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

7.3.2 Combination of Transformations

When we want to perform rotation, scaling and translation to a point, either we can apply the respective transformations in sequence, one after the other, or we can combine the transformation matrices to one matrix for rotation and scaling and add the translation vector. The general approach using the matrices defined in the previous sections is written as

$$\vec{P}' = SR\vec{P} + \vec{t}$$

In the detailed notation this translates to

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x \cos \varphi & -s_x \sin \varphi \\ s_y \sin \varphi & s_y \cos \varphi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}.$$

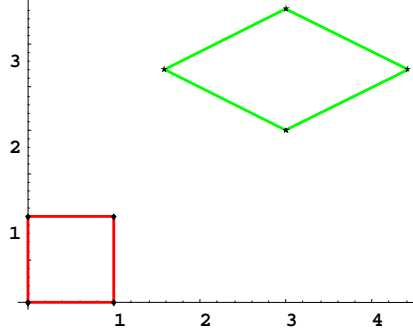
Example 60. Let S be a square defined with the points $P_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $P_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $P_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and

$P_4 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. We apply a rotation of 45 degrees, a scaling with the factor 2 in x -direction and 1 in y -direction. Finally, we shift the figure three units to the right and two units up. The result can be

computed according to the transformation matrix $T = \begin{pmatrix} \sqrt{2} & -\sqrt{2} \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \end{pmatrix}$ and the translation vector

$\vec{t} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$ as $P' = TP + \vec{t}$. The following figure shows the original square in red and the

transformed figure in green where $P'_1 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$, $P'_2 = \begin{pmatrix} 3+\sqrt{2} \\ 2+\frac{1}{\sqrt{2}} \end{pmatrix}$, $P'_3 = \begin{pmatrix} 3 \\ 2+\sqrt{2} \end{pmatrix}$ and $P'_4 = \begin{pmatrix} 3-\sqrt{2} \\ 2+\frac{1}{\sqrt{2}} \end{pmatrix}$.



7.3.3 Homogeneous Coordinates

An easier way to deal with geometric transformations is to use homogeneous coordinates.

Definition 37 (Homogeneous Coordinates). Every point with the Cartesian coordinates (x, y) can be assigned the homogeneous coordinates $(t \cdot x, t \cdot y, t)$. Conversely, given the homogeneous coordinates of a point (r, s, t) , we can determine its Cartesian coordinates as $x = \frac{r}{t}$ and $y = \frac{s}{t}$.

We assign to a point $P(x, y)$ its homogeneous coordinates $(x, y, 1)$. The geometric transformations can then be expressed by 3×3 -matrices.

$$R = \begin{pmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{rotation}$$

$$S = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{scaling}$$

$$T = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \quad \text{translation}$$

Note that now also the translation can be expressed through a translation matrix. This allows us to combine all three transformations in one single transformation matrix

$$U = \begin{pmatrix} s_x \cos \varphi & -s_x \sin \varphi & t_x \\ s_y \sin \varphi & s_y \cos \varphi & t_y \\ 0 & 0 & 1 \end{pmatrix}.$$

The general transformation of a point including rotation, scaling and translation can then be written simply as the multiplication of the transformation matrix with the point vector $\vec{P}' = U\vec{P}$ or

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x \cos \varphi & -s_x \sin \varphi & t_x \\ s_y \sin \varphi & s_y \cos \varphi & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Example 61. The transformation of the square in the previous example can now be written as

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \sqrt{2} & -\sqrt{2} & 3 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

7.3.4 Transformation between Coordinate Systems

In many applications, we have to transform coordinates from one system into another coordinate system. In principle, this relates to the geometric transformations discussed in the previous sections. However, often we do not know the rotation angle, scale factor or translation vector, or there are more distortions involved. In these cases, we must determine the transformation parameters from known coordinates of points in both systems. These points are called *control points*. The most common transformations used are

- similarity transformation
- affine transformation
- projective transformation

The *similarity transformation* (also called *HELMERT transformation*) scales, rotates, and translates the data. It does not independently scale the axes or introduce any skew. It is also known as *four-parameter transformation* and has the general form

$$\begin{aligned} x' &= Ax + By + C \\ y' &= -Bx + Ay + F \end{aligned}$$

A minimum of two control points is required to be able to compute the four parameters A , B , C , and F .

The *affine transformation* (or *6-parameter transformation*) will differentially scale, skew, rotate, and translate the data. It requires a minimum of three control points and has the general form

$$\begin{aligned} x' &= Ax + By + C \\ y' &= Dx + Ey + F \end{aligned}$$

Finally, the *projective transformation* (or *8-parameter transformation*) can compensate for greater distortions between the two coordinate systems and requires a minimum of four control points. It has the general form

$$\begin{aligned} x' &= \frac{Ax + By + C}{Gx + Hy + 1} \\ y' &= \frac{Dx + Ey + F}{Gx + Hy + 1} \end{aligned}$$

Example 62. Given two control points P_1 and P_2 , compute the parameters of a HELMERT transformation. The coordinates of the control points are measured (digitized) on a digitizing

device as $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$. The map coordinates of the ticks are given as $P'_1(x'_1, y'_1)$ and $P'_2(x'_2, y'_2)$. We can now compute the parameters of the HELMERT transformation by solving the following system of linear equations for A, B, C and F .

$$x'_1 = Ax_1 + By_1 + C$$

$$y'_1 = -Bx_1 + Ay_1 + F$$

$$x'_2 = Ax_2 + By_2 + C$$

$$y'_2 = -Bx_2 + Ay_2 + F$$

Normally, we use more than the required minimum number of control points. We then need to solve the resulting system of equations with a least squares approach. The *root mean square error* (RMS) indicates the goodness of fit. Ideally, the best fit would result in a RMS of zero, which is never the case when we use more than the required number of control points. However, the RMS should be as small as possible to achieve a reliable set of parameters for the transformation.

7.4 Applications in GIS

In GIS, we apply geometric transformations in many different ways. One use of transformations is in the graphic editing functions of every GIS. When we edit spatial features, we need to shift, rotate, skew and scale them.

Another important application lies in the transformation of coordinates of datasets as it occurs, for instance, in manual digitizing. Here, we have to set up a transformation from the device coordinates, i.e., the coordinates produced by the digitizing device – usually in millimeters or inches – to the world coordinates, i.e., the coordinates of the map projection.

Figure 24 shows a sketch of a digitizing tablet with a map mounted on it. The origin of the tablet coordinates lies at O_t . The map coordinate system is indicated with O_k and the axes x_k and y_k . On the map we have identified four ticks (or control points) designated with \oplus . The map coordinates of these ticks are either known or can be determined easily, for instance as grid intersection points or corner points of the map sheet whose coordinates can be read from the map.

The map usually is not aligned with the coordinate system of the tablet. Before we can start digitizing, we need to establish a relationship between the tablet coordinate and the map coordinate system. This is done by choosing a proper transformation and by computing its parameters. Usually, we select a four- or six-parameter transformation. With the given coordinates of the ticks (in the map coordinate system) and their measured coordinates (in the tablet coordinate system), we can compute the transformation parameters and subsequently apply the transformation to every point measured on the map. The transformation converts these coordinates into map coordinates.

Every GIS software package should provide this functionality for manual digitizing or general coordinate transformation. The `TRANSFORM` command in Workstation Arc/INFO is one example of such a function.

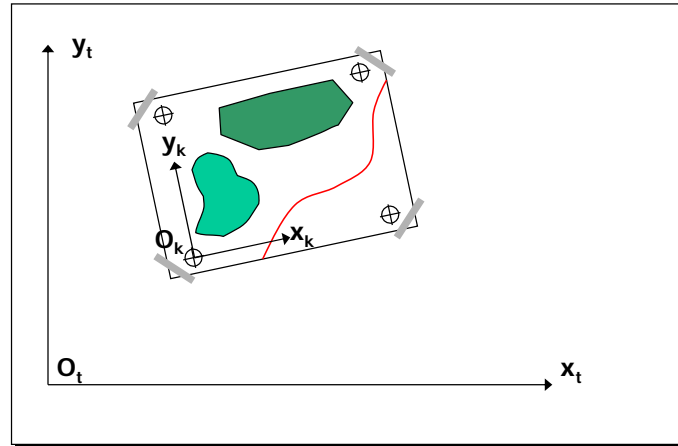


Figure 24. Manual digitizing setup

7.5 Exercises

- Exercise 24* Given the geographic coordinates of Vienna airport as $\varphi = 48.12$, $\lambda = 16.57$ and the radius of the Earth as $R = 6370\text{km}$, compute the Cartesian coordinates of the airport when the origin of the Cartesian coordinate system is located at the center of the earth and the axes directions are as illustrated in Figure 17.
- Exercise 25* Explain why each of the following expressions makes no sense when the operation “ \cdot ” denotes the dot product of vectors: (a) $\vec{a} \cdot (\vec{b} \cdot \vec{c})$, (b) $\vec{a} + (\vec{b} \cdot \vec{c})$, (c) $k \cdot (\vec{a} \cdot \vec{b})$.
- Exercise 26* Let $\vec{a} = (1, 3, 2)$, $\vec{b} = (1, 2, 3)$ and $\vec{c} = (2, -1, 4)$. Compute (a) $\vec{b} \times \vec{c}$, (b) $(\vec{a} \times \vec{b}) - 2\vec{c}$, and (c) $(\vec{a} \times \vec{b}) \times (\vec{b} \times \vec{c})$.
- Exercise 27* What is wrong with the expression $\vec{a} \times \vec{b} \times \vec{c}$?
- Exercise 28* Let $\vec{a} = (1, 3, 2)$, $\vec{b} = (1, 2, 3)$ and $\vec{c} = (2, -1, 4)$. Compute $(\vec{a}, \vec{b}, \vec{c})$.
- Exercise 29* Given the triangle $\triangle ABC$ with the coordinates $A(1, 0)$, $B(3, 0)$ and $C(2, 1)$ compute the coordinates of the resulting figure when the triangle is rotated by 60° , scaled with a factor of 1.5 in both x - and y -direction, and translated 2 units to the right and 3 units down.
- Exercise 30* Use a map sheet of your choice and perform the procedure for setting up the transformation parameters in manual digitizing using four control points and a six-parameter transformation.

Algebraic Structures

Mathematical structures are used to describe real world processes or phenomena. As we have seen before, there are three main structures in mathematics: algebras, topological structures, and order structures. In this chapter we will describe algebraic structures (or algebras) that are characterized by a set on which operations are defined for the elements of this set.

These operations are needed when we want to “compute” or “calculate” with the elements of a set. Often we need to identify mappings from one algebra to another. If the structure is preserved under such a mapping, we call it a structure preserving mapping or homomorphism.

8.1 Components of an Algebra

Definition 38 (Algebra). Whenever we specify an algebra we need to describe the following components:

- A set S , the *carrier* of the algebra.
- *Operations* defined on the elements of the carrier, and
- Distinguished elements of the carrier, the *constants* of the algebra

The carrier S of an algebra is a set of elements on which operations are defined. Examples of carriers are number sets such as \mathbb{N} (natural numbers), \mathbb{Z} (integers), or \mathbb{R} (real numbers). Operations are defined as a mapping $\circ: S^m \rightarrow S$ where the m is called the “arity” of the operation. Operations from $S^1 = S \rightarrow S$ are called *unary* operations. As an example of a unary operation consider the operation “ $-$ ” that assigns the negative value to an element, i.e., it takes the number x to $-x$. *Binary* operations are mappings from $S^2 \rightarrow S$ and operate on two elements of the carrier. Examples of a binary operation are addition $x + y$ and multiplication $x \cdot y$ of elements. The constants of an algebra are distinguished elements of the carrier set with properties of special importance. Algebras are denoted as n -tuples $\langle \text{carrier, operations, constants} \rangle$.

Example 63. The real numbers \mathbb{R} with the binary operations $(+)$ (addition) and multiplication (\cdot) , the unary operation $(-)$ and the constants 0 and 1 are an algebra that is represented as the 6-tuple $\langle \mathbb{R}, +, \cdot, -, 0, 1 \rangle$.

8.1.1 Signature and Variety

Often we look at a class of algebras such that every member of the class has the same characteristics.

Definition 39 (Signature of an algebra). Two algebras have the same *signature* (or are of the same *species*) if their n -tuples include the same number of operations and constants and the arities of corresponding operations are the same.

Example 64. The algebras $\langle \mathbb{R}, +, \cdot, 1, 0 \rangle$ and $\langle \wp(S), \cup, \cap, S, \emptyset \rangle$ have the same signature because they possess two binary operations $(+)$ and (\cdot) and (\cup) and (\cap) and two constants (1) and (0) and (S) and (\emptyset) , respectively.

Example 65. The algebras $\langle \mathbb{Z}, +, 0 \rangle$ and $\langle \mathbb{Z}, + \rangle$ are not of the same species, because the number of constants is not the same. The second algebra does not possess any constants.

Algebras that have the same signature need not be related at all. In order to be able to distinguish different classes of algebras that “behave” in the same way, we need certain rules that are valid for the elements of the carrier. Such “rules” are called *axioms* and are written as equations of elements of the carrier.

Definition 40 (Variety). A set of axioms for the elements of the carrier, together with a signature, specifies a class of algebras called *variety*.

Algebras that belong to the same variety behave in exactly the same way. Although the carrier set, operations and constants may be different, all algebras of the same variety obey the same axioms. In the following sections, we will discuss some of the more important varieties of algebras.

Example 66. Consider the variety of algebras with the same signature $\langle \mathbb{R}, +, 0 \rangle$ and the following axioms: (i) $x + y = y + x$, (ii) $(x + y) + z = x + (y + z)$, and (iii) $x + 0 = 0 + x = x$. Then $\langle \mathbb{Z}, +, 0 \rangle$, $\langle \wp(S), \cup, \emptyset \rangle$, $\langle \wp(S), \cap, S \rangle$, and $\langle \mathbb{Z}, \cdot, 1 \rangle$ are all members of this variety, where the binary operations are denoted as “+”, “ \cup ”, “ \cap ”, and “ \cdot ”, and the constants are “0”, “ \emptyset ”, “ S ”, and “1”, respectively. Any theorem proven for this variety will hold for all algebras that belong to this variety.

For the remainder of this chapter, whenever we deal with an arbitrary algebra A , we will use the following notation $A = \langle S, \circ, \Delta, k \rangle$, where S is the carrier, \circ denotes a binary operation, Δ a unary operation, and k a constant.

8.1.2 Identity and Zero Elements

Constants possess special properties relative to one or more operations in an algebra. The following definition describes the most important properties of constants for binary operations.

Definition 41 (Identity and Zero Element). Let \circ be a binary operation on S . An element $1 \in S$ is an *identity* (or *unit*) for the operation \circ if $\forall x \in S, 1 \circ x = x \circ 1 = x$. An element $0 \in S$ is a *zero* for the operation \circ if $\forall x \in S, 0 \circ x = x \circ 0 = 0$. If no confusion can result we may not specify the operation and speak of an *identity* (or *identity element*) and a *zero* (or *zero element*).

Example 67. The algebra $\langle \mathbb{Z}, \cdot, 1, 0 \rangle$ with the multiplication as operation has an identity 1 and a zero 0.

Example 68. The algebra $\langle \mathbb{Z}, +, 0 \rangle$ has an identity 0 but no zero element.

If identities exist, we can define the inverse with respect to an operation.

Let \circ be a binary operation on S , and 1 an identity for this operation. If $x \circ y = y \circ x = 1$ for every y in S , then x is called (two-sided) *inverse* of y with respect to the operation \circ .

Example 69. The algebra $\langle \mathbb{Z}, +, 0 \rangle$ has an identity 0 and every element $x \in \mathbb{Z}$ has an inverse with respect to the addition. The inverse of x is written as $-x$ and $x + (-x) = 0$.

Example 70. The algebra $\langle \mathbb{R}, \cdot, 1 \rangle$ has an identity 1 and all elements x of the real numbers except 0 have an inverse $x^{-1} = \frac{1}{x}$ such that $x \cdot \frac{1}{x} = 1$.

8.2 Varieties of Algebras

Algebras play an important role in many applications of computer science such as formal languages and automata theory as well as in coding theory and switching theory. In spatial analysis map algebra, i.e., operations on (usually raster) data sets, is very common. In this section, we will discuss a few algebras of importance.

8.2.1 Group

Many algebraic structures are the basis of arithmetic, as we usually know it for numbers (integers, rational and real numbers). One basic structure is a group that formalizes the arithmetic of one binary operation (usually addition or multiplication for number sets).

Definition 42 (Group). A *group* is an algebra with the signature $\langle S, \circ, ^{-1}, 1 \rangle$ with one binary (\circ) and one unary ($^{-1}$) operation, where $^{-1}$ is the inverse with respect to \circ , and the following axioms:

1. $a \circ (b \circ c) = (a \circ b) \circ c$
2. $a \circ 1 = 1 \circ a = a$
3. $a \circ \bar{a} = 1$

If the operation \circ is also commutative, then we call the group a *commutative group* (or *Abelian group*).

Example 71. The algebra $\langle \mathbb{Z}, +, -, 0 \rangle$ is a commutative group where \mathbb{Z} are the integers, $+$ the usual addition, $-$ the inverse (negative number) with regard to the addition, and 0 the identity for the addition. The axioms can be easily verified as:

1. $a + (b + c) = (a + b) + c$
2. $a + 0 = 0 + a = a$
3. $a + (-a) = 0$
4. $a + b = b + a$

Example 72. The algebra $\langle \mathbb{Z} - \{0\}, \bullet, ^{-1}, 1 \rangle$ is a commutative group where \mathbb{R} are the real numbers, \bullet is the usual multiplication, $^{-1}$ the inverse with regard to the multiplication, and 1 the identity for the multiplication. The axioms are verified as follows:

1. $a \bullet (b \bullet c) = (a \bullet b) \bullet c$
2. $a \bullet 1 = 1 \bullet a = a$
3. $a \bullet a^{-1} = 1$
4. $a \bullet b = b \bullet a$

Example 73. The natural numbers \mathbb{N} with addition and multiplication are not a group because there is no inverse with regard to addition and multiplication.

8.2.2 Field

Fields are very general algebras that formally describe the interrelation of two binary operations on a carrier set. Simply speaking, a field guarantees all arithmetic operations (as we know them for instance from the usual number sets) without restrictions (except division by zero).

Definition 43 (Field). A *field* is an algebra with the signature $\langle S, +, \circ, ^{-1}, ^{-1}, 0, 1 \rangle$ where $^{-1}$ and $^{-1}$ are the inverse operations for $+$ and \circ , respectively; and the following axioms:

1. $\langle S, +, ^{-1}, 0 \rangle$ is a commutative group
2. $a \circ (b \circ c) = (a \circ b) \circ c$
3. $a \circ (b + c) = a \circ b + a \circ c$
4. $(a + b) \circ c = a \circ c + b \circ c$

5. $\langle S - \{0\}, \circ, ^{-1}, 1 \rangle$ is a commutative group

Example 74. The real numbers $\langle \mathbb{R}, +, \bullet, ^{-1}, 0, 1 \rangle$ are a field with addition and multiplication as binary operations, and the inverse unary operations for addition and multiplication. The numbers 0 and 1 function as identity elements for $+$ and \bullet , respectively. The axioms are verified as

1. See Example 71
2. $a \bullet (b \bullet c) = (a \bullet b) \bullet c$ (associative law of multiplication)
3. $a \bullet (b + c) = a \bullet b + a \bullet c$ (distributive law)
4. $(a + b) \bullet c = a \bullet c + b \bullet c$ (distributive law)
5. See Example 72

8.2.3 Boolean Algebra

Definition 44 (Boolean Algebra). A *Boolean algebra* has a signature $\langle S, +, \circ, ^{-}, 0, 1 \rangle$ where $+$ and \circ are binary operations, and $^{-}$ is a unary operation (the *complementation*), with the axioms:

1. $a + b = b + a$
2. $a \circ b = b \circ a$
3. $(a + b) + c = a + (b + c)$
4. $(a \circ b) \circ c = a \circ (b \circ c)$
5. $a \circ (b + c) = a \circ b + a \circ c$
6. $a + (b \circ c) = (a + b) \circ (a + c)$
7. $a + 0 = a$
8. $a \circ 1 = a$
9. $a + \bar{a} = 1$
10. $a \circ \bar{a} = 0$

Example 75. The power set $\wp(A)$ of a given set A with the usual set operations of union, intersection and complement relative to A is a Boolean algebra $\langle \wp(A), \cup, \cap, ^{-}, \emptyset, A \rangle$. Let X , Y and Z be arbitrary subsets of A (i.e., elements of the power set of A) then the axioms can easily be verified as

1. $X \cup Y = Y \cup X$
2. $X \cap Y = Y \cap X$
3. $(X \cup Y) \cup Z = X \cup (Y \cup Z)$
4. $(X \cap Y) \cap Z = X \cap (Y \cap Z)$
5. $X \cap (Y \cup Z) = (X \cap Y) \cup (X \cap Z)$
6. $X \cup (Y \cap Z) = (X \cup Y) \cap (X \cup Z)$
7. $X \cup \emptyset = X$
8. $X \cap A = X$
9. $X \cup \bar{X} = A$
10. $X \cap \bar{X} = \emptyset$

8.2.4 Vector Space

Some algebraic structures are defined on more than one set. Vector spaces are one example.

Definition 45 (Vector Space). Let $\langle V, +, -, 0 \rangle$ be a commutative group and $\langle S, +, \cdot, -, ^{-1}, 0, 1 \rangle$ a field. V is called a *vector space* over S , if for all $a, b \in V$ and $\alpha, \beta \in S$

1. $\alpha \bullet (a + b) = \alpha \bullet a + \alpha \bullet b$
2. $(\alpha + \beta) \bullet a = \alpha \bullet a + \beta \bullet a$
3. $(\alpha \bullet \beta) \bullet a = \alpha \bullet (\beta \bullet a)$
4. $1 \bullet a = a$

The elements of V are called *vectors*; the elements of S are called *scalars*.

Example 76. The set of all vectors with $+$ as the vector addition is a vector space over the real numbers where \bullet is the multiplication of a vector with a scalar.

Example 77. The set of all matrices with the matrix addition is a vector space over the real numbers with \bullet being the multiplication of a matrix with a scalar.

Vector spaces play an important role in the mathematical discipline of linear algebra, a sub-discipline of algebra.

8.3 Homomorphism

Often we need to compare algebras to find out whether they are similar. If two algebras are similar they show the same “behavior” in terms of their operations and they have corresponding constants. Often we know an algebra very well; i.e., we have established theorems for this algebra. If we can show that a different algebra is related to the given algebra (usually we want to show that they are essentially the same in terms of their behavior), then the same theorems (in a related way) also hold for the new algebra.

A formal way of investigating related algebras is to establish a structure-preserving mapping from the (given) algebra to the new algebra. Such a mapping is called a homomorphism.

Definition 46 (Homomorphism and Isomorphism). Let $A = \langle S, \circ, \Delta, k \rangle$ and $A' = \langle S', \circ', \Delta', k' \rangle$ be algebras with the same signature, and let h be a function such that

1. $h: S \rightarrow S'$
2. $h(a \circ b) = h(a) \circ' h(b)$
3. $h(\Delta(a)) = \Delta'(h(a))$
4. $h(k) = k'$

Then h is called a *homomorphism* from A to A' . If the function h is bijective then we call it an *isomorphism* from A to A' , and A' is an isomorphic image of A under the map h .

In the definition above \circ and \circ' represent binary operations, Δ and Δ' unary operations, and k and k' are constants.

Two algebras that are isomorphic are essentially the same algebra with different names. A homomorphic image of an algebra is a “smaller” or “generalized” version of the given algebra.

Example 78. Let S be a non-empty set and two $A = \langle \wp(S), \cup, \cap, ^c, \emptyset, S \rangle$ and $B = \langle \{0,1\}, +, \cdot, ^c, 0, 1 \rangle$ be two Boolean algebras. For any $a \in S$ and $T \in \wp(S)$ the function $h: \wp(S) \rightarrow \{0,1\}$ defined as $h(T) = \begin{cases} 0 & \text{if } a \notin T \\ 1 & \text{if } a \in T \end{cases}$ is a homomorphism from A to B . Note that $h(\emptyset) = 0$ and $h(S) = 1$.

Example 79. Let \mathbb{R}^+ be the set of all positive real numbers. Then $\langle \mathbb{R}^+, \cdot, 1 \rangle$ is isomorphic to $\langle \mathbb{R}, +, 0 \rangle$ and the function $h: \mathbb{R}^+ \rightarrow \mathbb{R}$ with $h(x) = \log x$ is an isomorphism. The function h is surjective, because for $x > 0$ the equation $\log_a x = y$ always has a solution $x = a^y$. The logarithmic function is monotone increasing; therefore h is injective. Furthermore, $h(a \cdot b) = \log(a \cdot b) = \log(a) + \log(b) = h(a) + h(b)$ and $h(1) = \log(1) = 0$. The given isomorphism is the mathematical basis for the slide rule that replaces multiplication of numbers by addition of their logarithms.

8.4 Applications in GIS

Perhaps the most prominent application of algebras in GIS is the *map algebra*. The carrier set of the map algebra is the set of “maps”, i.e., data sets that are often referred to by coverage, shapefile, grid or layer.

We know many operations for manipulating maps. They range from simple arithmetic operations of addition, subtraction, multiplication or division to more complex operations of calculating slope, aspect or hillshade.

An example of a constant of the map algebra would be the zero grid, where every grid cell carries the value zero. Figure 25 shows the expanded user interface of the ArcMap Spatial Analyst raster calculator. Here, we see various arithmetic and logical operators as well as functions that can be applied to map layers.

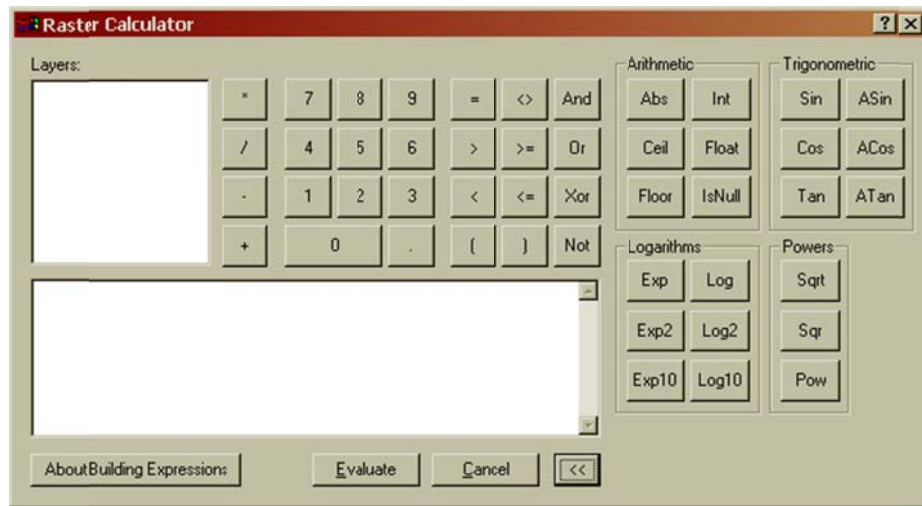


Figure 25. Raster calculator interface

The concept of structure-preserving mapping finds an application in spatial modeling, where we map a subset of the real world to a representation in a spatial feature model. We will return to this subject in chapter 13.

8.5 Exercises

Exercise 31 Given are two algebras: $\langle \mathbb{Z}, +, -, 0 \rangle$ with the integers as carrier set, two operations, addition (+) and negative number (-), and the constant 0; $\langle \mathbb{E}, +, -, 0 \rangle$ with the even numbers as carrier set, the constant 0 and the operations (+) and (-) defined in the usual way (addition and negative number, respectively). Show that both algebras are Abelian groups and that the function $f: \mathbb{Z} \rightarrow \mathbb{E}$, defined as $f(x) = 2x$, is an isomorphism.

Exercise 32 Let $\{T, F\}$ be a set where T stands for “true” and F stands for “false”. Show that $\langle \{T, F\}, \vee, \wedge, \neg, F, T \rangle$ is a Boolean algebra with the binary operations “and” (\wedge), “or” (\vee), and the unary operation “not” (\neg) being logical operators. The constants F and T are propositions that are always false (F) and always true (T), respectively.

Exercise 33 Given are two algebras: $\langle \mathbb{Z}, +, -, 0 \rangle$ with the integers as carrier set, two operations, addition (+) and negative number (-), and the constant 0; and $\langle B, +, -, 0 \rangle$ with the carrier set $B = \{0, 1\}$, the

constant 0 and two operations defined as


+	0	1
0	0	1
1	1	0

 and $(-x) = x$. Show that the function

$f: \mathbb{Z} \rightarrow B$, defined as $f(x) = \begin{cases} 0 & x \text{ is even} \\ 1 & x \text{ is odd} \end{cases}$, is a homomorphism. Why is f not an isomorphism?

Topology is a central concept in every GIS. It deals with the structural representation of spatial features and their properties that remain invariant under certain transformations. In this chapter, we introduce the mathematical concept of a topological space based on the topology that is induced on the real plane by a distance function.

We also show how simple structures can be used to build complex objects in GIS databases, and how to check the consistency of a two-dimensional topologic representation of spatial features.



9.1 Topological Spaces

In this chapter, we will deal with topological spaces, i.e., a set and a collection of subsets of this set that satisfy certain conditions. There are two equivalent approaches to define a topological space. The first one starts with the concept of a neighborhood of a point and defines a topological space as a system of neighborhoods that fulfill certain conditions. The concept of open sets follows from the definition of a neighborhood. The second approach starts from a family of subsets of a given set (which are called open sets) and defines a topology through properties of these open sets. The concept of a neighborhood follows from the definition of a topological space.

9.1.1 Metric Spaces and Neighborhoods

The first approach is more intuitive than the second one that is usually used in general topology (or point-set topology). For our purpose, we chose the intuitive approach on the Euclidean plane with a system of neighborhoods. In order to define a neighborhood we need the concept of a distance. Generally, this can be achieved with a metric space.

Definition 47 (Metric Space). Let X be a nonempty set and d a function $X \times X \rightarrow \mathbb{R}_0^+$ such that for every $x, y, z \in X$

- (i) $d(x, y) = 0$ if and only if $x = y$
- (ii) $d(x, y) = d(y, x)$
- (iii) $d(x, y) + d(y, z) \geq d(x, z)$ (triangle inequality)

We call the pair (X, d) a *metric space* and d a *distance function* (or *metric*) on X .

Example 80. Let us consider the real plane \mathbb{R}^2 equipped with the *Euclidean distance* $d_E(p, q) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$ between two points $p = (a_1, a_2)$ and $q = (b_1, b_2)$. We call this space the 2-dimensional *Euclidean space*. (\mathbb{R}^2, d_E) is a metric space. The Euclidean distance is the shortest distance between two points. This is the usual space of plane geometry. We can easily extend this space to three dimensions.

Example 81. The real numbers \mathbb{R} with the distance function $d(x, y) = |x - y|$ are a metric space.

In every metric space, we can define a neighborhood for points of this space.

Definition 48 (ε -neighborhood). In a metric space (X, d) , for each $x \in X$ and each $\varepsilon > 0$, we define an (open) ε -neighborhood of x as the set $N(x, \varepsilon) = \{y \mid y \in X \wedge d(x, y) < \varepsilon\}$. When no confusion is possible we call $N(x, \varepsilon)$ neighborhood and write $N(x)$.

The set $\mathcal{N}_d(x) = \{N(x, \varepsilon) \mid x \in X \wedge \varepsilon > 0\}$ is called the *neighborhood system* of x induced by the metric d . In short we will write $\mathcal{N}(x)$.

In the Euclidean plane \mathbb{R}^2 an *open disk* with radius ε around a point p is an ε -neighborhood (Figure 26).

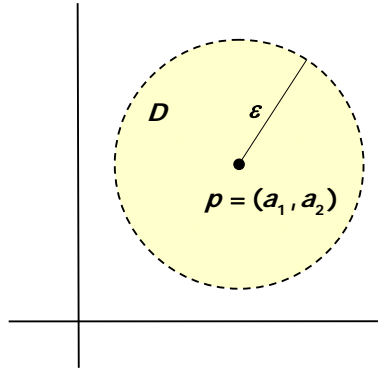


Figure 26. Open disk in \mathbb{R}^2

9.1.2 Topology and Open Sets

When we require a neighborhood system to satisfy certain conditions, we arrive at the definition of a topological space.

Definition 49 (Topological space). Let X be a set and for every $x \in X$ there exists a neighborhood system $\mathcal{N}(x) \subseteq \wp(X)$ that satisfies the following conditions (neighborhood axioms):

- (N1) The point x lies in each of its neighborhoods.
- (N2) The intersection of two neighborhoods of x is itself a neighborhood.
- (N3) Every superset U of a neighborhood N of x is a neighborhood of x . X is a neighborhood of x .
- (N4) Every neighborhood N of x contains a neighborhood V of x such that N is a neighborhood of every point of V .

We then call the set with its neighborhood system $(X, \mathcal{N}(x))$ a *topological space*. Sometimes we denote a topological space simply by X .

Figure 27 illustrates the four neighborhood axioms. With the help of the concept of a neighborhood, we can now define open sets.

Definition 50 (Open set). Let X be a topological space. A subset O of X is an *open set* if it is a neighborhood for each of its points.

Example 82. The open intervals (a, b) in the real numbers and the open disks in \mathbb{R}^2 are open sets.

Definition 51 (Closed set). Let X be a topological space. A set C is *closed* if its complement $X - C$ is open.

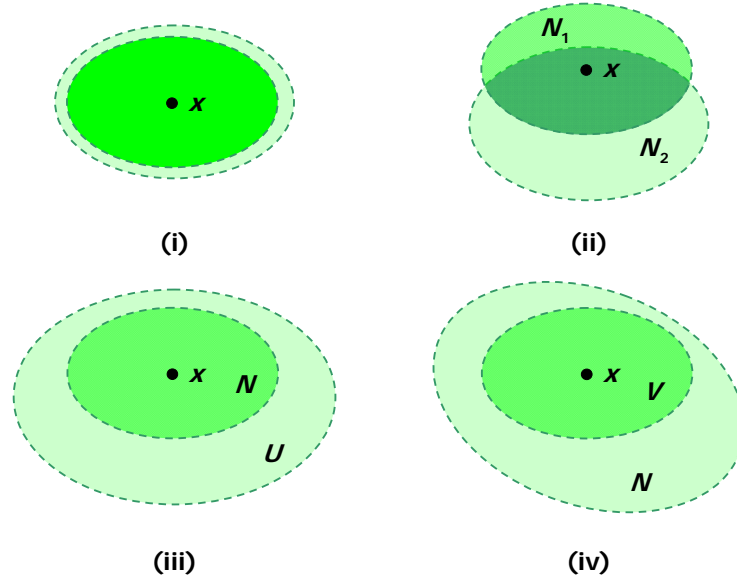


Figure 27. Neighborhood axioms

Example 83. The closed interval $[a, b]$ in the real numbers is a closed set, because its complement $(-\infty, a) \cup (b, \infty)$ is the union of two open sets, which again is an open set.

Example 84. The half-open interval $(a, b]$ in the real numbers is neither open nor closed.

The previous example shows that “closed” does not mean “not open”. Sets can be neither open nor closed, or they can be both open and closed (sometimes called *clopen* sets). The following statements can be proven to be true for open sets.

- (O1) The empty set \emptyset and the set X are open.
- (O2) The intersection of any finite number of open sets is open.
- (O3) The union of any number of open sets is open.
- (O4) A subset U of X is a neighborhood of $x \in X$ if and only if there exists an open set O with $x \in O \subseteq U$.

The intersection of an arbitrary number of open sets does not need to be open. Take for example the intersection of an infinite collection of open intervals

$$\left(-\frac{1}{n}, +\frac{1}{n}\right) \text{ for } n=1, 2, 3, \dots$$

Obviously, the intersection is the set $\{0\}$ which is not an open set.

It can be proven that the intersection of an arbitrary number of closed sets, and the union of a finite number of closed sets, are closed.

Our approach to the definition of a topological space is based on the concept of the ε -neighborhood defined in a metric space. For this definition we needed the distance, a concept which is too special for general topological spaces. Statement (O4) above gives us a way to define neighborhoods without the notion of distance. Here, we also see that a neighborhood does not need to be an open set; it can also be a closed set. As an example consider a point p of the Euclidean plane \mathbb{R}^2 . Every closed disk around p is a neighborhood of p , because it contains the open disk around p , which is an open set.

9.1.3 Continuous Functions and Homeomorphisms

We can define mappings between topological spaces. A function that maps the neighborhood of a point to the neighborhood of the image of this point is called a continuous function. We can define it formally as follows.

Definition 52 (Continuous function). Let $f: X \rightarrow Y$ be a function from the topological space X to the topological space Y . We call f *continuous* at point $x_0 \in X$ if for every neighborhood V of $f(x_0)$ there is a neighborhood U of x_0 such that the image of U , i.e., $f(U)$, is a subset of V . If f is continuous at every point of X , we call it a *continuous function*.

Figure 28 illustrates both concepts of continuity at a point and continuous function.

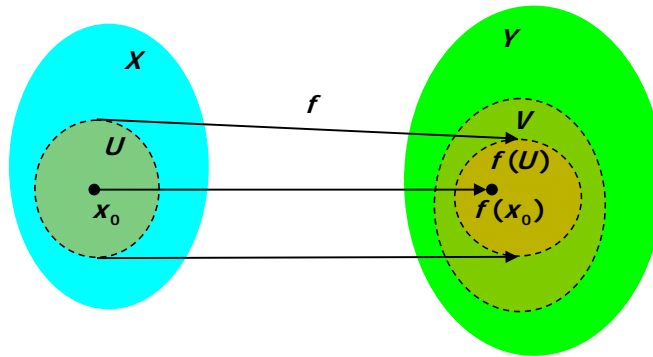


Figure 28. Continuous function

The definition above is valid for any two topological spaces. For the real numbers \mathbb{R} , the definition of continuity usually reduces to the following statement:

A function $f: \mathbb{R} \rightarrow \mathbb{R}$ is continuous at point x_0 if for every $\varepsilon > 0$ there exists a $\delta > 0$ such that $|x - x_0| < \delta$ implies $|f(x) - f(x_0)| < \varepsilon$. A function is continuous if it is continuous at every point. Continuity of a function essentially means that the graph of the function has no “jumps” or “gaps”.

Like in other mathematical structures, also for topological spaces we know structure-preserving mappings. They map one topological space to another topological space thereby preserving the topology.

Definition 53 (Homeomorphism). Let $h: X \rightarrow Y$ be a function from the topological space X to the topological space Y . If this function is continuous, bijective, and possesses a continuous inverse, we call it a *homeomorphism* (or *topological mapping*).

If two spaces are homeomorphic they are essentially the same and expose the same topological behavior.

Example 85. Let $X = (-1, 1)$ be an open interval in \mathbb{R} and $f: X \rightarrow \mathbb{R}$ a function defined as $f(x) = \tan \frac{\pi}{2} x$. This function is bijective, continuous and has a continuous inverse. Figure 29 shows the graph of the function. It is a homeomorphism. This means that the open interval $(-1, 1)$ and the real numbers are homeomorphic.

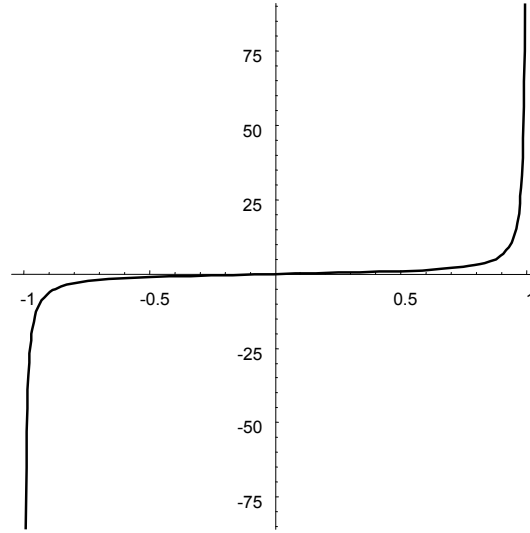


Figure 29. Example of a homeomorphic function

Example 86. The open disk $D_1 = \{(r, \theta) \mid r < 1\}$ given with its polar coordinates (r, θ) and radius 1 is homeomorphic to the open disk $D_2 = \{(r, \theta) \mid r < 2\}$ with radius 2 through the function $f((r, \theta)) = (2r, \theta)$.

A property of a topological space that is preserved by a homeomorphism is called a *topological property* or a *topological invariant*. Mathematical topology is mainly focused on properties of topological spaces that remain invariant under topological mappings.

The two previous examples show that length and area are not topological invariants, because the length of the interval $(-1, 1)$ is different from the “length” of the real line, and in the second example both disks are homeomorphic but their areas are not the same.

9.1.4 Alternate Definition of a Topological Space

As mentioned earlier, a different, yet equivalent, definition of a topological space starts with the idea of an open set, defines a topology as properties of a collection of open sets and derives the definition of a neighborhood from the open sets. We take the properties O1 to O3 of open sets as axioms and define a topological space as follows.

Definition 54 (Topological Space). Let X be a set and O a collection of subsets of X , i.e., $O \subseteq \wp(X)$. We call O a *topology* on X when the subsets satisfy the following three conditions:

$$(O1) \quad \emptyset \in O, X \in O$$

$$(O2) \quad A, B \in O \Rightarrow A \cap B \in O$$

$$(O3) \quad A_i \in O \Rightarrow \bigcup_{i \in I} A_i \in O$$

We call the O_i *open sets*, (X, O) a *topological space* and the elements $x \in X$ the *points* of the topological space.

The three conditions for a topology require that the empty set and the set itself must always be a member of the topology. Further, the intersection of a finite number of open

sets always is an open set, and the union of an arbitrary number of open sets is an open set.

With the help of the open sets used in the definition of a topology we can now define a neighborhood.

Definition 55. (Neighborhood). N is a *neighborhood* of the point x if $N \subseteq X$ and there exists an open set $A \in \mathcal{O}$ such that $x \in A \subseteq N$.

With this definition, we can prove the statements N1 to N4 of Definition 49 about neighborhoods to be true.

Example 87. Two extreme topologies can be found on any set X . The first one consists only of two elements $\{X, \emptyset\}$, the second one consists of all subsets of X , i.e., the power set $\wp(X)$. We can easily verify that the three conditions are satisfied for both topologies. The first topology is called *indiscrete topology*. It is the coarsest of all topologies, because it consists only of two elements. The second one is called *discrete topology*, which is the finest of all topologies.

Example 88. Consider the real line \mathbb{R}^1 . We call a subset A of \mathbb{R}^1 an open set if it is either empty or with each of its points $x \in A$ contains an open interval S_x that completely lies within A . All open intervals (a, b) on the real line \mathbb{R}^1 are open sets. The real line itself is an open set. Again, we can show that these open sets are a topology on \mathbb{R}^1 . We call it the *natural topology*. This can be extended to the \mathbb{R}^n with open disks, balls, etc.

Both approaches to the definition of a topological space as mentioned above are equally valid and lead to the same results. Figure 30 summarizes both approaches: the intuitive approach based on the concept of a neighborhood, and the set theoretic abstract approach based on the concept of open sets.

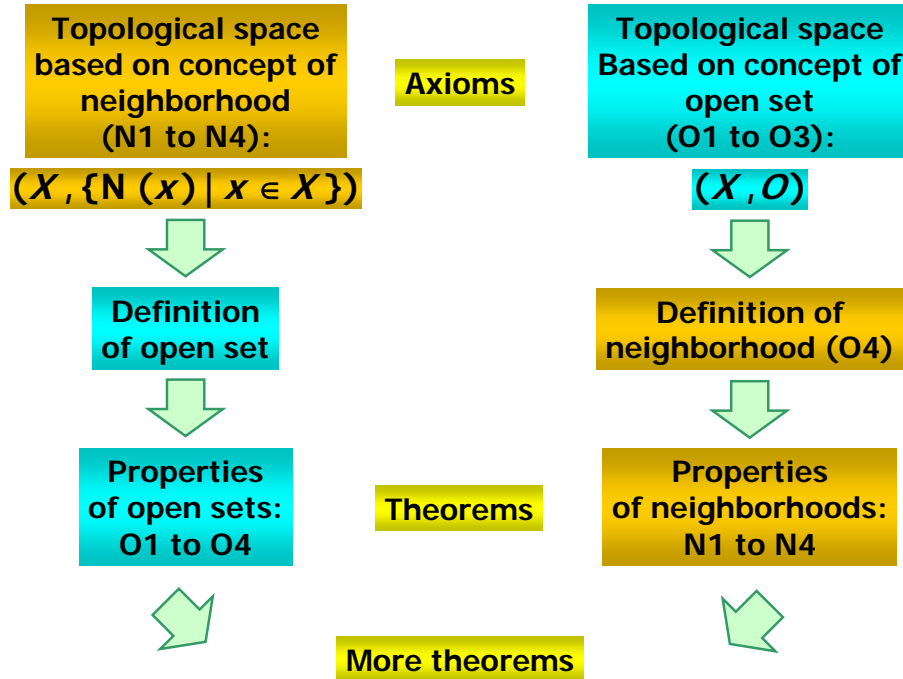


Figure 30. Equivalent approaches to the definition of a topological space, open sets and neighborhoods and related theorems

The first one defines a topological space through properties of neighborhoods. Open sets are then defined through neighborhoods, and the properties O1 to O4 follow as theorems.

The second approach defines a topological space through properties of open sets. It then defines neighborhoods through property O4 of open sets, and derives N1 to N4 as theorems about neighborhoods. More theorems about topological spaces follow from there.

9.2 Base, Interior, Closure, Boundary, and Exterior

From the definition of a topological space we know that the union of open sets is an open set. If in a topological space every open set can be generated as the union of some open sets, we call these sets a base of the topology.

Definition 56 (Base). Let X be a topological space and a collection \mathcal{B} of open sets such that every open set of the topology is the union of members of \mathcal{B} . Then \mathcal{B} is called a *base* for the topology and the elements of \mathcal{B} are called *basic open sets*.

An equivalent definition for a base requires that for every point $x \in X$ that belongs to an open set O there is always an element $B \in \mathcal{B}$ such that $x \in B \subset O$.

Example 89. The open disks in the Euclidean plane \mathbb{R}^2 are a base for the natural topology of the plane. This follows easily from the definition of a neighborhood. The number of the basic open sets is uncountable infinite.

Example 90. The open disks in the Euclidean plane \mathbb{R}^2 with radii and center coordinates being rational numbers are a base for the natural topology of the plane. Note that the number of the basic open sets is countably infinite.

Whereas a base for a topology is a global characteristic of a topological space, we can also define a local base at a point of a topological space. This is a local characteristic of a topological space determined only by the neighborhood of the point.

Definition 57 (Local base). A collection \mathcal{B} of neighborhoods of a point x of a topological space X is a *local base* at x if every neighborhood of x contains some member of \mathcal{B} .

Example 91. Consider the natural topology in the Euclidean plane \mathbb{R}^2 and a point x . The system of open disks \mathcal{B}_x with center x is a local base at x . This is true because for every open set O that contains x there is an open disk with center at x that is contained in O .

Example 92. Let x be a point of a metric space. The countably infinite set of ε -neighborhoods of x defined as $\{N(x, 1), N(x, \frac{1}{2}), N(x, \frac{1}{3}), \dots\}$ is a local base at x .

The relationship between a base of a topology and a local base at a point can be expressed in the following statement:

Let \mathcal{B} be the base for a topology and $x \in X$ a point of the topological space. Then the members of \mathcal{B} that contain x form a local base at x .

For further investigations, we need the concepts of interior, closure, boundary and exterior of a set.

Definition 58 (Interior, Closure, Boundary, Exterior). Given a subset A of a topological space X we define the interior, closure and boundary as follows:

- The union, of all open sets contained in A is called the *interior* of A (written as A°).
- The smallest closed set containing A is called the *closure* of A (written as \bar{A}), in other words it is the intersection of all closed sets containing A .¹³
- The *boundary* of set A is the intersection of the closure of A with the closure of its complement $X - A$. The boundary¹⁴ is written as ∂A .
- The *exterior* of a set A (written as A^-) is the interior of the complement of A , i.e., $A^- = (X - A)^\circ$

An open set is its own interior. A closed set is equal to its closure. The following table shows some properties of interior, closure, and boundary.

Table 11. Properties of interior, closure, and boundary of a set

Interior	Closure	Boundary
$A^\circ \subseteq A, (A^\circ)^\circ = A^\circ$	$A \subseteq \bar{A}, \bar{\bar{A}} = \bar{A}$	$\partial A = \bar{A} - A^\circ$
$A \subseteq B \Rightarrow A^\circ \subseteq B^\circ$	$A \subseteq B \Rightarrow \bar{A} \subseteq \bar{B}$	$\partial A = \bar{A} \cap (X - A^\circ)$
$(A \cap B)^\circ = A^\circ \cap B^\circ$	$\overline{A \cup B} = \bar{A} \cup \bar{B}$	$\partial A = \bar{A} \cap \overline{X - A}$
$\left(\bigcup_{i \in I} A_i\right)^\circ \supseteq \bigcup_{i \in I} A_i^\circ$	$\overline{\bigcup_{i \in I} A_i} \supseteq \bigcup_{i \in I} \bar{A}_i$	$\partial A = X - (A^\circ \cup (X - A^\circ))$
$\left(\bigcap_{i \in I} A_i\right)^\circ \subseteq \bigcap_{i \in I} A_i^\circ$	$\overline{\bigcap_{i \in I} A_i} \subseteq \bigcap_{i \in I} \bar{A}_i$	$\partial A = \partial(X - A)$

Example 93. Consider the set $X = \{a, b, c, d, e\}$, the topology O defined on X as $O = \{X, \emptyset, \{a\}, \{c, d\}, \{a, c, d\}, \{b, c, d, e\}\}$ and the subset $A = \{b, c, d\}$ of X . The interior of A is $A^\circ = \{c, d\}$, because the only open sets contained in A are $\{c, d\}$ and \emptyset whose union is $\{c, d\}$. The closure of A is $\bar{A} = \{b, c, d, e\}$, because among the closed sets¹⁵ of X , i.e., \emptyset , X , $\{b, c, d, e\}$, $\{a, b, e\}$, $\{b, e\}$, and $\{a\}$, the smallest one that contains A is $\{b, c, d, e\}$. The boundary of A is the difference of the closure with the interior, i.e., $\partial A = \bar{A} - A^\circ = \{b, c, d, e\} - \{c, d\} = \{b, e\}$. The exterior of A is the interior of the complement of A , i.e., $\{a, e\}^\circ$, which results to $\{a\}$.

Example 94. Let us consider an open subset A of the Euclidean plane \mathbb{R}^2 . Figure 31 illustrates the interior, boundary, closure and exterior of the set.

¹³ Note that we use for closure the same symbol as for the set complement. They are, however, not related to each other.

¹⁴ The boundary of a set is often denoted as *frontier* of a set.

¹⁵ The closed sets are the complements of the open sets.

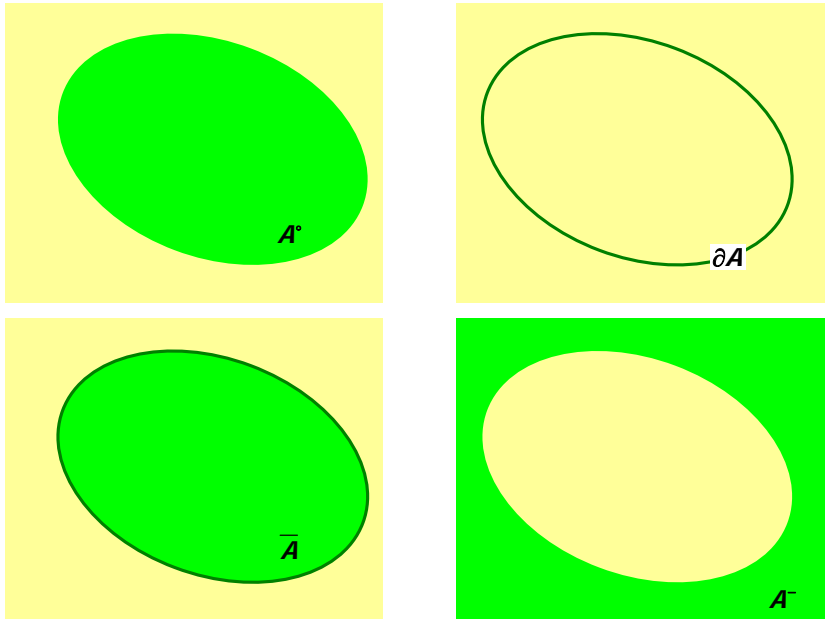


Figure 31. Interior (upper left), boundary (upper right), closure (lower left) and exterior (lower right) of an open set

9.3 Classification of Topological Spaces

There are several ways to classify topological spaces. Usually, this is done according to the degree to which their points are separated, regarding their compactness, their overall size, and their connectedness. Let us look at each of them in some detail.

9.3.1 Separation Axioms

In topology we know many different ways to distinguish disjoint sets and distinct points. We first present axioms that separate two distinct points.

Definition 59 (T_0 space). We call a topological space T_0 (or a T_0 space) if for two distinct points at least one has a neighborhood that does not contain the other point.

Definition 60 (T_1 space). We call a topological space T_1 (or a T_1 space) if two distinct points have neighborhoods that do not contain the other point.

Definition 61 (HAUSDORFF Space). A topological space X is called a *HAUSDORFF space* or T_2 space if two distinct points $a, b \in X$ possess disjoint open neighborhoods. In other words, there exist two open sets A and B with $a \in A$ and $b \in B$ and $A \cap B = \emptyset$.

Every metric space with the metric topology is T_2 . HAUSDORFF spaces are always T_1 , and every T_1 space is always T_0 .

Figure 32 illustrates separation axioms T_0 , T_1 , and T_2 .

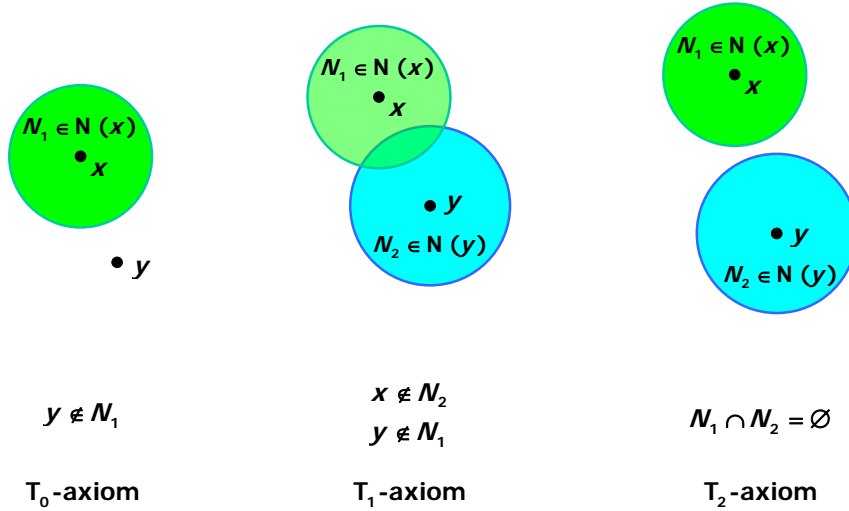


Figure 32. Separation axioms T_0 , T_1 , and T_2

We now look at axioms that separate sets. First we define an axiom that separates closed sets from the points that lie outside that set.

Definition 62 (Regular space). If a topological space is T_1 and for every closed set C and every point x outside C there exists an open set A that contains C and a disjoint neighborhood N of x , then we call this space a *regular space* or T_3 space.

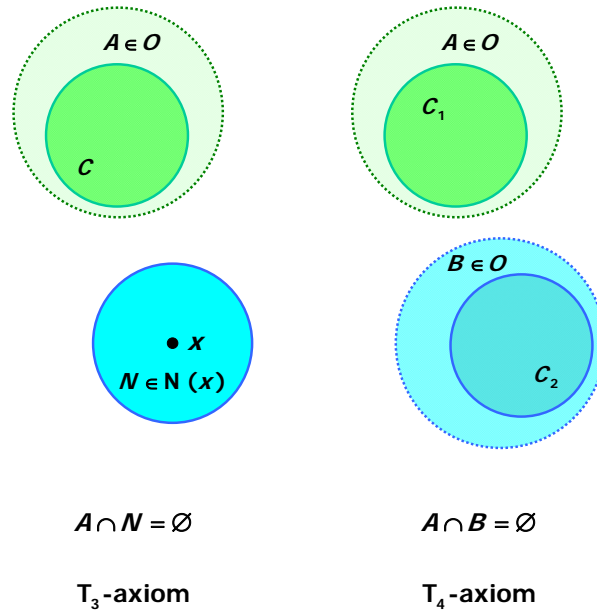
Every metric space with the metric topology is regular. Every regular space is a HAUSDORFF space. The converse is not true, because there are HAUSDORFF spaces that are not regular.

Finally, we introduce a separation axiom that separates closed sets.

Definition 63 (Normal space). If a topological space is T_1 and for any two disjoint closed sets C_1 and C_2 there exist disjoint neighborhoods that contain the closed sets, then we call this space a *normal space* or T_4 space.

Every metric space with the metric topology is normal, and every normal space is regular. The converse is not true, because there exist regular spaces that are not normal.

Figure 33 illustrates the axioms that lead to the definition of regular (T_3 and T_1 axioms) and normal spaces (T_4 and T_1 axioms).

Figure 33. Separation axioms T₃ and T₄

The following Figure 34 shows the relations between the separation characteristics of topological spaces.

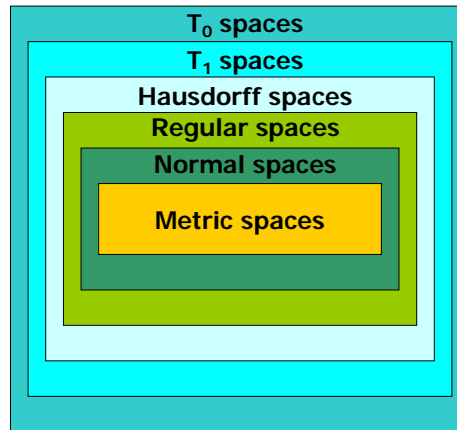


Figure 34. Relationship between separation characteristics of topological spaces

Separation characteristics are topological properties of a space, i.e., if a topological space X has a certain separation characteristic and there is a homeomorphism from X to a topological space Y , also Y will have the same characteristic. We see that a metric space is a very special case of a topological space that possesses all separation characteristics.

9.3.2 Compactness

In this section we discuss properties of topological spaces whose conditions are stronger than separation characteristics. These conditions are defined by what is called open covers. Finally, we will see that subsets of a Euclidean space which are both closed and bounded¹⁶ are of special importance.

¹⁶ A set is *bounded* when it is contained in some open ball with finite radius.

Definition 64 (Open cover). Let X be a topological space and \mathcal{F} a family of open subsets of X . If the union of these subsets is the whole space X , we call \mathcal{F} an *open cover* of X . If \mathcal{F}' is a subfamily of \mathcal{F} with $\bigcup \mathcal{F}' = X$, then \mathcal{F}' is a *subcover* of \mathcal{F} .

Example 95. Consider all open disks in the Euclidean plane \mathbb{R}^2 whose radius is 1 and their centers have integer coordinates. The union of these disks covers the whole space. Therefore, they are an open cover of \mathbb{R}^2 . If we leave out one disk, their union is not the whole space any more. Therefore, this family of open disks has no subcover.

If a topological space has a finite subcover, we call this space with a special name.

Definition 65 (Compact space). If every open cover of a topological space X has a finite subcover, we call X a *compact* space.

The following spaces are compact:

- The closed unit interval $[0, 1]$
- Any finite topological space
- A closed interval, disk or ball in \mathbb{R}^1 , \mathbb{R}^2 , or \mathbb{R}^3 , respectively

The following statements can be made about compact spaces. Their proofs can be found in the topological literature:

- (i) A continuous image of a compact space is compact.¹⁷
- (ii) A closed subset of a compact space is compact.
- (iii) A subset of the Euclidean n -space is compact if and only if it is closed and bounded.
- (iv) A compact HAUSDORFF space is normal.

The results from the previous section about separation and the last item from the previous list tell us that metric spaces as well as compact HAUSDORFF spaces are normal.

If we relax the condition for compactness to have a countable instead of a finite cover we arrive at the definition of a LINDELÖF space.

Definition 66 (LINDELÖF space). If every open cover of a topological space X has a countable subcover, we call X a *LINDELÖF* space (or we say that space X is LINDELÖF). Compact spaces are always LINDELÖF.

Example 96. The Euclidean plane \mathbb{R}^2 equipped with the natural topology of open disks is a LINDELÖF space.

Compactness is a topological property that remains invariant under homeomorphisms.

¹⁷ This proposition states that compactness as a topological property is even preserved under the weaker condition of a continuous mapping (and not a homeomorphism).

9.3.3 Size

A further characterization of topological spaces can be done according to their size. A measure for the size of a set is its cardinality, or number of elements. We recall from chapter 5 on set theory that a set is *countable* if it has either a finite number of elements or is countably infinite.

In order to proceed we need the definition of the term dense.

Definition 67 (Dense set). A subset A of a topological space X is *dense* if its closure is X , i.e., $\overline{A} = X$.

Example 97. The rational numbers \mathbb{Q} are a dense subset of the real numbers, because it can be shown that $\overline{\mathbb{Q}} = \mathbb{R}$. The rational numbers are countably infinite.

With both properties of a set to be countable and dense we can impose some limit on the size of a topological space which leads to the definition of a separable topological space.

Definition 68 (Separable space). A topological space is *separable* if it has a countable dense subset.

Example 98. The n -dimensional Euclidean space is separable.

Definition 69 (First-countable). A topological space is first-countable if every point has a countable local base.

Example 99. Every metric space is first-countable. According to Example 92 we have identified a countable local base for a metric space. It is therefore first-countable.

Example 100. Every discrete topological space is first-countable.

First-countable is a local property of a topological space, which is solely determined by the properties of the neighborhoods of a point. Another property of a topological space is related more to a global characteristic of space.

Definition 70 (Second-countable). A topological space is second-countable if it has a countable base for its topology.

Example 101. The Euclidean plane \mathbb{R}^2 is second-countable. According to Example 90 the open disks with rational radii and center coordinates are a countable base for \mathbb{R}^2 . Therefore, \mathbb{R}^2 is second-countable.

For topological spaces we can make the following statements with regard to their size characteristics:

If a topological space is second-countable, then it is also first-countable, separable, and LINDELÖF.

The properties of a topological space to be separable, first-countable, and second-countable are topological properties and remain invariant under homeomorphisms.

9.3.4 Connectedness

Connectedness of topological spaces deals with the property of such spaces that they cannot be divided into two disjoint nonempty open sets whose union is the entire space.

Definition 71 (Connected Space). A space X is *connected* if whenever it is represented as the union of two nonempty subsets $X = A \cup B$ then $\overline{A} \cap B \neq \emptyset$ or $A \cap \overline{B} \neq \emptyset$.

Intuitively speaking, a space is connected if it appears in one piece or it cannot be represented as the union of two disjoint open subsets. The following conditions on a topological space X are equivalent to formulate connectedness:

- (i) X is connected.
- (ii) The only subsets of X that are both open and closed are the empty set \emptyset and X .
- (iii) X cannot be expressed as the union of two disjoint nonempty open sets.

Example 102. The Euclidean space \mathbb{R}^n is connected, because the empty set and \mathbb{R}^n are the only sets that are both open and closed.

Example 103. Let $X = \{a, b, c, d, e\}$ be a set and $O = \{X, \emptyset, \{a\}, \{c, d\}, \{a, c, d\}, \{b, c, d, e\}\}$ a topology on X . Then X is not connected, because $\{a\}$ and $\{b, c, d, e\}$ are disjoint open sets and $X = \{a\} \cup \{b, c, d, e\}$ is the union of two disjoint nonempty open subsets.

A somewhat stronger condition can be stated when we consider how two points in a topological space can be connected.

Definition 72 (Path-connected space). A topological space X is *path-connected* if any two points $x_1, x_2 \in X$ of the space can be connected by a path. A *path* in a topological space X is a continuous function $f : [0, 1] \rightarrow X$ such that $f(0) = x_1$ (beginning point) and $f(1) = x_2$ (end point).

In general, every path-connected space is connected. The converse is not true. However, for regions¹⁸ of the Euclidean plane \mathbb{R}^2 with the natural topology we have the following result:

Every open connected subset of \mathbb{R}^2 is path-connected.

Connectedness is a topological property, i.e., it remains invariant under homeomorphisms. The image of a connected set under a continuous mapping is connected.

¹⁸ An open connected subset of a topological space is called a *region*.

9.4 Simplicial Complexes and Cell Complexes

The topological spaces we have treated so far are usually very general and too complex for many investigations. We, therefore, look for simpler spaces that can be used instead. These spaces can then be pieced together to form more complex spaces, yet keeping a recognizable shape and being easy to handle.

One class of these simple spaces is polyhedra. A polyhedron is a topological space that is built of simple building blocks, the simplexes. A generalization of polyhedra leads to cell complexes (or CW complexes) glued together from cells.

9.4.1 Simplexes and Polyhedra

We first need to introduce the concept of a simplex. Simply speaking, a simplex is the simplest geometric figure of a respective geometric dimension in the Euclidean space, i.e., a point in a zero-dimensional space, a straight line segment in a one-dimensional space, a triangle in a two-dimensional space, and a tetrahedron in a three-dimensional space.

Definition 73 (Simplex). Given $k+1$ points $v_0, v_1, \dots, v_k \in \mathbb{R}^n$ in general position, where $k \leq n$, we call the smallest convex set containing them a *closed k -simplex* (or *simplex of dimension k*), written as $\bar{\sigma}^k$. The points v_0, \dots, v_k are called the *vertices* of the simplex. A closed simplex can be written as $\sigma^k = \lambda_0 v_0 + \lambda_1 v_1 + \dots + \lambda_k v_k$ where the $\lambda_0, \dots, \lambda_k \in \mathbb{R}_0^+$ and $\lambda_0 + \lambda_1 + \dots + \lambda_k = 1$.

If we require $\lambda_0, \dots, \lambda_k \in \mathbb{R}^+$ (positive real numbers excluding zero) we get an *open k -simplex*, written as σ^k .

Figure 35 illustrates the definition with closed simplexes of dimensions 0, 1, 2, and 3.

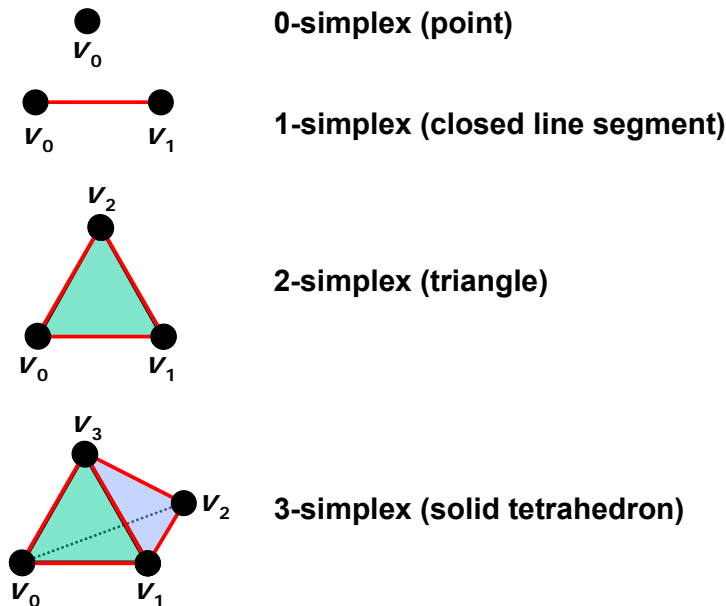


Figure 35. Simplexes of dimension 0, 1, 2, and 3

The convex hull of a nonempty subset of the vertices of a closed simplex $\bar{\sigma}^k$ is called a *face* of the simplex. If a simplex is of dimension n then a k -face is a simplex of dimension $k < n$.

In Figure 35 the closed 1-simplex has two 0-faces v_0 and v_1 . The triangle has three 0-faces, v_0, v_1 and v_2 , and three 1-faces, v_0v_1 , v_1v_2 , and v_2v_0 . The tetrahedron has four 0-faces, six 1-faces, and four 2-faces.

A simplex is a topological space with the natural topology derived from its embedding Euclidean space. We can now piece together simplexes in a defined way to a simplicial complex.

Definition 74 (Simplicial Complex). A finite collection K of closed simplexes in a Euclidean space \mathbb{R}^n is called a *simplicial complex* if the following two conditions are satisfied:

1. For every simplex all of its faces must also be in the collection.
2. If two simplexes intersect then they must do so in a common face.

Figure 36 shows two collections of simplexes in the 2-dimensional Euclidean space. The one on the left is a simplicial complex. The right one violates the conditions for a simplicial complex. The lower triangle touches the other one on the base. There is, however, no common face. The line segment intersects the upper triangle, but not in a common face.

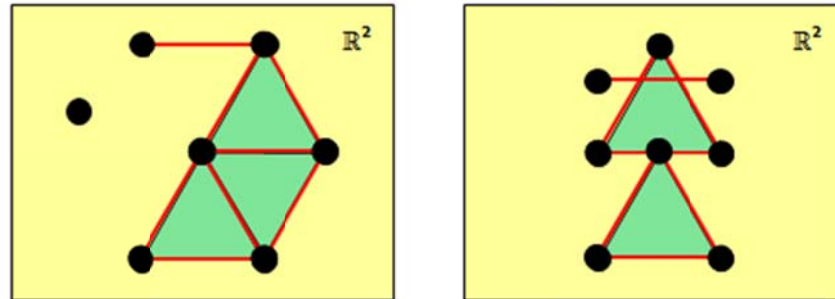


Figure 36. Valid simplicial complex (left) and invalid simplicial complex (right)

Note that a simplicial complex is a set of simplexes and as such not a topological space. However, if we consider the union of all simplexes in a simplicial complex as a subset of a Euclidean space, we can apply the subspace topology¹⁹ and make it a topological space. A simplicial complex K , when viewed in this way, is a topological space that we call a *polyhedron*, written as $|K|$.

Polyhedra possess useful properties. As closed and bounded subsets of a Euclidean space they are compact and a metric space.

9.4.2 Cells and Cell Complexes

For many topological investigations polyhedra are too special and too complex. On the other hand, general topological spaces are too general. A concept in between that also possesses many useful properties is a cell complex.

¹⁹ Given a topological space X and a subset Y of this space, we define a *subspace* by intersecting the open sets of X with Y . This gives us the open sets for a new topology on Y , the *subspace topology*. Y is called a *subspace* of X .

Definition 75 (Unit Ball, Unit Sphere, Unit Cell, Cell). Let \mathbb{R}^n be the n -dimensional Euclidean space with the natural topology.

The subspace $D^n = \{x \in \mathbb{R}^n \mid |x| \leq 1\}$ is called the n -dimensional *unit ball*.

The $(n-1)$ -dimensional subspace $S^{n-1} = \{x \in \mathbb{R}^n \mid |x| = 1\}$ is called the $(n-1)$ -dimensional *unit sphere*.

The subspace $\overset{\circ}{D}^n = \{x \in \mathbb{R}^n \mid |x| < 1\}$ is called the n -dimensional *unit cell*. A topological space homeomorphic to $\overset{\circ}{D}^n$ is called a n -dimensional *cell* (or n -cell).

Example 104. In \mathbb{R}^2 the unit ball is a closed disk with radius 1, the unit sphere is the circle with radius 1, and the unit cell is the open disk with radius 1.

Example 105. Every open n -simplex is a n -cell.

Figure 37 shows unit balls of dimension 0, 1, 2, and 3 and corresponding 0-, 1-, 2-, and 3-cells.

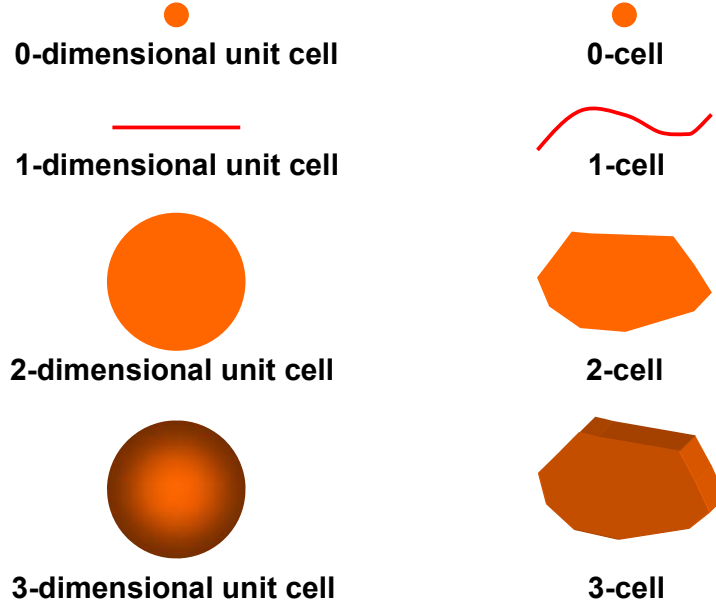


Figure 37. Unit balls and cells

We can now generalize the concept of a simplicial complex to a cell complex, which is defined as a collection of cells that are glued together in a certain way.

Definition 76 (Cell decomposition, skeleton). A *cell decomposition* is a topological space X and a set \mathcal{C} of subspaces of X whose elements are cells such that X is the disjoint union of these cells, i.e., $X = \bigcup_{c \in \mathcal{C}} c$. The n -dimensional *skeleton* of X is the subspace $X^n = \bigcup \{c \in \mathcal{C} \mid \dim(c) \leq n\}$. We have then a sequence of subspaces $\emptyset = X^{-1} \subset X^0 \subset X^1 \subset \dots \subset X^{n-1} \subset X^n$ with $\bigcup X^n = X$.

Figure 38 shows a cell decomposition of a 2-dimensional space with the 1- and 0-dimensional skeletons.

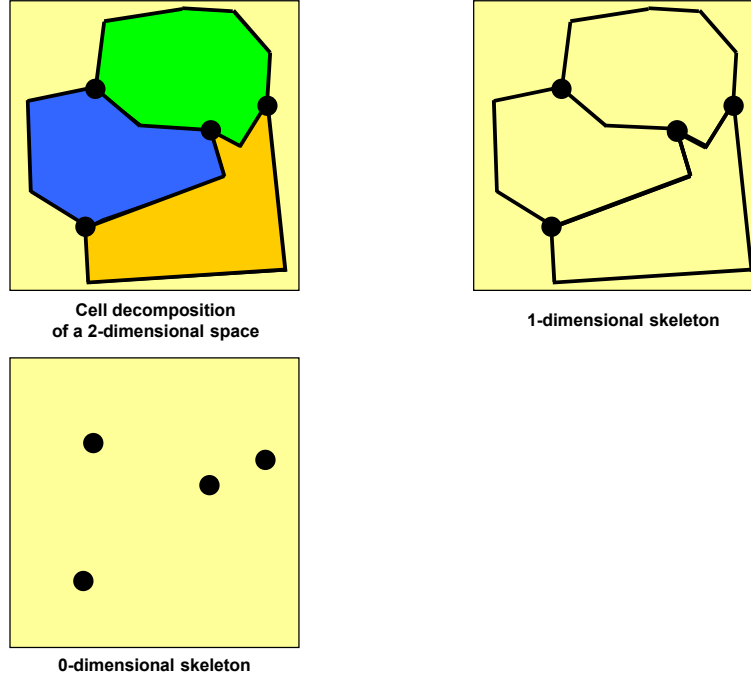


Figure 38. Cell decomposition and skeletons

Example 106. The open simplexes of a polyhedron $|K|$ are a cell decomposition of $|K|$. This means that the disjoint union of 0-, 1-, and 2-cells (of for instance a 2-dimensional polyhedron) is equal to $|K|$.

Definition 77 (Closure and boundary of cells). For every cell we have \bar{c} a closed cell or the *closure* of c in X . The difference $\partial c = \bar{c} - c$ is the *boundary* of c .

It is important to note that the boundary of a cell in general is not the same as the boundary of a set. The boundary of a set is always defined with regard to an embedding space, whereas the boundary of a cell is depending on the dimension of the cell which is clearly determined.

Example 107. Consider a line segment L in \mathbb{R}^3 . The point set topological boundary of L is defined as $\partial L = \bar{L} \cap \overline{\mathbb{R}^3 - L}$, which is all of \bar{L} . If, however, L is a 1-dimensional cell, then its boundary are the two end points.

Definition 78 (Cell complex). A HAUSDORFF space X with a cell decomposition is a *cell complex* (or *CW complex*) if the following conditions are met:

1. For every cell c there exists a continuous function $f: D^n \rightarrow X$ such that $f(S^{n-1}) \subset X^{n-1}$ and the open cell c is a homeomorphic image of the unit cell, i.e., $f(D^n) = c$.
2. Every closed cell \bar{c} is contained in a finite union of open cells.
3. A subspace $A \subset X$, such that for every cell c , $A \cap \bar{c}$ is closed in \bar{c} , is closed in X .

A CW complex is n -dimensional if $X = X^n \neq X^{n-1}$. If a cell complex has a finite number of cells it is called a finite CW complex..

Condition 1 defines a function from the n -dimensional unit ball to the space X such that the open cell appears as a homeomorphic image of the unit cell and the $(n-1)$ -sphere is continuously mapped to a subset of the X^{n-1} -space. In particular, we have $f : (D^n, S^{n-1}) \rightarrow (\bar{c}, \partial c)$ or $f(D^n \cup S^{n-1}) = c \cup \partial c$. The closed cell and the boundary are compact. Condition 2 is also called closure finite; condition 3 is the condition for the so-called weak topology.

The following statements underline the differences between CW complexes and simplicial complexes:

1. Cells of a CW complex need not be geometric simplexes.
2. The closure of a n -cell need not be a n -ball and the boundary of a n -cell need not be a $(n-1)$ -sphere.
3. Not for every $k < n$ with n being the dimension of the CW complex there need to be cells of dimension k . However, every non-empty CW complex has at least one 0-cell.
4. The closure \bar{c} and boundary ∂c of a cell need not be the union of cells.

CW complexes can be easily constructed. Figure 39 illustrates the construction of a 2-dimensional CW complex. We start with a discrete space X^0 (consisting of at least one 0-cell); we then glue 1-cells so that we get X^1 , then we glue 2-cells which gives us X^2 . We see that $X^0 \subset X^1 \subset X^2$.

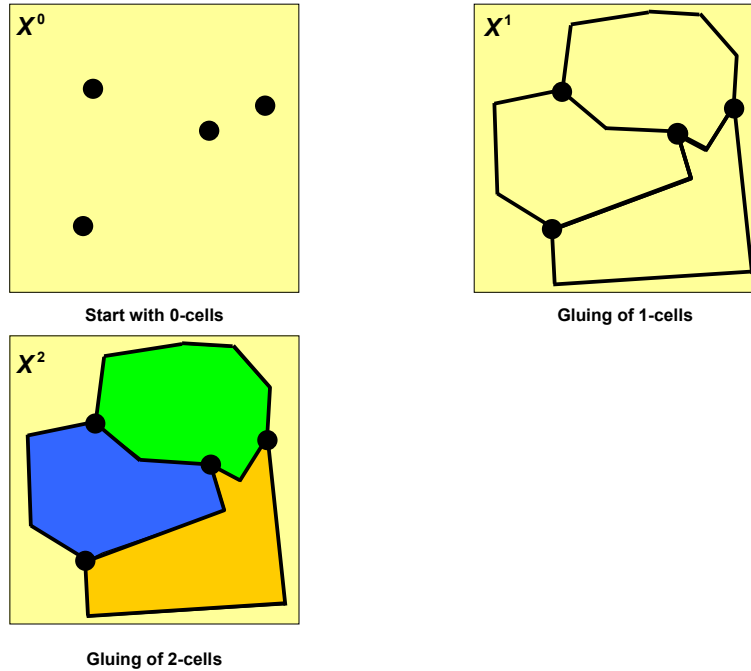


Figure 39. Construction of a CW complex

9.5 Applications in GIS

The space used in GIS to represent spatial features is predominantly the 2-or 3-dimensional Euclidean space \mathbb{R}^2 or \mathbb{R}^3 equipped with the natural topology of open disks and balls. The Euclidean space is a metric space (therefore also a normal, regular, and HAUSDORFF space), second-countable (therefore also first-countable, separable, and

LINDELÖF), and connected. Closed and bounded subsets of a Euclidean space are compact (such as closed cells and simplexes).

Spatial data sets consisting of linear features (network) in \mathbb{R}^2 or \mathbb{R}^3 are 1-dimensional CW complexes where the arcs are the 1-cells and the nodes the 0-dimensional skeleton. Polygon feature data sets are 2-dimensional CW complexes with the polygons as 2-cells and the bounding arcs and nodes as the 1-dimensional skeleton.

9.5.1 Spatial Data Sets

To represent 2-dimensional spatial features in a GIS we have two options: (i) to use a simplicial complex, i.e., to represent all spatial features as a set of simplexes with certain conditions (see Definition 74 of a simplicial complex), or (ii) to represent them as a cell complex by considering the cells being glued together in a proper way (see Definition 78 of a CW complex).

In the first case, we must represent all features by a set of triangles. On the one hand, triangles are very simple structures and easy to handle; on the other hand, every polygon has to be approximated by a potentially large number of triangles which is often undesirable. An exception is a triangular irregular network (TIN) to represent a digital elevation model.

In the second case all features are cells glued together. This approach is much more suitable for general polygon features, because it avoids the use of triangles. In fact, every topologically structured data set in a GIS database is a digital representation of a 2-dimensional cell complex.

Figure 40 shows a 2-dimensional spatial data set as a cell complex embedded in the \mathbb{R}^2 . This complex consists of four 0-cells (1, 2, 3, 4), six 1-cells (a, b, c, d, e, f), and three 2-cells (A, B, C). The embedding Euclidean space \mathbb{R}^2 functions as the “world polygon” or “outside polygon” often denoted as W or O.

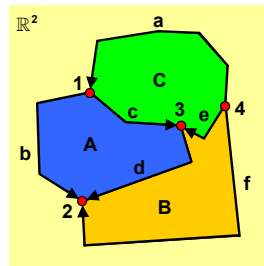


Figure 40. Two-dimensional spatial data set as cell complex

A data structure to represent this cell complex is the so-called arc-node structure. The 0-cells are the nodes and the 1-cells are the arcs between the nodes. Every arc has a start and an end node, thereby defining an orientation of the arc²⁰. The orientation is indicated by arrows in the figure. For every arc we note which polygon (2-cell) lies to the left and which one to the right of it viewed in the direction from start node to end node.

Networks, like road or river networks, are best modeled as a one-dimensional subset (or skeleton) of a cell complex. The topological relationships are then reduced to incidence relationships between edges (arcs) and nodes. Such a structure is also called a graph. Graph theory, although closely related to topology, has developed as an independent mathematical discipline. A special type of graph frequently used in GIS is a planar graph. It is completely embedded in the plane such that no two edges intersect except at nodes.

²⁰ The orientation of an arc is usually determined by the digitization process, i.e., a line is followed from the beginning (start node) to the end (end node).

An implementation of the arc-node structure in a relational database needs one table for the arc-node relations, one for the polygon attributes, and one for the vertices of the arcs. Table 12 shows the arc table of the arc-node structure of the cell complex in Figure 40.

Table 12. Arc table for the arc-node structure

Arc-id	Start-node	End-node	Left-polygon	Right-polygon
a	4	1	C	W
b	1	2	A	W
c	1	3	C	A
d	3	2	B	A
e	4	3	B	C
f	4	2	W	B

9.5.2 Topological Transformations

As we have seen, topology is the branch of mathematics that deals with properties of spaces that remain invariant (i.e., do not change) under topological mappings. Assume you have spatial features stored in a database using the arc-node structure. When you apply a transformation (such as a map projection) to the data set, the neighborhood relationships between A, B, and C remain, and the boundary lines have the same start and end nodes. The areas are still bounded by the same boundary lines, only their shapes and the length of the perimeters have changed (Figure 41).

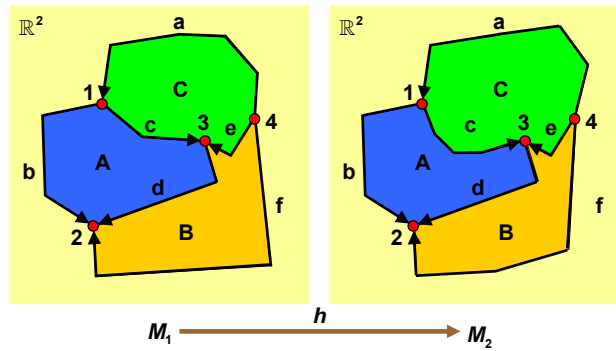


Figure 41. Topological mapping

Topologically speaking, we have applied a homeomorphism $h: M_1 \rightarrow M_2$ from the cell complex M_1 to the cell complex M_2 . They are topologically equivalent.

9.5.3 Topological Consistency

A representation of a cell complex must be consistent, i.e., the topological properties must not be violated. If we can show that the following rules are satisfied for every element in the data set, we know that it is a topologically consistent 2-dimensional configuration.

- (TC1) Every 1-cell is bounded by two 0-cells. (Every arc has a start node and an end node).
- (TC2) For every 1-cell there are two 2-cells. (For every arc there exist two adjacent polygons, the left and right polygon).
- (TC3) Every 2-cell is bounded by a closed cycle of 0- and 1-cells. (Every polygon has a closed boundary consisting of an alternating sequence of nodes and arcs.)
- (TC4) Every 0-cell is surrounded by a closed cycle of 1- and 2-cells. (Around every node there exists an alternating closed sequence of arcs and polygons.)

(TC5) Cells intersect only in 0-cells. (If arcs intersect, they do so in nodes.)

These rules cannot be applied without additions or modifications to other dimensions. In the following we will discuss how these conditions can be checked when we have an arc table.

TC1 demands that every arc must have a start node and end node. The presence of a NOT NULL value for the Start-node and End-node for every arc is sufficient.

TC2 ensures the neighborhood relationship of polygons. The presence of a NOT NULL Left-polygon and Right-polygon for every arc is sufficient.

TC3 ensures that polygons are closed, i.e., starting at any node of the boundary of a polygon, we have a closed cycle of nodes and arcs. We will illustrate a procedure for polygon A in Figure 40:

Select all rows from the arc table where A appears either as Right-polygon or Left-polygon.

Arc-id	Start-node	End-node	Left-polygon	Right-polygon
a	4	1	C	W
b	1	2	A	W
c	1	3	C	A
d	3	2	B	A
e	4	3	B	C
f	4	2	W	B

Make sure that for all selected records A appears always as the Left-polygon or always as the Right-polygon.²¹ For those rows where A is not the Right-polygon, we must swap Left- and Right-polygon. Of course, if we do that, we must also swap Start- and End-node to maintain orientation. In our case we must swap for arc b, which results in the following configuration:

Arc-id	Start-node	End-node	Left-polygon	Right-polygon
a	4	1	C	W
b	2	1	W	A
c	1	3	C	A
d	3	2	B	A
e	4	3	B	C
f	4	2	W	B

We now start at any Start-node of the selected rows and chain through the nodes. In our example let us start with arc c and node 1 (Figure 42). The end-node of c is 3. In the next step look up the record where 3 appears as the start node and continue as before. When we return to the node where we started, the cycle is closed and the polygon boundary is closed. Otherwise, we have an inconsistency in the polygon boundary.

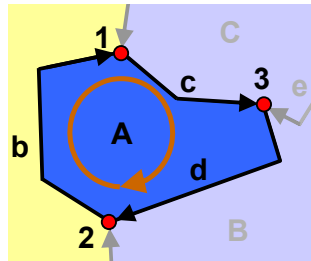


Figure 42. Closed polygon boundary check

²¹ The choice could be based on the fact that for two out of three rows this condition is already fulfilled.

TC4 ensures the planarity of the cell complex near a node, i.e., for every node there must be an “umbrella” of a closed cycle of alternating 1-cells and 2-cells. We will illustrate a procedure for node 3 in Figure 40:

Select all rows from the arc table where 3 appears either as Start-node or End-node.

Arc-id	Start-node	End-node	Left-polygon	Right-polygon
a	4	1	C	W
b	1	2	A	W
c	1	3	C	A
d	3	2	B	A
e	4	3	B	C
f	4	2	W	B

Make sure that for all selected records 3 appears always as the Start-node or always as the End-node. In our example we want 3 always to be the End-node. For those rows where 3 is not the End-node, we must swap Start- and End-node. Of course, if we do that, we must also swap Left- and Right-polygon to maintain orientation. In our case we must swap for arc d, which gives us

Arc-id	Start-node	End-node	Left-polygon	Right-polygon
a	4	1	C	W
b	2	1	W	A
c	1	3	C	A
d	2	3	A	B
e	4	3	B	C
f	4	2	W	B

We now start at any Left-polygon of the selected rows and chain through the polygons. In our example let us start with arc c and Left-polygon C (Figure 43). The Right-polygon of c is A. In the next step look up the record where A appears as the Left-polygon and continue as before. When we return to the polygon where we started, the cycle is closed and the “umbrella” is closed. Otherwise, we have an inconsistency in the node.

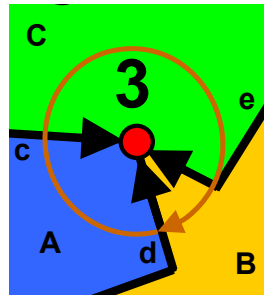


Figure 43. Node consistency check

TC5 must be checked by calculating intersections of arcs and pointing out intersections at locations without nodes.

9.5.4 Spatial Relations

Whereas relationships between simplexes or cells define consistency constraints for spatial data, we can use the topological properties of interior, boundary, and exterior to define relationships between spatial features. Since the properties of interior, boundary, and exterior do not change under topological mappings, we can investigate their possible relations between spatial features.

Let us assume two spatial regions A and B . Both have their respective boundary, interior, and exterior. When we consider all possible combinations of intersections between the boundaries, the interiors, and the exteriors of A and B , we know that these

will not change under any topological transformation. This can be put into a rectangular schema $I_9(A, B)$ which is called the 9-intersection, written as

$$I_9(A, B) = \begin{pmatrix} A^\circ \cap B^\circ & A^\circ \cap \partial B & A^\circ \cap B^- \\ \partial A \cap B^\circ & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^\circ & A^- \cap \partial B & A^- \cap B^- \end{pmatrix}.$$

From these intersection patterns, we can derive eight mutual spatial relationships between two regions. If, for instance, the boundary of A intersects the boundary of B , the interiors of A and B do not intersect, and the exteriors of A and B intersect, we say that A and B meet. Figure 44 shows all possible eight spatial relationships: disjoint, meet, equal, inside, covered by, contains, covers, and overlap. These relationships can be used, for instance, in queries against a spatial database.

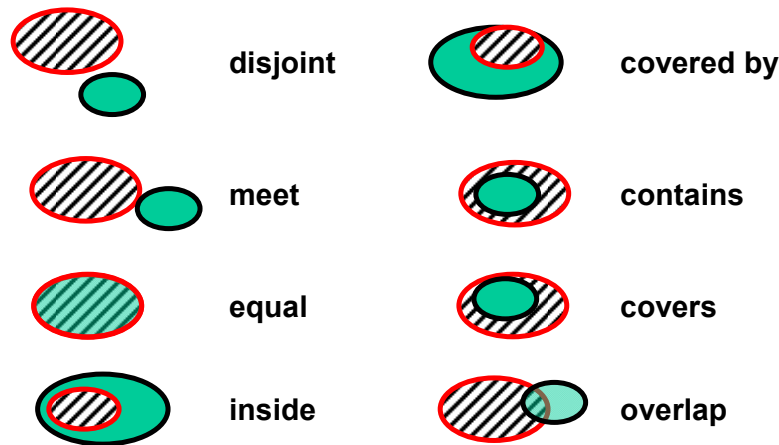


Figure 44. Spatial relationships between two simple regions based on the 9-intersection

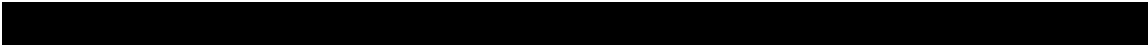
9.6 Exercises

Exercise 34

Ordered Sets

One of the basic structures mathematical disciplines are built upon is order. A set is said to be (partially) ordered when an order relation is defined between its elements, which makes them comparable. The study of partially ordered sets and lattices (a special kind of ordered set) is covered by an extensive amount of mathematical literature. This theory has mainly been applied in computer science, such as in multiple inheritance or Boolean algebra.

In this chapter, we will introduce the basic principles of partially ordered sets and lattices and show how they can be applied to spatial features and their relationships with each other.



10.1 Posets

Definition 79 (Partially Ordered Set). Let P be a set. A binary relation \leq on P such that, for every $x, y, z \in P$

1. $x \leq x$ (reflexive)
2. if $x \leq y$ and $y \leq x$, then $x = y$ (antisymmetric)
3. if $x \leq y$ and $y \leq z$, then $x \leq z$ (transitive)

is called a *partial order* on P . A set P equipped with a reflexive, antisymmetric and transitive relation (order relation) \leq is called a *partially ordered set* (or *poset*) and is written as $(P; \leq)$. Usually we will write P with the meaning ‘ P is a poset’.

For every partially ordered set P we can find a new poset, the dual of P , by defining that $x \leq y$ in the dual if and only if $y \leq x$ in P . Any statement about a partially ordered set can be turned into a statement of its dual by replacing \leq with \geq and vice versa.

Example 108. The natural numbers with the relation \leq read as “less than or equal” are a poset.

When we take the power set $\wp(X)$ of a set X , i.e. all subsets of X , then $\wp(X)$ is ordered by set inclusion and for every $A, B \in \wp(X)$ we define $A \leq B$ if and only if $A \subseteq B$.

Example 109. For spatial subdivisions A and B the order relation $A \leq B$ means that “ A is contained in B ” or dually, that “ B contains A ”.

Any hierarchy is a poset with at most one element directly above any element. A special type of hierarchy—and therefore a more special type of poset—is the *totally ordered set* (or *chain*). This is a hierarchy in which at most one element is directly below any specific element, which means that every element can be compared with every other element in the set. The integer space is a typical example of a total ordering.

10.1.1 Order Diagrams

For every (finite) poset there exists a graphical representation, the *diagram* (or *Hasse diagram*) of the poset. To describe how to construct a diagram we need the idea of covering:

Definition 80 (Cover). By “ A covers B ” (or “ B is covered by A ”) in a poset P we mean that $B \leq A$ and there exists no $x \in P$ that $B < x < A$ and we write $A >-B$ or $B-<A$. In other words, A covers B means that A is immediately greater than B and there is no other element in between. The set of all elements that cover an element X is called the *cover* of X , written as X^+ . Dually, the set of all elements that are covered by X is called the *cocover* of X and we write X_- .

A diagram of a poset P is drawn as a configuration of circles (representing the elements of P) and connecting lines (indicating the covering relation), where the circle for element A is drawn above the circle for element B , when A covers B . The circles are connected with a straight line. For a finite poset we obtain the diagram of the dual by turning it upside down. Figure 45 shows a poset and its corresponding diagram.

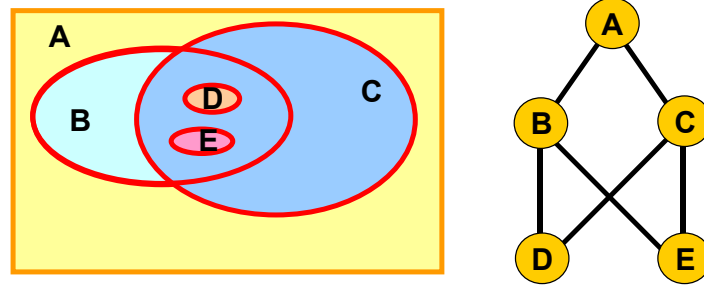


Figure 45. Poset and corresponding diagram

The circles and the connecting lines in a diagram can be viewed as vertices and edges of a graph. Since the order relation defines a direction for the edges, a Hasse diagram is a directed graph. There are no cycles in that graph, i.e. starting from a specific node and moving along the edges of the graph in the given direction, no node is visited twice. Such a graph is called a *directed acyclic graph* (or *dag*). There are many algorithms for traversing directed acyclic graphs and for other related operations.

Definition 81 (Maximum and Minimum). Let P be a poset and $S \subseteq P$. An element $a \in S$ is the *greatest* (or *maximum*) *element* of S if $a \geq x$ for every $x \in S$ and we write $a = \max S$. The greatest element of P , if it exists, is called the *top element* of P and the *least* (or *minimum*) *element* of S , written as $\min S$, and the *bottom element* of P , if it exists, are defined by duality.

Example 110. In $\wp(X)$ we have X as the top element and the empty set as the bottom element.

The natural numbers under their usual order have 1 as the bottom element but no top element.

10.1.2 Upper and Lower Bounds

Definition 82 (Upper Bound). Let P be a poset and $S \subseteq P$. An element $x \in P$ is an *upper bound* of S if $s \leq x$ for all $s \in S$. A *lower bound* is defined by duality. The set of all upper bounds of S is denoted by S^* (or “ S upper”) and the set of all lower bounds (or “ S lower”) is written as S_* ; in other words we define $S^* = \{x \in P \mid (\forall s \in S) s \leq x\}$ and $S_* = \{x \in P \mid (\forall s \in S) s \geq x\}$.

If S^* has a least element, it is called *least upper bound* (l.u.b.), also *join* or *supremum*. By duality, if S_* has a largest element, it is called *greatest lower bound* (g.l.b.), *meet* or *infimum*. If a least upper bound or a greatest lower bound exists, it is always unique. For the least upper bound and the greatest lower bound of two elements x and y we write $\sup\{x, y\}$ or $x \vee y$ (read as “ x join y ”) and $\inf\{x, y\}$ or $x \wedge y$ (read as “ x meet y ”), respectively. For a subset S we write $\vee S$ (the “*join of S* ”) or $\sup S$ and $\wedge S$ (the “*meet of S* ”) or $\inf S$.

There are cases when a greatest lower bound or a least upper bound does not exist. This may be the case because elements do not have common bounds or because a g.l.b. or l.u.b. does not exist. Take for example the two elements B and C of Figure 45. The set of their lower bounds are D and E . However, none of the lower bounds is greater than the other, they are not comparable. Therefore, there is no greatest lower bound for the subset $\{B, C\}$ (see Figure 46).

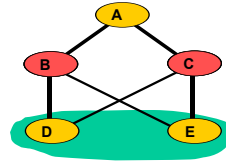


Figure 46. Lower bounds

10.2 Lattices

In the previous section we have seen that in the general case of a partially ordered set we cannot expect that join and meet always exist. Therefore, a more specific order structure is needed.

Definition 83 (Lattice). A lattice L is a poset in which every pair of elements has a least upper bound and a greatest lower bound. A lattice is called *complete*, when meet and join exist for every subset of the poset²².

If L is a lattice then \wedge and \vee are binary operations on L and we have an algebraic structure $\langle L, \wedge, \vee \rangle$ with \wedge and \vee satisfying the following conditions for all $a, b, c \in L$:

1. $(a \vee b) \vee c = a \vee (b \vee c)$, $(a \wedge b) \wedge c = a \wedge (b \wedge c)$ (associative laws)
2. $a \vee b = b \vee a$, $a \wedge b = b \wedge a$ (commutative laws)
3. $a \vee a = a$, $a \wedge a = a$ (idempotency laws)
4. $a \vee (a \wedge b) = a$, $a \wedge (a \vee b) = a$ (absorption laws)

Every set L with two binary operations satisfying conditions (1) to (4) is a lattice. We see that a lattice can be viewed as either being an order structure or an algebraic structure. Many theories, e.g. Boolean algebras, rely heavily on the algebraic properties of lattices.

The order relation \leq is related to the algebraic operations of \wedge and \vee by the following statement:

Let L be a lattice and let x and y be elements of L . Then $x \leq y$ is equivalent to each of the conditions: $x \wedge y = x$ and $x \vee y = y$.

It can be proven that every finite lattice is complete. This is an important result because it means that whenever we have a lattice with a finite number of elements we can always find least upper bounds and greatest lower bounds for every subset of the lattice.

Example 111. Every chain is a lattice in which $x \vee y = \max\{x, y\}$ and $x \wedge y = \min\{x, y\}$. Therefore, the natural numbers, integers, rational and real numbers all are lattices under their usual order. None of them is a complete lattice. To show this let us take any of these sets and determine

²² Note that the difference between a lattice and a complete lattice is in the existence of the meet and join for every *pair* of elements (lattice) or every *subset* of elements (complete lattice).

the supremum of the set itself. Since there is no greatest number in any of these sets, the set of the upper bounds is empty and a least upper bound does not exist.

Example 112. The power set $\wp(X)$ of any set X is a complete lattice where meet and join are defined as $\bigwedge \{A_i \mid i \in I\} = \bigcap_{i \in I} A_i$ and $\bigvee \{A_i \mid i \in I\} = \bigcup_{i \in I} A_i$, respectively.

If a subset $S \subseteq \wp(X)$ is closed under finite unions and intersections, it is called a *lattice of sets*. It is called a *complete lattice of sets* if it is closed under arbitrary unions and intersections. Meet and join are then defined as set intersection and set union.

If L is a complete lattice then the following is true for every $S, T \subseteq L$:

- (i) $\forall s \in S, s \leq \bigvee S$ and $s \geq \bigwedge S$.
- (ii) Let $x \in L$. Then $x \leq \bigwedge S$ if and only if $x \leq s$ for all $s \in S$.
- (iii) Let $x \in L$. Then $x \geq \bigvee S$ if and only if $x \geq s$ for all $s \in S$.
- (iv) $\bigvee S \leq \bigwedge T$ if and only if $s \leq t$ for all $s \in S$ and all $t \in T$.
- (v) If $S \subseteq T$, then $\bigvee S \leq \bigvee T$ and $\bigwedge S \geq \bigwedge T$.
- (vi) $\bigvee (S \cup T) = (\bigvee S) \vee (\bigvee T)$ and $\bigwedge (S \cup T) = (\bigwedge S) \wedge (\bigwedge T)$.

10.3 Normal Completion

Not every poset is a lattice, because posets exist in which not all subsets have greatest lower bounds and least upper bounds. For example, the subset $\{B, C\}$ of the order in Figure 46 has no greatest lower bound. It is, however, possible, to add elements to a poset to create a lattice. This is in fact possible with all posets.

It is even more interesting to find the smallest number of elements necessary to add to a poset to create a lattice. In other words, we want to build the minimal containing lattice of a poset. The method for doing this is called *normal completion*.

In order to define the normal completion we need the concept of a closure operator, which is defined as follows:

Definition 84 (Closure). Let X be a set. A map $C : \wp(X) \rightarrow \wp(X)$ is called a *closure operator* on X if, for all $A, B \subseteq X$:

1. $A \subseteq C(A)$
2. If $A \subseteq B$, then $C(A) \subseteq C(B)$
3. $C(C(A)) = C(A)$

A subset A of X is called *closed* if $C(A) = A$.

The following theorem summarizes the important facts about the normal completion of posets. It even gives us a procedure for building the normal completion lattice.

Let P be a poset and $(A^*)_*$ the set of the lower bounds of the upper bounds of a subset A of P . Then

1. $C(A) = (A^*)_*$ defines a closure operator on P .

2. The family $DM(P) = \{A \subseteq P \mid (A^*)_* = A\}$ is a complete lattice (the *DEDEKIND-MACNEILLE completion*, or *normal completion* or *completion by cuts* of P), when ordered by inclusion, in which $\bigwedge \{A_i \mid i \in I\} = \bigcap_{i \in I} A_i$ and $\bigvee \{A_i \mid i \in I\} = C(\bigcup_{i \in I} A_i)$.
3. The map $\varphi: P \rightarrow DM(P)$ defined by $\varphi(x) = (x^*)_*$ for all $x \in P$ is an *order-embedding*, i.e. it is order-preserving and injective. In fact φ can be defined as $\varphi(x) = x_* = \{y \in P \mid y \leq x\}$, because $(x^*)_* = x_*$ for all $x \in P$. $DM(P)$ is a completion of P via φ and all greatest lower bounds and least upper bounds which exist in P are preserved. This means, if $A \subseteq P$ and $\bigvee A$ exists in P , then $\varphi(\bigvee A) = \bigvee \varphi(A)$, and $\varphi(\bigwedge A) = \bigwedge \varphi(A)$.
4. $DM(P)$ is the smallest lattice in which P can be embedded in the sense, that if L is any other lattice such that $P \subseteq L$, we have $P \subseteq DM(P) \subseteq L$.

By calculating the normal completion we have to look at all subsets of the poset P . For practical applications, this is rather inefficient, because every set with n elements has 2^n subsets.

The theorem above has a simple corollary, which yields two important properties of the normal completion lattice:

1. If L is a lattice, then $L = DM(L)$.
2. For all posets P we have $DM(P) = DM(DM(P))$.

First, the corollary tells us that whenever the poset is already a lattice, the normal completion does not add anything to the lattice. It leaves the lattice unchanged. Secondly, it follows from the idempotency of a closure operator that applying the completion more than once does not increase the number of elements added to the completion lattice, i.e. the number of elements in the completion lattice is bounded by 2^n for n elements in the poset.

10.3.1 Special Elements

Let P be a poset and S a subset of the poset. We had defined upper bounds and lower bounds for the subset S . The set of all upper bounds of S was denoted as S^* and the set of all lower bounds of S as S_* . For the normal completion lattice we need to identify all $(S^*)_*$ for all subsets of P .

If there exists a greatest element in P it is called *top element* and written as \top ; if there is a least element in P it is called *bottom element* and written as \perp .

There are two cases that require special attention: when $S = P$ and when $S = \emptyset$. First, let us investigate the case when $S = P$. If P has a top element, then $P^* = \{\top\}$ and $\sup P = \top$. When P has no top element, then $P^* = \emptyset$ and there is no supremum of P . By duality, if P has a bottom element, then $P_* = \{\perp\}$ and $\inf P = \perp$. If P does not have a bottom element, then $P_* = \emptyset$ and the infimum does not exist.

Now, let us assume that $S = \emptyset$, i.e., S is the empty subset of P . Then (vacuously) for all $s \in S$ we have that $s \leq x$ for every element $x \in P$. Thus $\emptyset^* = P$ and $\sup \emptyset$ exists, if and only if P has a bottom element; i.e., then we have $\sup \emptyset = \perp$. Dually, $\emptyset_* = P$

(because again we have vacuously that for all $s \in S = \emptyset$, $s \geq x$ for every $x \in P$) and $\inf P = \top$ whenever P has a top element. Table 13 summarizes the result.

Table 13. Special elements and the closure operator in the normal completion

Subset	S^*		S_*	
	Top element exists	No top element	Bottom element exists	No bottom element
P	$\{\top\}$	\emptyset	$\{\perp\}$	\emptyset
\emptyset	P	P	P	P

From the table above we can derive the following sets:

$$(P^*)_* = P$$

$$(\emptyset^*)_* = \begin{cases} \{\perp\} & \text{if } P \text{ has a bottom element} \\ \emptyset & \text{otherwise} \end{cases}$$

10.3.2 Normal Completion Algorithm

The algorithm for the normal completion can be written as follows:

1. Determine all subsets, i.e., the power set $\wp(P)$, of the poset P .
2. For every subset $S \in \wp(P)$ determine $(S^*)_*$.
3. Arrange all $(S^*)_*$ to a poset where \subseteq (subset) is the order relation.
4. Identify every element $a \in P$ of the original poset with its corresponding $(a^*)_*$ in the new poset.
5. Assign suitable symbols to the remaining elements of the new poset.
6. The resulting poset is the normal completion lattice of P .

To illustrate how this works we take the poset of Figure 45 and build the normal completion lattice according to the algorithm. First we determine all subsets of the poset $\{A, B, C, D, E\}$. This results in 32 sets. For every subset S we must then compute $(S^*)_*$. The result is given in Table 14.

Table 14. Normal completion

	S	S^*	$(S^*)_*$
1	\emptyset	$\{A, B, C, D, E\}$	\emptyset
2	$\{A\}$	$\{A\}$	$\{A, B, C, D, E\}$
3	$\{B\}$	$\{A, B\}$	$\{B, D, E\}$
4	$\{C\}$	$\{A, C\}$	$\{C, D, E\}$
5	$\{D\}$	$\{A, B, C, D\}$	$\{D\}$
6	$\{E\}$	$\{A, B, C, E\}$	$\{E\}$
7	$\{A, B\}$	$\{A\}$	$\{A, B, C, D, E\}$
8	$\{A, C\}$	$\{A\}$	$\{A, B, C, D, E\}$
9	$\{A, D\}$	$\{A\}$	$\{A, B, C, D, E\}$
10	$\{A, E\}$	$\{A\}$	$\{A, B, C, D, E\}$
11	$\{B, C\}$	$\{A\}$	$\{A, B, C, D, E\}$
12	$\{B, D\}$	$\{A, B\}$	$\{B, D, E\}$

13	$\{B, E\}$	$\{A, B\}$	$\{B, D, E\}$
14	$\{C, D\}$	$\{A, C\}$	$\{C, D, E\}$
15	$\{C, E\}$	$\{A, C\}$	$\{C, D, E\}$
16	$\{D, E\}$	$\{A, B, C\}$	$\{D, E\}$
17	$\{A, B, C\}$	$\{A\}$	$\{A, B, C, D, E\}$
18	$\{A, B, D\}$	$\{A\}$	$\{A, B, C, D, E\}$
19	$\{A, B, E\}$	$\{A\}$	$\{A, B, C, D, E\}$
20	$\{A, C, D\}$	$\{A\}$	$\{A, B, C, D, E\}$
21	$\{A, C, E\}$	$\{A\}$	$\{A, B, C, D, E\}$
22	$\{A, D, E\}$	$\{A\}$	$\{A, B, C, D, E\}$
23	$\{B, C, D\}$	$\{A\}$	$\{A, B, C, D, E\}$
24	$\{B, C, E\}$	$\{A\}$	$\{A, B, C, D, E\}$
25	$\{B, D, E\}$	$\{A, B\}$	$\{B, D, E\}$
26	$\{C, D, E\}$	$\{A, C\}$	$\{C, D, E\}$
27	$\{A, B, C, D\}$	$\{A\}$	$\{A, B, C, D, E\}$
28	$\{A, B, C, E\}$	$\{A\}$	$\{A, B, C, D, E\}$
29	$\{A, B, D, E\}$	$\{A\}$	$\{A, B, C, D, E\}$
30	$\{A, C, D, E\}$	$\{A\}$	$\{A, B, C, D, E\}$
31	$\{B, C, D, E\}$	$\{A\}$	$\{A, B, C, D, E\}$
32	$\{A, B, C, D, E\}$	$\{A\}$	$\{A, B, C, D, E\}$

The resulting sets are $\{A, B, C, D, E\}$, $\{B, D, E\}$, $\{C, D, E\}$, $\{D, E\}$, $\{D\}$, $\{E\}$, and \emptyset . When we arrange them in a poset according to the subset relation, we get the normal completion lattice (Figure 48)²³.

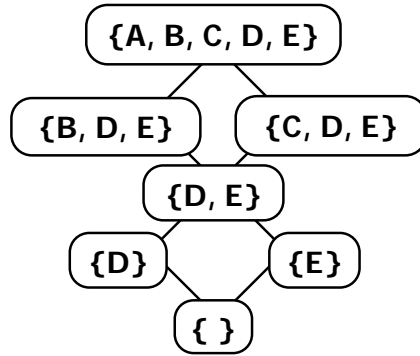


Figure 47. Normal completion lattice

Finally, we identify the original poset elements with their corresponding lattice elements and denote the newly created elements with X and $\{\}$. Figure 48 shows the normal completion of the poset. We see that two new elements were added to the poset to form a lattice.

²³ Note that we may use either $\{\}$ or \emptyset to denote the empty set.

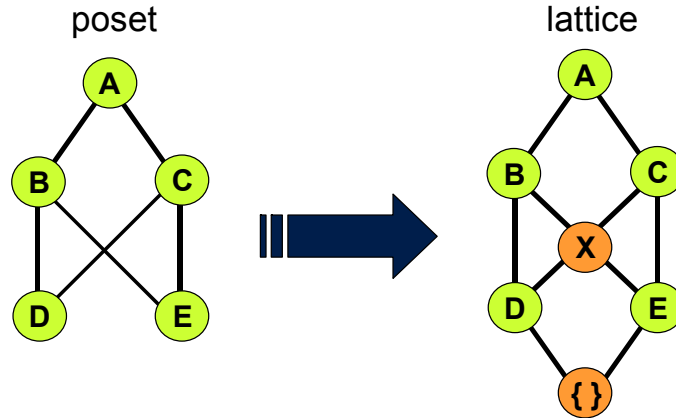


Figure 48. Normal completion

The new elements can be interpreted in a geometric way as shown in Figure 49. Element X can be interpreted as the intersection of B and C .

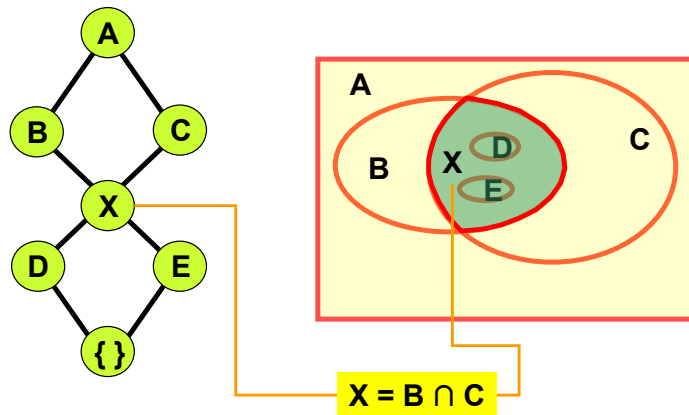


Figure 49. Geometric interpretation of new lattice elements

10.4 Application in GIS

The intuitive interpretation of order relations as “is contained in” or, dually, as “contains” can be used for relationships among spatial features such as polygons, lines and points. The structure of a poset accommodates both strict hierarchies (every object has exactly one parent object) and relationships where one object possesses more than one parent object.

Examples of hierarchies are administrative subdivisions where for instance every county belongs to exactly one state, and every state belongs to exactly one country. General posets can be used to represent situations where one object belongs to several parents, such as agricultural production zones that may be part of several municipalities, or regions that are composed of several unconnected polygons such as the Hawaiian Islands.

10.5 Exercises


Exercise 35 From the poset in Figure 45 determine the upper bounds of (a) $\{D\}$, (c) $\{D, C\}$, (c) $\{A\}$.

- Exercise 36* From the poset in Figure 45 determine the greatest lower bound of (a) $\{B, D\}$, (b) $\{A\}$, (c) $\{A, B, C\}$
- Exercise 37* The following relationships are given for the four regions A, B, C, and D: C is contained in A and D is contained in B. Draw the poset for the four regions, compute and draw the normal completion lattice.

Graph Theory

The origin of graph theory lies in the investigation of topological problems given by a set of points and the connections between them. Today, graph theory is a branch of mathematics in its own right dealing with problems that can be represented by a collection of vertices and connecting edges.

This chapter deals with the basic principles of graphs, their representation and ways to traverse them. The importance of graph theory for the analysis of transpiration and flow problems is highlighted in the section on applications to GIS.



11.1 Introducing Graphs

Generally, the origin of graph theory is attributed to the Swiss mathematician LEONHARD EULER who published a paper in 1736 on what is now commonly known as the Königsberg bridge problem. Figure 50 shows a sketch of the seven bridges across the river Pregel in Königsberg (which is today's Kaliningrad). The problem is to determine whether it is possible to make a circular walk through Königsberg by starting at a river bank and crossing every bridge exactly once.

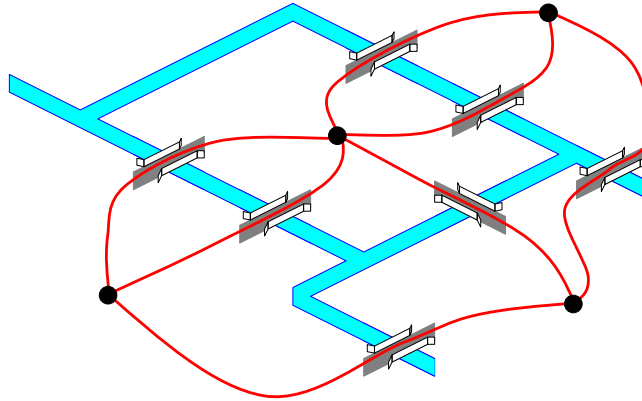


Figure 50. The seven bridges of Königsberg

Euler solved the problem by abstracting the island and river banks to points and representing the bridges by lines connecting these points. In the figure they are represented by black points and red lines.

Figure 51 shows these points (*vertices*) and lines (*edges*) in a schematic way with the vertices numbered v_1 to v_4 , and the edges e_1 to e_7 . Such a configuration is called a *graph*. Starting from an arbitrary vertex we find after some tries that such a circular walk is impossible²⁴.

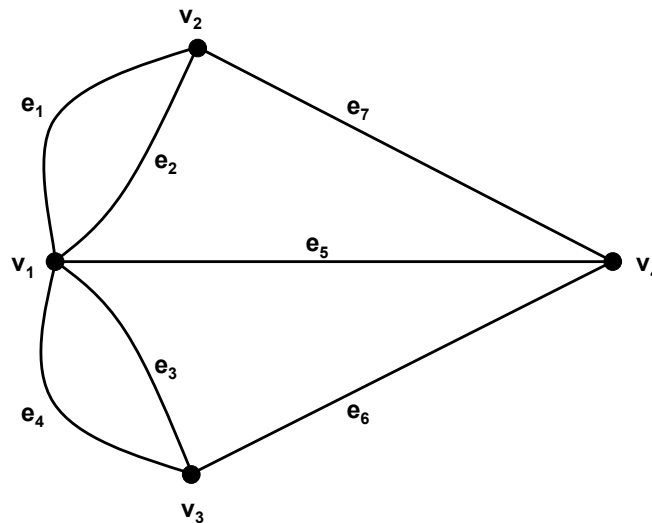


Figure 51. Graph of the Königsberg bridge problem

²⁴ We will see later that the problem is to find an Eulerian circuit in the graph and that there is a theorem stating when such a circuit exists.

11.1.1 Basic Concepts

Definition 85 (Graph). Given a non-empty set $V = \{v_1, v_2, \dots, v_n\}$, the *vertex-set*, a set $E = \{e_1, e_2, \dots, e_m\}$, the *edge-set*, and a function $g : E \rightarrow V \times V$, the *incidence map*, which assigns to every element of E a pair (v_i, v_j) of elements of V . We call the triple $G = (V, E, g)$ a *graph*.

The elements of V are called *points* (or *vertices*), the elements of E are called *edges*. For an edge $e = (v_i, v_j)$ vertices v_i and v_j are called *end points* of e ; we say that e is *incident with* v_i and v_j , and that v_i is *adjacent to* v_j . If $g(e) = (v, v)$ we call e a *loop*. If $g(e_i) = g(e_j)$ we call e_i and e_j *parallel edges*.

Here, we will deal only with finite graphs, i.e., the number of vertices and the number of edges are both finite. When there is no confusion possible we will denote a graph with $G = (V, E)$ for short.

Example 113. The graph for the Königsberg bridge problem in Figure 51 can be written as $G = (V, E, g)$ with the vertex-set $V = \{v_1, v_2, v_3, v_4\}$, the edge-set $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$ and the incidence map defined as $g(e_1) = (v_1, v_2)$, $g(e_2) = (v_1, v_2)$, $g(e_3) = (v_1, v_3)$, $g(e_4) = (v_1, v_3)$, $g(e_5) = (v_1, v_4)$, $g(e_6) = (v_3, v_4)$, $g(e_7) = (v_2, v_4)$. Edges e_1, e_2 and e_3, e_4 are parallel. The graph contains no loop.

A graph without loops and parallel edges is called a *simple graph*. A graph with loops and parallel edges is sometimes called a *multigraph*. The number of edges incident with a vertex v is called the *degree* of v and is written as $d(v)$. A vertex v with $d(v) = 0$ is called an *isolated vertex*.

Example 114. In the previous example the degree of vertex v_1 is 5, and the degrees of vertices v_2, v_3 and v_4 are three.

A graph is called *complete* if there exists an edge for every pair of distinct vertices. For n vertices the complete graph is denoted by K_n . We call a graph *regular* when every vertex has the same degree. If the degree is k then the graph is *k-regular*. A complete graph K_n is $(n-1)$ -regular, i.e., every vertex in a complete graph K_n has degree $(n-1)$. Figure 52 illustrates some complete graphs.

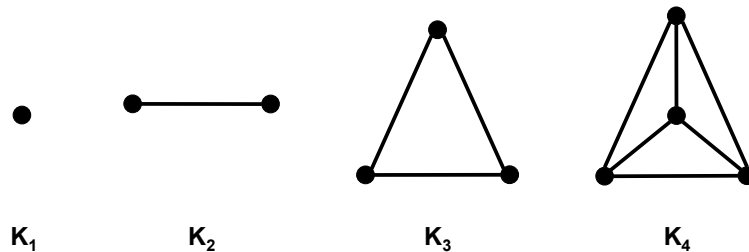


Figure 52. Complete graphs

Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are *isomorphic* if there is a bijective mapping $i : V_1 \rightarrow V_2$ preserving the incidence relationships, i.e., for every $v_1, v_2 \in V_1$ we have $(v_1, v_2) \in E_1$ implies $(i(v_1), i(v_2)) \in E_2$. Isomorphic graphs have the same

structure, although they might look quite different at a first glance. Figure 53 shows two isomorphic graphs.

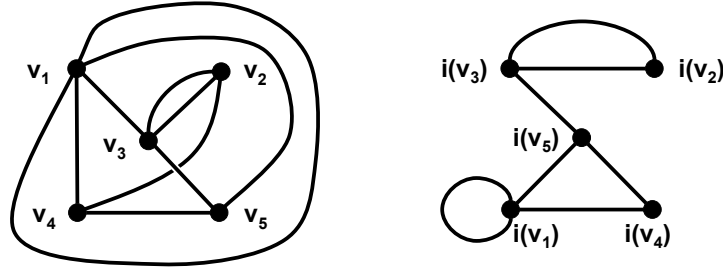


Figure 53. Isomorphic graphs

If we remove a number of edges or vertices from a graph G we obtain a *subgraph* $S \subseteq G$. The removal of a vertex implies that all edges incident with it must also be removed. However, if we remove an edge the vertices remain. The result could be some isolated vertices. A subset of vertices $V' \subset V$ and edges with both end-points in V' is called a *subgraph induced* by V' .

11.1.2 Path, Circuit, Connectivity

For many applications we need to traverse a graph, sometimes in a particular way. A *path* from v_1 to v_n is a sequence of alternating vertices and edges $P = v_1, e_1, v_2, e_2, \dots, e_{n-1}, v_n$ such that for $1 \leq i < n$, e_i is incident with v_i and v_{i+1} . For a simple graph it is sufficient to list only the vertices in a path. If $v_1 = v_n$ then the path is called a *cycle* or *circuit*. A path is called a *simple* path if every vertex is visited only once. In a *simple* circuit every vertex appears once except that $v_1 = v_n$. The *length* of a path or a circuit is the number of edges it contains.

Example 115. In the graph of Figure 51 $P = v_1, e_2, v_2, e_1, v_4$ is a simple path from v_1 to v_4 . $C = v_4, e_5, v_1, e_4, v_3, v_1, e_3, v_4$ is circuit. Note that it is not a simple circuit, because vertex v_1 is visited twice.

When we assign a number (weight) to each edge of a graph we get a *weighted* graph. In many applications such a weight is being used to represent the length of an edge as distance or travel time. This must not be confused with the length of a path as defined above.

Two vertices v_i and v_j are *connected* if there exists a path from v_i to v_j . Every vertex is connected to itself. A subgraph induced by a set of vertices is called a *component* of a graph. A graph with only one component is called *connected*, otherwise it is *disconnected*. If the removal of a vertex v would disconnect the graph, then v is called an *articulation point*. A *block* is a graph without any articulation point. If the removal of an edge e would disconnect the graph then this edge is called a *cut-edge*.

Figure 54 shows an example of a connected and disconnected graph. Graph H has two components. The vertices $v_1, v_2 \in G$ are articulation points; $e_1 \in G$ is a cut-edge.

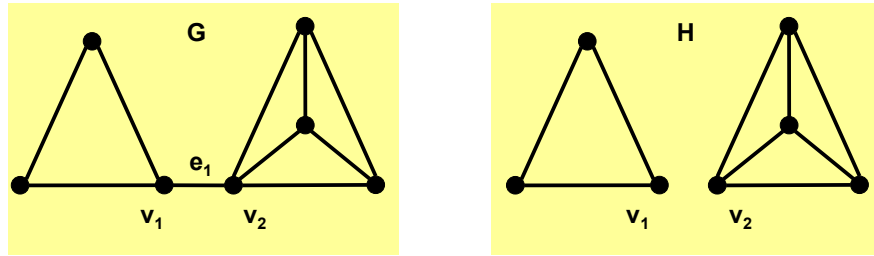


Figure 54. Connected (G) and disconnected (H) graph

11.2 Important Classes of Graphs

11.2.1 Directed Graph

If for every edge we assign one of its vertices as start point the graph becomes a *directed* graph (or *digraph*). We draw the edges of a digraph with arrows indicating their direction. A directed graph without cycles is called a *directed acyclic graph* (or DAG). DAG's play an important role in the representation of partially ordered sets. Digraphs are used to represent transportation or flow problems. Figure 55 shows two directed graphs. Graph G contains a cycle; H is a DAG.

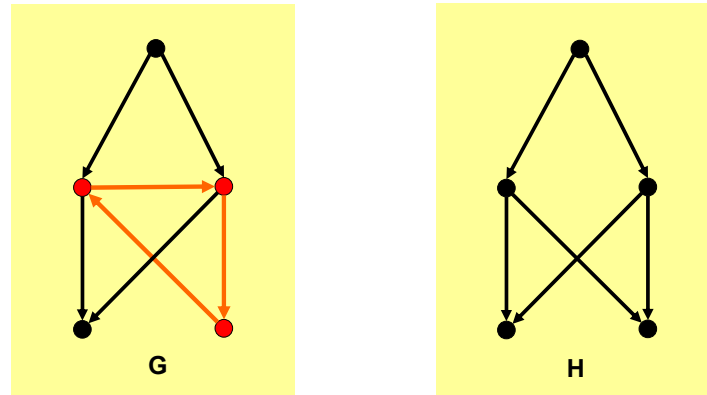


Figure 55. Directed graphs

In a digraph an edge (v_i, v_j) is said to be *incident from* v_i and *incident to* v_j . The number of edges incident from a vertex v is called the *out-degree* $d^+(v)$, the number of edges incident to v is the *in-degree* $d^-(v)$. A digraph is *symmetric* if for every edge (v_i, v_j) there is also an edge (v_j, v_i) . A digraph is *balanced* if for every vertex v the out-degree is equal to the in-degree, i.e., $d^+(v) = d^-(v)$.

11.2.2 Planar Graph

An important class of graphs is the planar graphs. A graph is *planar* if it can be drawn on a plane surface without intersecting edges²⁵. Such a representation divides the plane into connected regions (or *faces*). The faces are bound by edges of the graph. One face encloses the graph. This face is often called the exterior face.

²⁵ This class of graphs plays an important role in the structuring of two-dimensional spatial data sets for GIS.

A planar graph in the real drawing plane corresponds to a 2-dimensional cell complex, where the vertices correspond to the 0-cells, the edges to the 1-cells, and the faces to the 2-cells. Clearly, this cannot be extended to higher dimensions.

Figure 56 shows a planar graph. This graph has four vertices $V = \{v_1, v_2, v_3, v_4\}$, six edges $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$, and four faces $F = \{f_1, f_2, f_3, f_4\}$. Face f_4 is the exterior face.

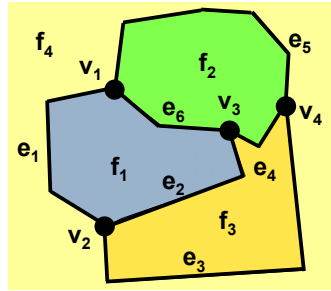


Figure 56. Planar graph

EULER's formula connects the number of vertices, edges and faces. It states that for every connected planar graph with n vertices, e edges, and f faces we have

$$n - e + f = 2$$

If we do not count the exterior face the formula changes to

$$n - e + f = 1$$

Example 116. For the planar graph in Figure 56 with four vertices, six edges, and four faces we have $4 - 6 + 4 = 2$.

For every planar graph G we can construct a graph G^* whose vertices are the regions of G ; the edges represent the adjacency of faces, i.e., there is an edge connecting two vertices of G^* if the two corresponding faces of G are adjacent. The edge is drawn crossing the bounding edge of the faces in G . Such a graph is called the *dual* graph. It is again planar. Figure 57 shows the planar dual graph of the graph in Figure 56.

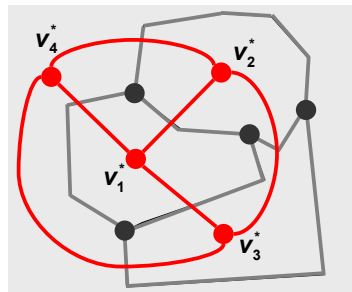


Figure 57. Dual graph

11.3 Representation of Graphs

For many computational purposes we need efficient data structures and algorithms to represent and traverse graphs. The best known structures to represent a graph are adjacency matrices and adjacency lists.

Given a graph $G = (V, E)$ with n vertices an *adjacency matrix* is an $n \times n$ matrix A , such that:

$$A(i, j) = \begin{cases} 1 & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases}$$

For an undirected graph $A(i, j) = A(j, i)$. For a digraph A is usually asymmetric.

Figure 58 shows an undirected graph G_1 and a directed graph G_2 .

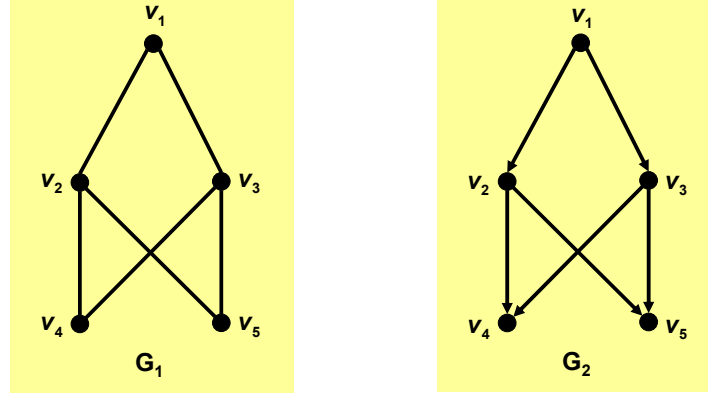


Figure 58. Undirected and directed graph

If we sort the columns and rows from v_1 to v_5 the adjacency matrices for G_1 and G_2 are written as:

$$A(G_1) = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} \quad A(G_2) = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

An *adjacency list* L shows for every vertex the vertices adjacent to it. The adjacency lists for the graphs in Figure 58 are:

$$L(G_1): \begin{cases} v_1: & v_2, v_3 \\ v_2: & v_1, v_4, v_5 \\ v_3: & v_1, v_4, v_5 \\ v_4: & v_2, v_3 \\ v_5: & v_2, v_3 \end{cases} \quad L(G_2): \begin{cases} v_1: & v_2, v_3 \\ v_2: & v_4, v_5 \\ v_3: & v_4, v_5 \\ v_4: & - \\ v_5: & - \end{cases}$$

We see easily that the storage requirement for adjacency matrices is usually higher than adjacency lists.

For directed acyclic graphs it is often more convenient to represent also the transitive relationships in the graph. This means that if we have $(v_i, v_j) \in E$ and $(v_j, v_k) \in E$ then we also show the relationship (v_i, v_k) in the matrix or list. The relationships (v_i, v_i) are trivially contained in the transitive closure. This is called the *transitive closure* of the graph. For the directed acyclic graph G_2 in Figure 58 we represent the transitive closure as:

$$A(G_2) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad L(G_2): \begin{array}{l} v_1: v_2, v_3, v_4, v_5 \\ v_2: v_4, v_5 \\ v_3: v_4, v_5 \\ v_4: - \\ v_5: - \end{array}$$

Based on the representation of a graph by a matrix or a list, efficient algorithms can be formulated to traverse the graph. Traversal means that from a given start point all other vertices of the graph are visited. One of the best known graph traversal algorithms is the *depth-first-search* (DFS). It works in the following way:

Starting from a given vertex v visit an adjacent vertex that has not yet been visited.

If no such vertex can be found then return to the vertex visited just before v and repeat step 1.

11.4 Eulerian and Hamiltonian Tours, Shortest Path Problem

As mentioned above we often want to traverse a graph in a particular manner. When we do not distinguish between path and cycle we will talk about a *tour*.

A tour through a graph in which every edge is traversed exactly once is called an *Eulerian* tour. If we traverse the graph visiting each vertex exactly once we call this a *Hamiltonian* tour. The shortest path problem is about finding the shortest path between two given vertices in a graph.

11.4.1 Eulerian Graphs

An *Eulerian graph* is an undirected graph or digraph containing an Eulerian circuit. The following statements can be proven for undirected graphs and digraphs:

An undirected graph contains an Eulerian circuit if and only if it is connected and the number of vertices with odd degree is 0.

An undirected graph contains an Eulerian path if and only if it is connected and the number of vertices with odd degree is 2 (denoted with v_1 and v_2).

A digraph contains an Eulerian circuit if and only if it is connected and balanced.

A digraph contains an Eulerian path if and only if it is connected and for the degrees of the vertices we have:

$$d^+(v) = d^-(v) \quad \text{for all } v \neq v_1 \text{ or } v_2$$

$$d^+(v_1) = d^-(v_1) + 1$$

$$d^-(v_2) = d^+(v_2) + 1$$

Example 117. When we recall the Königsberg bridge problem, we see that the question is whether there exists an Eulerian circuit in the graph of Figure 51. The graph is connected. However, there are four vertices with odd degree. Therefore, the problem cannot be solved.

Example 118. A famous problem in graph theory is the *Chinese postman problem*. In its colloquial form it is about a postman who has to deliver the mail. In order to be more efficient the question is whether he can traverse the street network of his town in such a

way that starting at the post office he walks every street not more than once before he returns to the office. In graph theoretic terms we are checking whether there is an Eulerian cycle in the graph of the street network.

11.4.2 Hamiltonian Tours

A graph is called *Hamiltonian* if it contains a Hamiltonian circuit. Unlike with Eulerian graphs we do not have a simple way of determining whether a graph is Hamiltonian. All known algorithms to find a Hamiltonian tour in graph are either inefficient or solve the problem only through approximations.

Example 119. A generalization of the Hamiltonian cycle problem is the *traveling salesman problem*. The problem can be formulated as: Given a number of cities and the costs of traveling from one city to the other, what is the cheapest roundtrip that visits every city and then returns to the starting city? The cities are the vertices of a weighted graph. There are no efficient algorithms to solve the problem. Good approximations exist, however.

11.4.3 Shortest Path Problem

The shortest path problem can be formulated as follows. Given are a weighted graph with $f: E \rightarrow \mathbb{R}$ assigning weights to the edges, and two vertices v_1 and v_2 . Find a path P from v_1 to v_2 such that for all edges $e \in P$ of the path $\sum_{e \in P} f(e)$ is minimal among all paths connecting v_1 and v_2 .

Example 120. A well known algorithm to solve the shortest path problem for a connected digraph with non-negative weights is DIJKSTRA's algorithm named after the Dutch computer scientist EDSGER DIJKSTRA.

11.5 Applications in GIS

Graphs have played an important role in GIS right from the early beginnings. The reason is that in the early days of GIS the storage of map data (or cartographic data) was the focus of interest. Early data structures for the representation of spatial data (predominantly two-dimensional) are almost exclusively based on planar graphs.

One of the famous examples is the GBF/DIME (Geographic Base File/Dual Independent Map Encoding) file of the United States Bureau of the Census. This file structure was introduced to conduct the 1970 census. The United States Geological Survey developed the DLG (Digital Line Graph) file format to store and transfer topographic base data.

In terms of data modeling planar graphs have been used extensively to represent two-dimensional spatial data. So-called *topological graphs* are the backbone of efficient representations. A topological graph is isomorphic to a planar graph embedded in \mathbb{R}^2 . The vertices are usually called nodes, the edges are called arcs and the faces are the polygons. Such a topological graph is homeomorphic to a 2-dimensional cell complex. A *network* consisting of nodes (vertices) and arcs (edges) can be considered a graph or a 1-dimensional cell complex. Therefore, planar graphs or cell complexes can be used interchangeably as long as we do not exceed the 2-dimensional space. For 3-dimensional configurations we need to turn to topology. Beside the representation of spatial features graphs play an important role in the representation and analysis of networks.

Definition 86 (Network). A *network* is a finite connected digraph in which one vertex x with $d^+(x) > 0$ is the *source* of the network, and one vertex y with $d^-(y) > 0$ is the *sink* of the network.

The network analysis functions of a GIS provide tools to find the shortest path from a A to B, to perform allocation analysis, to trace a network path, as well as location-allocation analyses.


11.6 Exercises

Exercise 38 Draw K_5 .

Exercise 39

Fuzzy Logic and GIS

Many phenomena show a degree of vagueness or uncertainty that cannot be properly expressed with crisp sets of class boundaries. Spatial features often do not have clearly defined boundaries, and concepts like “steep”, “close”, or “suitable” can better be expressed with degrees of membership to a fuzzy set than with a binary yes/no classification. This chapter introduces the basic principles of fuzzy logic, a mathematical theory that has found many applications in various domains. It can be applied whenever vague phenomena are involved.



12.1 Fuzziness

In human thinking and language we often use uncertain or vague concepts. Our thinking and language is not binary, i.e., black and white, zero or one, yes or no. In real life we add much more variation to our judgments and classifications. These vague or uncertain concepts are said to be fuzzy. We encounter *fuzziness* almost everywhere in our everyday lives.

12.1.1 Motivation

When we talk about tall people, the concept of “tall” will be depending on the context. In a society where the average height of a person is 160cm, somebody will be considered to be tall differently from a population with an average height of 180cm. In land cover analysis we are not able to draw crisp boundaries of, for instance, forest areas or grassland. Where does the grassland end and the forest start? The boundaries will be vague or fuzzy.

In real life applications we might look for a suitable site to build a house. The criteria for the area that we are looking for could be formulated as follows. The site must

- have *moderate* slope
- have *favorable* aspect
- have *moderate* elevation
- be *close to* a lake
- be *not near* a major road
- not be located in a restricted area

All the conditions mentioned above (except the one for the restricted area) are vague, but correspond to the way we express these conditions in our languages and thinking. Using the conventional approach the above mentioned conditions would be translated into crisp classes, such as

- slope less than 10 degrees
- aspect between 135 degrees and 225 degrees, or the terrain is flat
- elevation between 1,500 meters and 2,000 meters
- within 1 kilometer from a lake
- not within 300 meters from a major road

If a location falls within the given criteria we would accept it, otherwise (even if it would be very close to the set threshold) it would be excluded from our analysis. If, however, we allow degrees of membership to our classes, we can accommodate also those locations that just miss a criterion by a few meters. They will just get a low degree of membership, but will be included in the analysis. Usually, we assign a degree of membership to a class as a value between zero and one, where zero indicates no membership and one represents full membership. Any value in between can be a possible degree of membership.

12.1.2 Fuzziness versus Probability

Degrees of membership as values ranging between zero and one look very similar to probabilities, which are also given as a value between zero and one. We might be tempted to assume that fuzziness and probability are basically the same. There is, however, a subtle, yet important, difference.

Probability gives us an indication with which likelihood an event will occur. Whether it is going to happen, is not sure depending on the probability. *Fuzziness* is an indication to what degree something belongs to a class (or phenomenon). We know that the phenomenon exists. What we do not know, however, is its extent, i.e., to which degree

members of a given universe belong to the class. In the following sections we will establish the mathematical basis to deal with vague and fuzzy concepts.

12.2 Crisp Sets and Fuzzy Sets

In general set theory an element is either a member of a set or not. We can express this fact with the characteristic function for the elements of a given universe to belong to a certain subset of this universe. We call such a set a *crisp set*.

Definition 87 (Characteristic function). Let A be a subset of a universe X . The *characteristic function* χ_A of A is defined as $\chi_A : X \rightarrow \{0,1\}$ with

$$\chi_A(x) = \begin{cases} 1 & \text{iff } x \in A \\ 0 & \text{iff } x \notin A \end{cases}$$

In this way we always can clearly indicate whether an element belongs to a set or not. If, however, we allow some degree of uncertainty as to whether an element belongs to a set, we can express the membership of an element to a set by its membership function.

Definition 88 (Fuzzy set). A *fuzzy set* A of a universe X is defined by a *membership function* μ_A such that $\mu_A : X \rightarrow [0,1]$ where $\mu_A(x)$ is the *membership value* of x in A . The universe X is always a crisp set.

If the universe is a finite set $X = \{x_1, x_2, \dots, x_n\}$, then a fuzzy set A on X is expressed

as $A = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \dots + \mu_A(x_n)/x_n = \sum_{i=1}^n \mu_A(x_i)/x_i$. The term $\mu_A(x_i)/x_i$ indicates the membership value to fuzzy set A for x_i . The symbol “/” is called *separator*, Σ and “+” function as *aggregation* and *connection* of terms.

If the universe is an infinite set $X = \{x_1, x_2, \dots\}$, then a fuzzy set A on X is expressed as $A = \int_X \mu_A(x)/x$. The symbols \int and “/” function as aggregation and separator.²⁶

The *empty fuzzy set* \emptyset is defined as $\forall x \in X, \mu_{\emptyset}(x) = 0$.

For every element of the universe X we trivially have $\forall x \in X, \mu_X(x) = 1$, i.e., the universe is always crisp.

A membership function assigns to every element of the universe a degree of membership (or membership value) to a fuzzy set. This membership value must be between zero (no membership) and one (definite membership). All other values between zero and one indicate to which degree an element belongs to the fuzzy set. It is important to note that the membership degree of 1 does not need to be obtained for members of a fuzzy set.

Example 121. Let us take three persons A, B, and C and their respective heights as 185cm (A), 165cm (B) and 186cm (C). We want to assign the different persons to classes for short, average, and tall people, respectively.

²⁶ Note that the symbols Σ , $+$, and \int are not to be interpreted in their usual meaning as sum, addition, and integral.

If we take a crisp classification and set the class boundaries to $(-, 165]$ for short, $(165, 185]$ for average, and $(185, -)$ for tall, we see that A falls into the average class, B into the short class, and C into the tall class. We also see that A is nearly as tall as C, and yet they fall into different classes. The characteristic functions of the three classes are displayed in Table 15.

Table 15. Characteristic function for height classes

	Short	Average	Tall
A	0	1	0
B	1	0	0
C	0	0	1

When we choose a fuzzy set approach, we need to define three membership functions for the three classes, respectively (Figure 59).

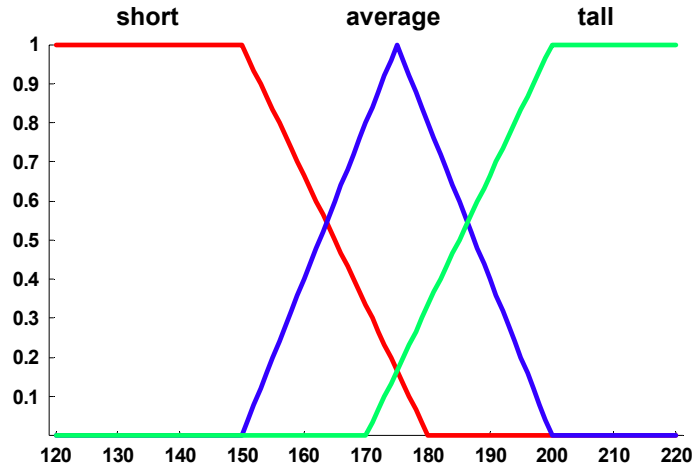


Figure 59. Membership functions for “short”, “average”, and “tall”

For *short* we select a linear membership function that produces a membership value of one for persons shorter than 150cm and decreases until it reaches zero at 180cm.

The membership function for the *average* class produces values equal zero for persons shorter than 150cm, it then increases until it reaches one at 175cm. From there it decreases until it reaches zero at 200cm.

The membership function for the *tall* class is zero up to 170cm. From there it increases until it reaches one at 200cm. The membership values for the three persons to the three classes are given in Table 16.

Table 16. Membership values for the height classes

	Short	Average	Tall
A	0.00	0.60	0.50
B	0.50	0.60	0.00
C	0.00	0.56	0.53

Using the fuzzy set approach we can much better express the fact that A and C are nearly the same height and that both have a higher degree of membership to the average class than to short or tall, respectively.

12.3 Membership Functions

The selection of a suitable membership function for a fuzzy set is one of the most important activities in fuzzy logic. It is the responsibility of the user to select a function that is a best representation for the fuzzy concept to be modeled. The following criteria are valid for all membership functions:

- The membership function must be a real valued function whose values are between 0 and 1.
- The membership values should be 1 at the center of the set, i.e., for those members that definitely belong to the set.
- The membership function should fall off in an appropriate way from the center through the boundary.
- The points with membership value 0.5 (crossover point) should be at the boundary of the crisp set, i.e., if we would apply a crisp classification, the class boundary should be represented by the crossover points.

We know two types of membership functions: (i) linear membership functions and (ii) sinusoidal membership functions. Figure 60 shows the linear membership function. This function has four parameters that determine the shape of the function. By choosing proper values for a , b , c , and d , we can create S-shaped, trapezoidal, triangular, and L-shaped membership functions.

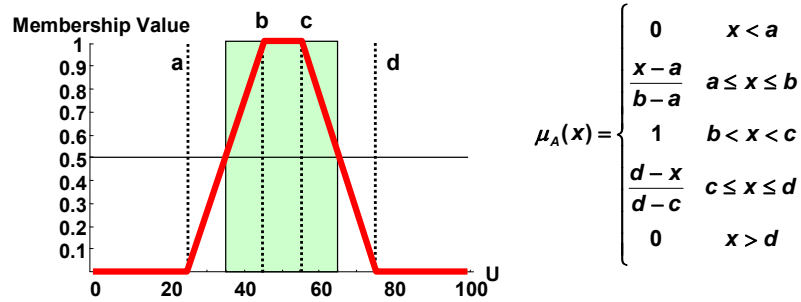


Figure 60. Linear membership function

If a rounded shape of the membership function is more appropriate for our purpose we should choose a sinusoidal membership function (Figure 61). As with linear membership functions we can achieve S-shaped, bell-shaped, and L-shaped membership functions by proper selection of the four parameters.

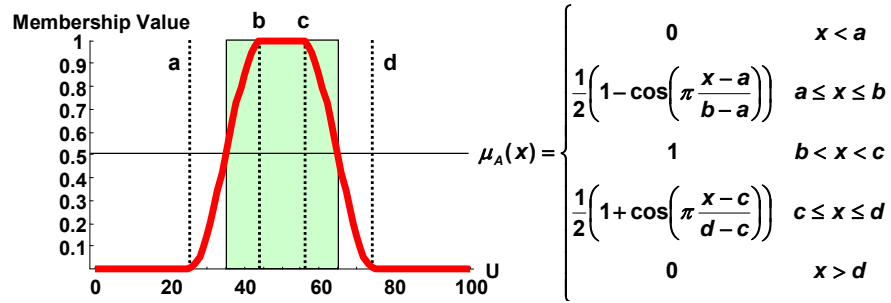


Figure 61. Sinusoidal membership function

A special case of the bell-shaped membership functions is the Gaussian function derived from the probability density function of the normal distribution with two parameters c (mean) and σ (standard deviation). Although this membership function is derived from probability theory, it is used here as a membership function for a fuzzy set.

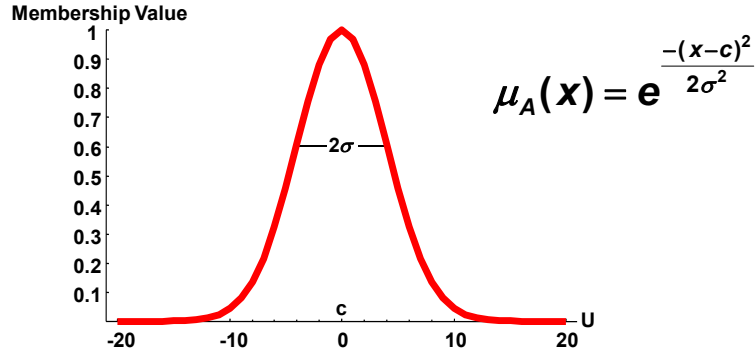


Figure 62. Gaussian membership function

Example 122. The membership functions in Example 121 are linear functions with the following parameters:

$$\mu_{\text{Short}}(x) = \begin{cases} 1 & x \leq 150 \\ \frac{180-x}{30} & 150 < x \leq 180 \\ 0 & x > 180 \end{cases}$$

$$\mu_{\text{Average}}(x) = \begin{cases} 0 & x \leq 150 \\ \frac{x-150}{25} & 150 < x \leq 175 \\ \frac{200-x}{25} & 175 \leq x \leq 200 \\ 0 & x > 200 \end{cases}$$

$$\mu_{\text{Tall}}(x) = \begin{cases} 0 & x \leq 170 \\ \frac{x-170}{30} & 170 < x \leq 200 \\ 0 & x > 200 \end{cases}$$

12.4 Operations on Fuzzy Sets

Operations on fuzzy sets are defined in a similar way as for crisp sets. However, not all rules for crisp set operations are also valid for fuzzy sets. Like for crisp sets we have subset, union, intersection, and complement. In addition, there are alternate operations for union and intersection of fuzzy sets.

Definition 89 (Support). All elements of the universe X that have a membership value greater than zero for a fuzzy set A are called the *support* of A , or $\text{supp}(A) = \{x \in X \mid \mu_A(x) > 0\}$.

Example 123. The support of the fuzzy set for short people (Example 121) is those persons who are shorter than 150cm.

Definition 90 (Height). The *height* of a fuzzy set A is the largest membership value in A , written as $\text{hgt}(A)$. If $\text{hgt}(A) = 1$ then the set is called *normal*.

Example 124. The height of the fuzzy sets Short, Average, and Tall is 1. They are all normal fuzzy sets.

We can always normalize a fuzzy set by dividing all its membership values by the height of the set.

Definition 91 (Equality). Two fuzzy sets A and B are *equal* (written as $A = B$) if for all members of the universe X their membership values are equal, i.e., $\forall x \in X, \mu_A(x) = \mu_B(x)$.

Subsets in fuzzy sets are defined by fuzzy set inclusion.

Definition 92 (Inclusion). A fuzzy set A is *included* in a fuzzy set B (written as $A \subseteq B$) if for every element of the universe the membership values for A are less than or equal to those of B , i.e., $\forall x \in X, \mu_A(x) \leq \mu_B(x)$.

When we look at the graph of the membership functions a fuzzy set A will be included in fuzzy set B when the graph of A is completely covered by the graph of B (Figure 63).

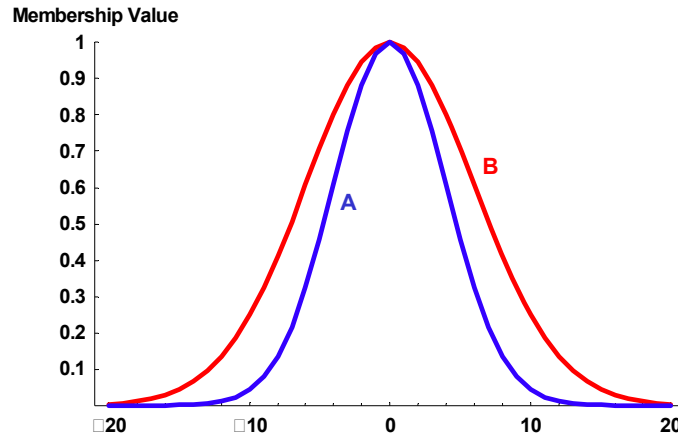


Figure 63. Set inclusion

For the union of two fuzzy sets we have more than one operator. The most common ones are presented here.

Definition 93 (Union). The *union* of two fuzzy sets A and B can be computed for all elements of the universe X by one of the three operators:

1. $\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$
2. $\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$
3. $\mu_{A \cup B}(x) = \min(1, \mu_A(x) + \mu_B(x))$

The max-operator is a *non-interactive* operator in the sense that the membership values of both sets do not interact with each other. In fact, one set could be completely ignored in a union operation when it is included in the other. The two other operators are called *interactive*, because the membership value of the union is determined by the membership values of both sets.

Example 125. Figure 64 illustrates the union operators for the fuzzy sets Short and Average from Example 121.

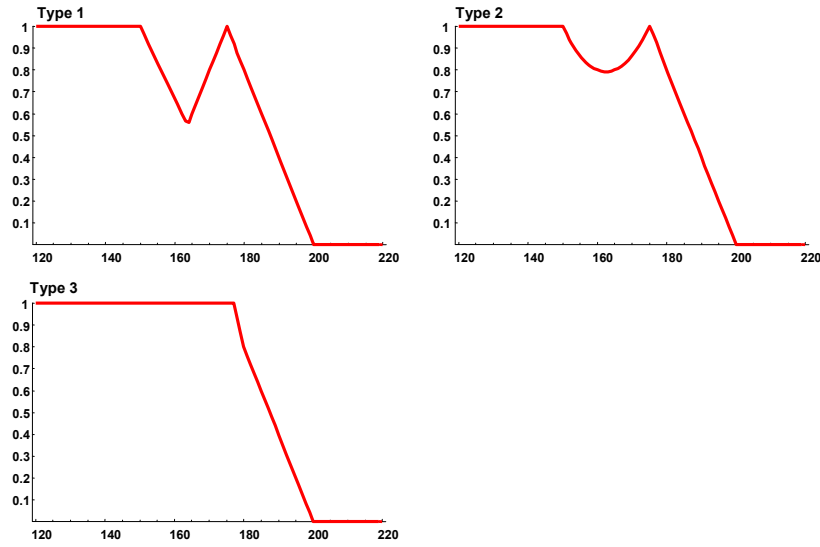


Figure 64. Fuzzy set union operators

Definition 94 (Intersection). The *intersection* of two fuzzy sets A and B can be computed for all elements of the universe X by one of the three operators:

1. $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$
2. $\mu_{A \cap B}(x) = \mu_A(x) \cdot \mu_B(x)$
3. $\mu_{A \cap B}(x) = \max(0, \mu_A(x) + \mu_B(x) - 1)$

The min-operator is non-interactive, the two others are interactive operators as explained above.

Example 126. Figure 65 illustrates the intersection of the fuzzy sets Short and Average from Example 121.

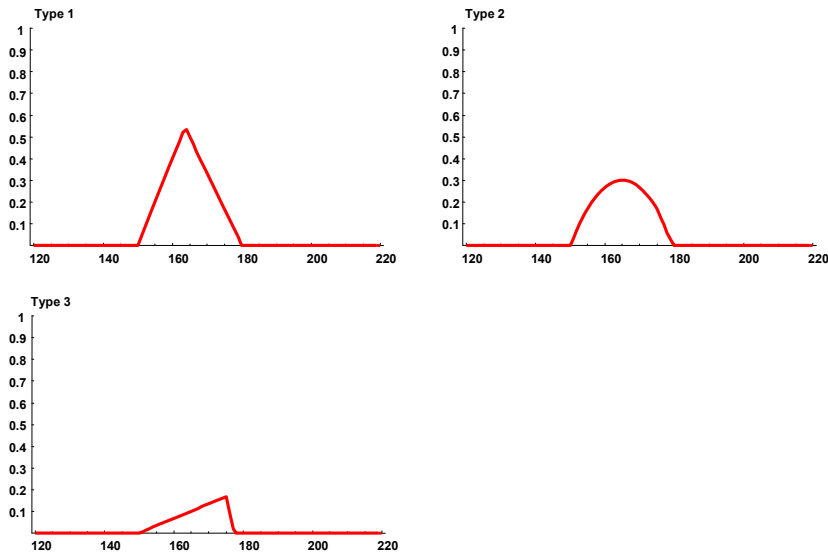


Figure 65. Fuzzy set intersection

Definition 95 (Complement). The *complement* of a fuzzy set A in the universe X is defined as $\forall x \in X, \mu_{\bar{A}}(x) = 1 - \mu_A(x)$.

Example 127. Figure 66 shows the fuzzy set Average from Example 121 and its complement.

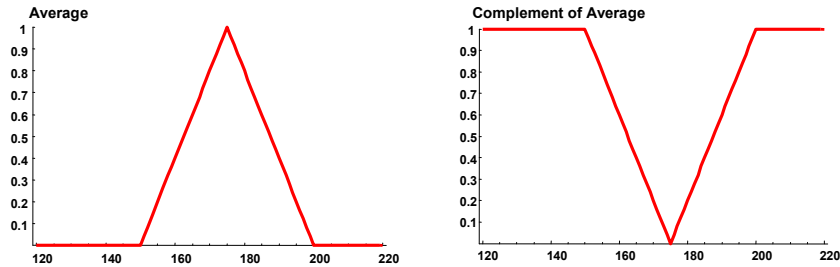


Figure 66. Fuzzy set and its complement

Many rules for set operations are valid for both crisp and fuzzy sets. Table 17 shows the rules that are valid for both.

Table 17. Rules for set operations valid for crisp and fuzzy sets

1.	$A \cup A = A$	idempotent law
2.	$A \cap A = A$	
3.	$(A \cup B) \cup C = A \cup (B \cup C)$	associativity
4.	$(A \cap B) \cap C = A \cap (B \cap C)$	
5.	$A \cup B = B \cup A$	commutativity
6.	$A \cap B = B \cap A$	
7.	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	distributivity
8.	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	
9.	$\overline{A \cup B} = \bar{A} \cap \bar{B}$	DE MORGAN's law
10.	$\overline{A \cap B} = \bar{A} \cup \bar{B}$	
11.	$\overline{\bar{A}} = A$	double complement

Table 18 shows those rules that in general are valid for crisp sets but not for fuzzy sets.

Table 18. Rules valid only for crisp sets

1.	$A \cup \bar{A} = X$	law of the excluded middle
2.	$A \cap \bar{A} = \emptyset$	law of contradiction

Figure 67 illustrates that the law of the excluded middle and the law of contradiction does not generally hold for fuzzy sets.

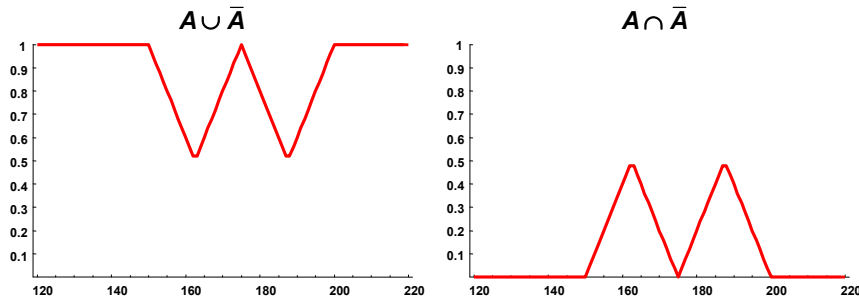


Figure 67. Law of the excluded middle and law of contradiction for fuzzy set Average.

12.5 Alpha-Cuts

If we wish to know all those elements of the universe that belong to a fuzzy set and have at least a certain degree of membership, we can use α -level sets.

Definition 96 (α -Cut). A (weak) α -cut (or α -level set) A_α with $0 < \alpha \leq 1$ is the set of all elements of the universe such that $A_\alpha = \{x \in X \mid \mu_A(x) \geq \alpha\}$. A strong α -cut $A_{\bar{\alpha}}$ is defined as $A_{\bar{\alpha}} = \{x \in X \mid \mu_A(x) > \alpha\}$.

Example 128. The 0.8-cut of the fuzzy set Tall contains all those persons who are 194cm or taller. With α -level sets we can identify those members of the universe that typically belong to a fuzzy set.

12.6 Linguistic Variables and Hedges

In mathematics variables usually assume numbers as values. A *linguistic variable* is a variable that assumes linguistic values which are words (*linguistic terms*). If, for example, we have the linguistic variable “height”, the linguistic values for height could be “short”, “average”, and “tall”. These linguistic values possess a certain degree of uncertainty or vagueness that can be expressed by a membership function to a fuzzy set. Often, we modify a linguistic term by adding words like “very”, “somewhat”, “slightly”, or “more or less” and arrive at expressions such as “very tall”, “not short”, or “somewhat average”.

Such modifiers are called *hedges*. They can be expressed with operators applied to the fuzzy sets representing linguistic terms (see Table 19).

Table 19. Operators for hedges

Operator	Expression
Normalization	$\mu_{\text{norm}(A)}(x) = \frac{\mu_A(x)}{\text{hgt}(\mu_A)}$
Concentration	$\mu_{\text{con}(A)}(x) = \mu_A^2(x)$
Dilation	$\mu_{\text{dil}(A)}(x) = \sqrt{\mu_A(x)}$
Negation	$\mu_{\text{not}(A)}(x) = \mu_{\bar{A}}(x) = 1 - \mu_A(x)$
Contrast intensification	$\mu_{\text{int}(A)}(x) = \begin{cases} 2\mu_A^2(x) & \text{if } \mu_A(x) \in [0, 0.5] \\ 1 - 2(1 - \mu_A(x))^2 & \text{otherwise} \end{cases}$

The following Table 20 shows the models being used to represent hedges for linguistic terms.

Table 20. Hedges and their models

Hedge	Operator
very A	$\text{con}(A)$
more or less A (fairly A)	$\text{dil}(A)$
plus A	$A^{1.25}$
not A	$\text{not}(A)$
slightly A	$\text{int}(\text{norm}(\text{plus } A \cap \text{not}(\text{very } A)))$

Example 129. Figure 68 shows the membership functions for Tall, Very Tall, and Very Very Tall.

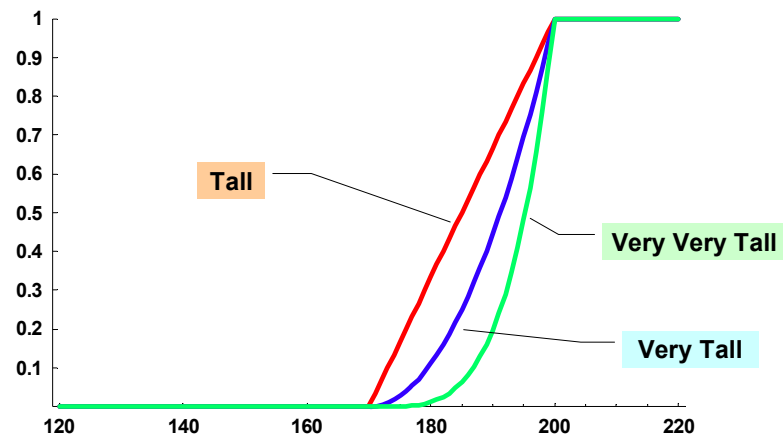


Figure 68. Membership functions for Tall, Very Tall, and Very Very Tall

Example 130. Figure 69 shows the membership functions for Tall and Not Very Tall.

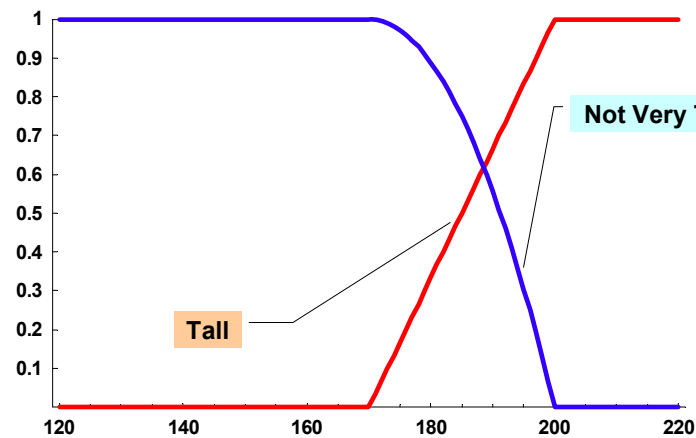


Figure 69. Membership function for Tall and Not Very Tall

Example 131. Figure 70 shows the membership functions for Tall and Slightly Tall.

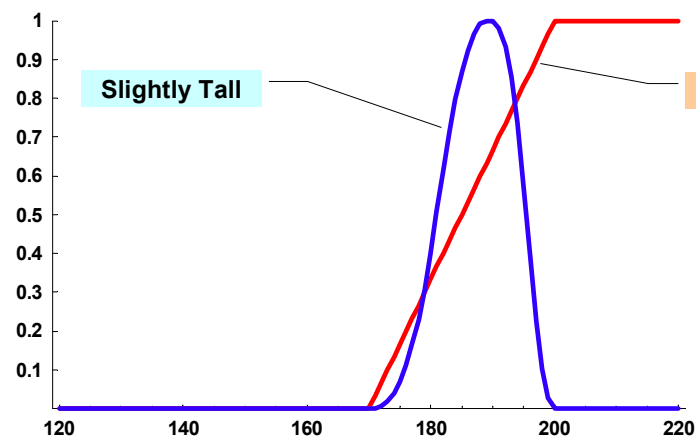


Figure 70. Membership function for Tall and Slightly Tall

12.7 Fuzzy Inference

In binary logic we have only two possible values for a logical variable, true or false, 1 or 0. As we have seen in this chapter, many phenomena can be better represented by fuzzy

sets than by crisp classes. Fuzzy sets can also be applied to reasoning when vague concepts are involved.

In binary logic reasoning is based on either deduction (*modus ponens*) or induction (*modus tollens*). In fuzzy reasoning we use a *generalized modus ponens* which reads as

Premise₁: If x is A then y is B
 Premise₂: x is A'
 Conclusion: y is B'

Here, A , B , A' , and B' are fuzzy sets where A' and B' are not exactly the same as A and B .

Example 132. Consider the generalized *modus ponens* for temperature control:

Premise₁: If the temperature is *low* then set the heater to *high*
 Premise₂: Temperature is *very low*
 Conclusion: Set the heater to *very high*

With logic inference we normally have more than one rule. In fact, the number of rules can be rather large. We know several methods for fuzzy reasoning.

12.7.1 MAMDANI'S Direct Method

Here, we discuss the methods known as MAMDANI'S direct method. It is based on a generalized *modus ponens* of the form

$$p \Rightarrow q: \begin{cases} \text{If } x \text{ is } A_1 \text{ and } y \text{ is } B_1 \text{ then } z \text{ is } C_1 \\ \text{If } x \text{ is } A_2 \text{ and } y \text{ is } B_2 \text{ then } z \text{ is } C_2 \\ \vdots \\ \text{If } x \text{ is } A_n \text{ and } y \text{ is } B_n \text{ then } z \text{ is } C_n \end{cases}$$

$$\frac{p_1: \quad \quad \quad x \text{ is } A', y \text{ is } B'}{q_1: \quad \quad \quad z \text{ is } C'}$$

Premise₁ becomes a set of rules as illustrated in Figure 71. A , B , and C are fuzzy sets, x and y are *premise variables*, z is the *consequence variable*.²⁷

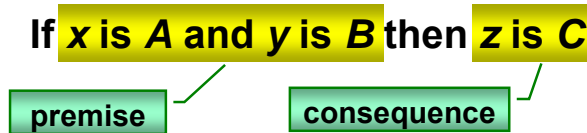


Figure 71. Inference rule in MAMDANI'S direct method

The reasoning process is then straightforward according to the following procedure. Let x_0 and y_0 be the input for the premise variables.

1. Apply the input values to the premise variables for every rule and compute the minimum of $\mu_A(x_0)$ and $\mu_B(y_0)$:

²⁷ There can be more than two premise variables to express complex rules. The procedure can be extended to this case without any problems.

$$\text{Rule}_1: m_1 = \min(\mu_{A_1}(x_0), \mu_{B_1}(y_0))$$

$$\text{Rule}_2: m_2 = \min(\mu_{A_2}(x_0), \mu_{B_2}(y_0))$$

$$\vdots$$

$$\text{Rule}_n: m_n = \min(\mu_{A_n}(x_0), \mu_{B_n}(y_0))$$

2. Cut the membership function of the consequence $\mu_{C_i}(z)$ at m_i :

$$\text{Conclusion of rule}_1: \mu_{C'_1}(z) = \min(m_1, \mu_{C_1}(z)) \quad \forall z \in C_1$$

$$\text{Conclusion of rule}_2: \mu_{C'_2}(z) = \min(m_2, \mu_{C_2}(z)) \quad \forall z \in C_2$$

$$\vdots$$

$$\text{Conclusion of rule}_n: \mu_{C'_n}(z) = \min(m_n, \mu_{C_n}(z)) \quad \forall z \in C_n$$

3. Compute the final conclusion by determining the union of all individual conclusions from step 2:

$$\mu_C(z) = \max(\mu_{C'_1}(z), \mu_{C'_2}(z), \dots, \mu_{C'_n}(z))$$

The result of the final conclusion is a fuzzy set. For practical reasons we need a definite value for the consequence variable. The process to determine this value is called *defuzzification*. There are several methods to defuzzify a given fuzzy set. One of the most common is the *center of gravity* (or center of area).

For a discrete fuzzy set the center of area is computed as

$$z_0 = \frac{\sum \mu_C(z) \cdot z}{\sum \mu_C(z)}$$

For a continuous fuzzy set this becomes

$$z_0 = \frac{\int \mu_C(z) \cdot z dz}{\int \mu_C(z) dz}$$

Example 133. Given the speed of a car and the distance to a car in front of it, we would like to determine whether we should break, maintain the speed, or accelerate. Assume the following set of rules for the given situation:

- Rule 1 If the distance between the cars is short and the speed is low then maintain speed
- Rule 2 If the distance between the cars is short and the speed is high then reduce speed
- Rule 3 If the distance between the cars is long and the speed is low then increase speed
- Rule 4 If the distance between the cars is long and the speed is high then maintain speed

Distance, speed, and acceleration are linguistic variables with the values “short”, “long”, “high”, “low”, and “reduce”, “maintain”, and “increase”, respectively. They can be modeled as fuzzy sets (Figure 72).

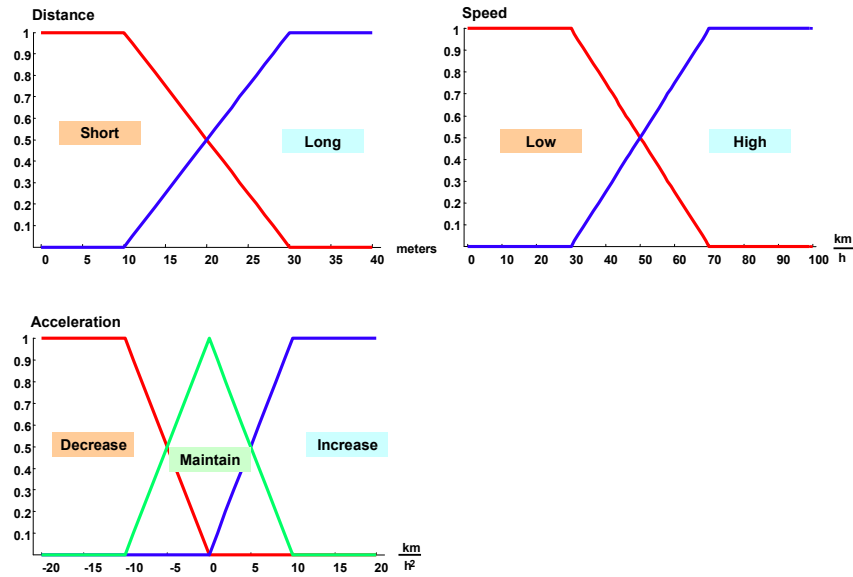


Figure 72. Fuzzy sets of the rules

With a given distance $x_0 = 15$ meters and a speed of $y_0 = 60$ km/h we perform step 1. The results are shown in Table 21.

Table 21. Fuzzy inference step 1

Rule	Short	Long	Low	High	Min
1	0.75		0.25		0.25
2	0.75			0.75	0.75
3		0.25	0.25		0.25
4		0.25		0.75	0.25

Now we must cut the membership function for the conclusion variable at the minimum values from step 1. The result is illustrated in Figure 73.

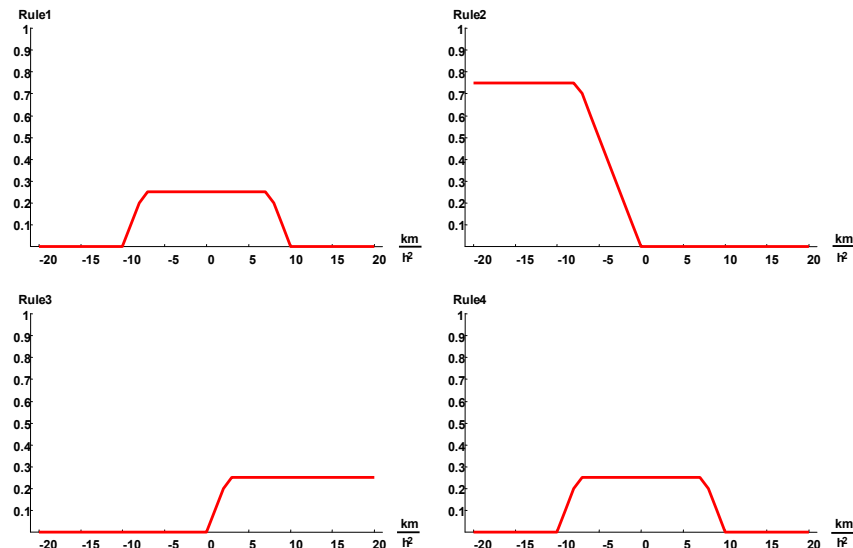


Figure 73. Fuzzy inference step 2

Finally, we must combine the individual membership functions from step 2 to the final result and defuzzify it. The union of the four membership functions is displayed in Figure 74. The final value after defuzzification is -5.46 and is indicated by the blue dot. The conclusion of this fuzzy inference is that when the distance between the cars is 15 meters and the speed is 60 km/h, then we have to break gently to reduce the speed.

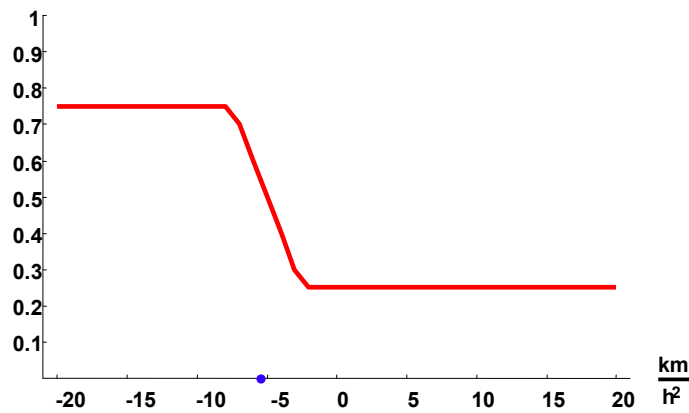


Figure 74. Fuzzy inference final result

12.7.2 Simplified Method

Often, the defuzzification process is too time-consuming and complicated. An alternative approach is the simplified method where the conclusion is a real value c instead of a fuzzy set. It is based on a generalized *modus ponens* of the form:

$$p \Rightarrow q: \begin{cases} \text{If } x \text{ is } A_1 \text{ and } y \text{ is } B_1 \text{ then } z \text{ is } c_1 \\ \text{If } x \text{ is } A_2 \text{ and } y \text{ is } B_2 \text{ then } z \text{ is } c_2 \\ \vdots \\ \text{If } x \text{ is } A_n \text{ and } y \text{ is } B_n \text{ then } z \text{ is } c_n \end{cases}$$

$$\frac{p_1: \quad \quad \quad x \text{ is } A', y \text{ is } B'}{q_1: \quad \quad \quad z \text{ is } c'}$$

Premise₁ becomes a set of rules as illustrated in Figure 75. The premise variables are fuzzy sets; the conclusion is a real number (fuzzy singleton).

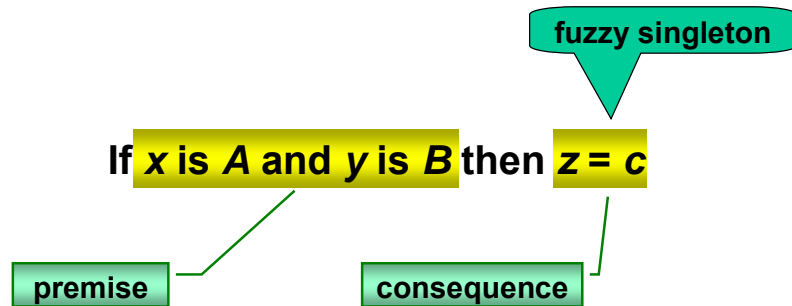


Figure 75. Simplified Method

The reasoning process is then straightforward in analogy to the previous method with the difference that the result is not a fuzzy set that needs to be defuzzified but we can compute the final result directly after step 2 in the algorithm.

The algorithm works as outlined in the following procedure. Let x_0 and y_0 be the input for the premise variables.

1. Apply the input values to the premise variables for every rule and compute the minimum of $\mu_{A_i}(x_0)$ and $\mu_{B_i}(y_0)$:

$$\text{Rule}_1: m_1 = \min(\mu_{A_1}(x_0), \mu_{B_1}(y_0))$$

$$\text{Rule}_2: m_2 = \min(\mu_{A_2}(x_0), \mu_{B_2}(y_0))$$

$$\vdots$$

$$\text{Rule}_n: m_n = \min(\mu_{A_n}(x_0), \mu_{B_n}(y_0))$$

2. Compute the conclusion value per rule as:

$$\text{Conclusion of rule}_1: c'_1 = m_1 \cdot c_1$$

$$\text{Conclusion of rule}_2: c'_2 = m_2 \cdot c_2$$

$$\vdots$$

$$\text{Conclusion of rule}_n: c'_n = m_n \cdot c_n$$

3. Compute the final conclusion as:

$$c' = \frac{\sum_{i=1}^n c'_i}{\sum_{i=1}^n m_i}$$

Example 134. Given the slope and the aspect maps of a region and the following set of rules, we can conduct a risk analysis based on degrees of risk ranging from 1 (low risk) to 4 (very high risk). The fuzzy sets for flat and steep slope are displayed in Figure 18 and Figure 19-

- Rule 1 If slope is flat and aspect is favorable then risk is 1
 Rule 2 If slope is steep and aspect is favorable then risk is 2
 Rule 3 If slope is flat and aspect is unfavorable then risk is 1
 Rule 4 If slope is steep and aspect is unfavorable then risk is 4

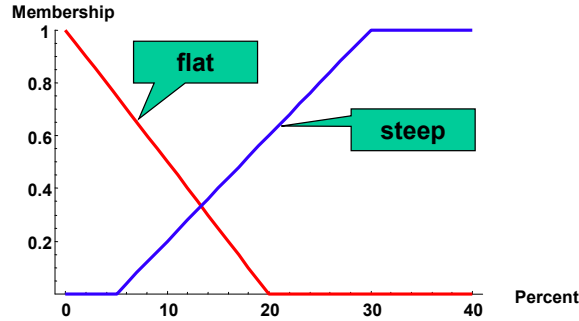


Figure 76. Membership functions for flat and steep slope

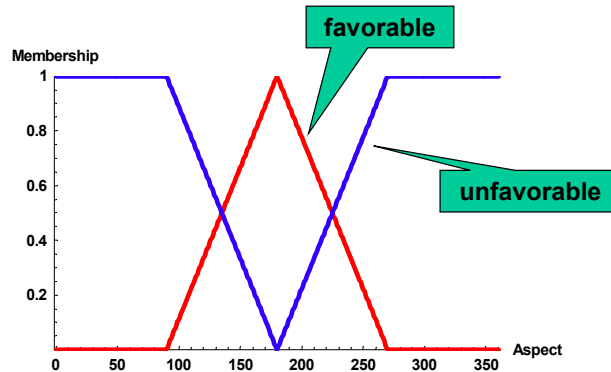


Figure 77. Membership functions for favorable and unfavorable aspect.

For a slope of 10 percent and an aspect of 180 degrees we have the following results:

	Slope (s)	Aspect (a)	Min(s,a)	Conclusions
Rule1	0.5	1	0.5	0.5
Rule2	0.2	1	0.2	0.4
Rule3	0.5	0	0	0
Rule4	0.2	0	0	0

For the final result we get $c' = \frac{0.5+0.4+0+0}{0.5+0.2+0+0} = 1.29$, which means a low risk.

12.8 Applications in GIS

Many spatial phenomena are inherently fuzzy or vague or possess indeterminate boundaries. Fuzzy logic has been applied for many areas in GIS such as fuzzy spatial analysis, fuzzy reasoning, and the representation of fuzzy boundaries. The following example illustrates how a fuzzy set can be computed from a given grid data set.

12.8.1 Objective

The objective of this analysis is to determine high elevation in the area covered by the 1 : 24,000 topographic map sheet of Boulder, Colorado

12.8.2 Fuzzy Concepts

Elevation is considered *high* when it is above 1,700 meters. We represent the features meeting the criterion as a fuzzy set with a sinusoidal membership function (Figure 78) defined as

$$\mu_{\text{high elevation}}(x) = \begin{cases} 0 & x \leq 1700 \\ \frac{1}{2} \left(1 - \cos \left(\pi \frac{x-1700}{300} \right) \right) & 1700 < x \leq 2000 \\ 1 & x > 2000 \end{cases}$$

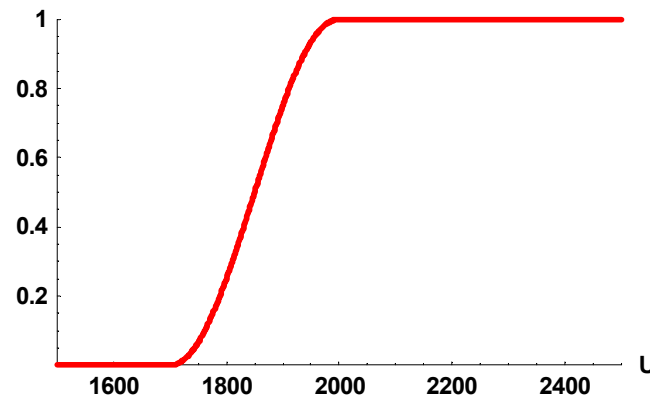


Figure 78. Membership function for “high elevation”

12.8.3 Software Approach

The 1 : 24K DEM was downloaded from the USGS and imported into ArcGIS as a grid ELEVATION. In principle, there are several ways to solve the problem: we can use ArcInfo GRID, ArcMap Spatial Analyst, or ArcView 3.x Spatial Analyst. We can even create our own fuzzy logic tool using the scripting environment of the geoprocessor in ArcGIS 9. In the following, all approaches are illustrated. The grid involved is

ELEVATION. The fuzzy set will be a grid FELEVATION whose values are between zero and one.

12.8.3.1 ArcInfo GRID

To compute the fuzzy set we use an AML script that is run from ArcInfo GRID and the DOCELL block:

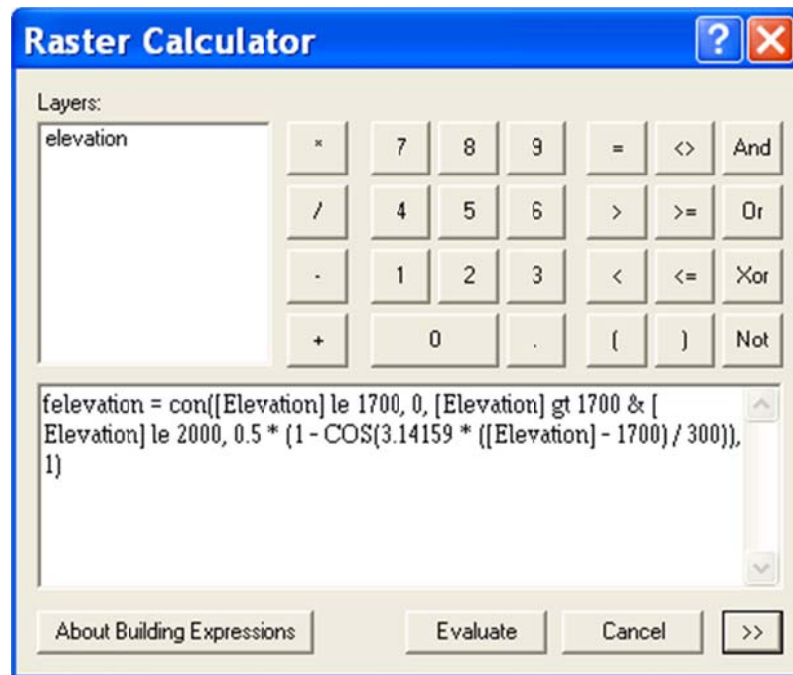
```
/*
/* high elevation
/* =====
/*
docell
  if (elevation le 1700) felevation = 0
  if (elevation gt 1700 & elevation le 2000) ~
    felevation = 0.5 * (1 - COS(3.14159 * (elevation - 1700) / 300))
  if (elevation gt 2000) felevation = 1
end
```

We can also use the GRID CON command:

```
/* high elevation
/* =====
/*
felevation = con(elevation le 1700, 0, elevation gt 1700 & elevation ~
le 2000, 0.5 * (1 - COS(3.14159 * (elevation - 1700) / 300)), 1)
```

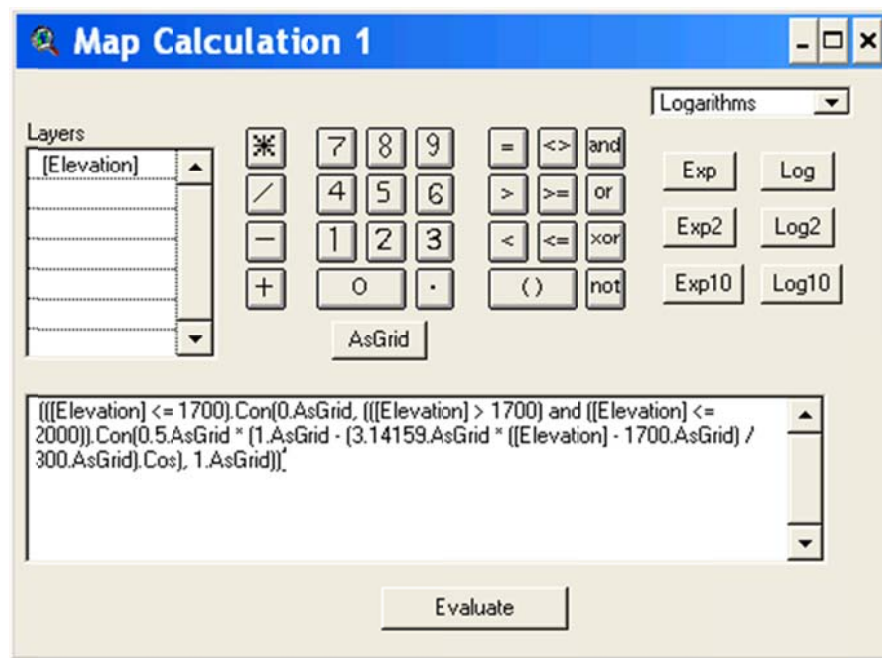
12.8.3.2 ArcMap Spatial Analyst

To solve the problem we use the raster calculator of the Spatial Analyst. The following screen dump shows the command to produce the required fuzzy set.



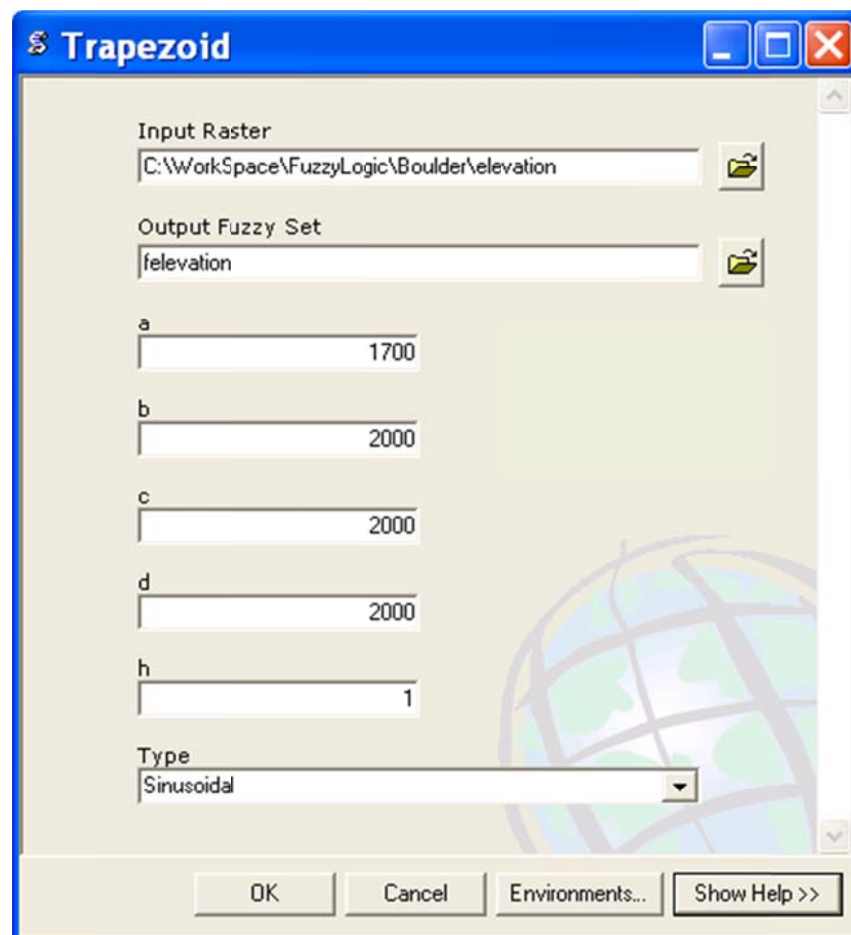
12.8.3.3 ArcView 3.x

If you do not have ArcGIS available, the same results can be achieved by using requests in the ArcView GIS Spatial Analyst map calculator. The following screen dump shows how to use the Avenue **Con** request for computing the fuzzy set for high elevation.



12.8.3.4 ArcGIS 9 Script

We have written a Python script that generates a fuzzy raster data set from a given input raster data set. This script is used here as a tool in the ArcToolbox.



12.8.4 Result

Figure 79 shows the result of the analysis with a fuzzy logic approach (left map) and a crisp approach (right map). The grid size has been set to 10 meters according to the grid cell size of the elevation model.

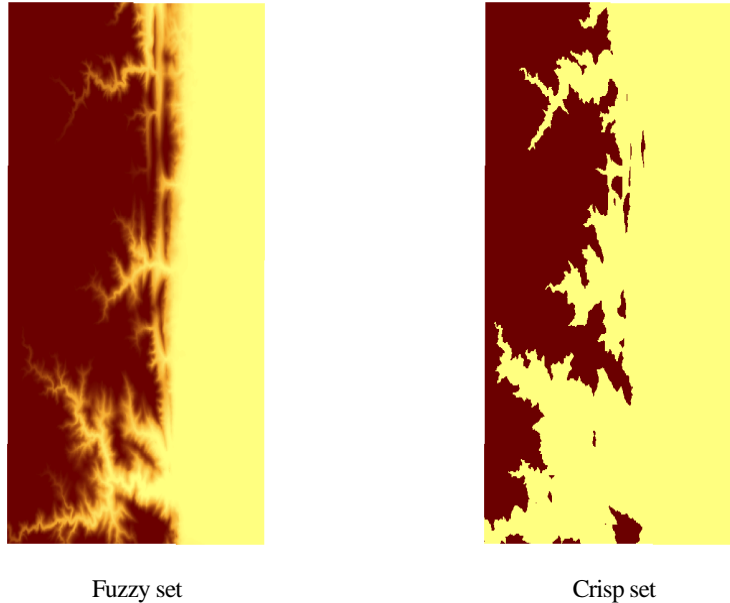


Figure 79. Analysis with a fuzzy logic approach (left) and a crisp approach (right)

12.9 Exercises

Exercise 40 Determine a linear membership function for “moderate elevation” when the ideal elevation is between 400 and 600 meters.

Exercise 41 Determine a Gaussian membership function for the aspect “south.”

Exercise 42 Use the 1:24,000 digital topographic data set of Boulder, Colorado and determine a suitable site with the following characteristics:

- (i) moderate slope
- (ii) favorable aspect
- (iii) moderate elevation
- (iv) near a lake or reservoir
- (v) not very close to a major road, and
- (vi) not in a park or military reservation.

Choose suitable membership functions for the fuzzy terms.

Exercise 43 Design a simple fuzzy reasoning system for avalanche risk in the Rocky Mountains (Boulder, CO area). The variables involved are slope, aspect, snow cover change. For simplicity we do not consider surface cover. The snow cover change must be simulated. The rules are given as:

Rule 1: If the slope is very steep and the aspect is unfavorable and the snow cover change is big then the risk is very high.


Rule 2: If the slope is moderate and the aspect is unfavorable and the snow cover change is big then the risk is moderate.

- Rule 3: If the slope is steep and the aspect is unfavorable and the snow cover change is small then the risk is low.
- Rule 4: If the slope is not steep and the aspect is unfavorable and the snow cover change is big then the risk is moderate.

Spatial Modeling

We live in a constantly changing world. What we perceive with our senses are processes, states and events that happen or exist in the real world. They may be natural or man-made, and are called real world phenomena. In our brains, we process the input from our senses, which leads to mental models, learning, cognition, and knowledge. Everything that happens in the real world and that we process through our senses leads to models of reality that we create for ourselves. Usually, we agree on common principles and interpretations derived from experience, research or learning that enable us to reach a consensus on the perception of real world phenomena.

This chapter deals with spatial modeling, a process that maps aspects of the real world to abstract models. We discuss fundamental principles of space and time and their bearing on GIS.



13.1 Real World Phenomena And Their Abstractions

In the real world, we distinguish between natural and man-made phenomena. Natural phenomena exist independently from human actions and are subject only to the laws of nature. Examples are the landscape (topography), the weather, or natural processes that shape and influence them. Man-made phenomena are objects that have come into existence by human activities through construction or building processes.

Based on these phenomena we develop high-level abstract models for particular purposes and applications. Features (abstractions of phenomena) populate these models that are usually organized in layers. Examples of such models are a cadastre, topography, soil, hydrography, land cover, or land use.

The fact that these models are abstractions of the real world can be illustrated by the example of a cadastre. Let us assume that a cadastre is a legal and organizational framework for the handling of land. It is a very important, clearly described and understood concept. Yet, we do not see a cadastre when we look around us. What we see are real world phenomena such as buildings, roads, fences around pieces of land, and people. A cadastre abstracts from certain phenomena and their relationships to create something new that is real in a given context.

Layers are an ordering principle for real world phenomena. Again, when we look around us, we do not see the world in layers. Yet, we are used to organize phenomena of the real world in such a way that we classify them according to a perceived purpose or characteristic into subsets (layers) that allow us to deal with them efficiently.

13.1.1 Spatial Data And Information

In order to conceptualize mental models of the real world, we need to categorize the phenomena we observe. These phenomena exist in space and time and have therefore a spatial (geometric) and a temporal extent. They possess certain thematic characteristics (also called attributes) that allow us not only to refer to them in terms of spatiotemporal, but also according to thematic information. The thematic information of real world phenomena is the basis for the definition of layers.

Humans perceive signals through their senses, process them and extract information that leads to knowledge and wisdom. Here, we focus on spatial information, i.e., information concerning phenomena with a spatiotemporal extent. It is obvious that thematic information is an integral part of spatial phenomena.

Data are representations of information in a computer. Spatial data refer to spatial information that we store in a computer for processing and analysis.

The following example illustrates the principle. Assume that we stand on top of a hill and look at the landscape surrounding us. What we see are meadows, fields, trees, and roads. The meadows are green with grass, the fields are yellow, the trees are green, and the road is covered with brownish-black asphalt. Our brains have processed the optical signals that we receive through our eyes and our minds recognize the phenomena observed. We also have assigned attributes to them such as color, (relative) size and we might remember that five years ago the road was not paved, and what is a field now has been a forest.

Since we want to build a land cover database of this area, we need to store a representation of the phenomena in a computer database. To achieve this we need to define features, collect (spatial and attribute) data about them and enter them into a database. A geographic information system (GIS) will be used to enter, store and maintain, process, analyze and display these data.

13.2 Concepts Of Space And Time

Space and time are two closely related concepts that have been the subject of philosophical and scientific consideration since the dawn of mankind. The space that humans live in is the three-dimensional (Euclidean) space as a frame of reference for our senses of touch and sight. Of all possible physical and mathematical spaces, this is the space that is illustrative and that we perceive as being real²⁸.

Time is a measure for change in our immediate experience. Usually, we assume time to be of a continuous linear nature extending from the past, through the present into the future.

Space and time (at least as we perceive them) are so well known and appear to be given beyond any doubt that we hardly ever contemplate their structure and characteristics in our everyday lives. When we deal with information systems that process and manipulate spatiotemporal features, we need clear and well-understood models of space and time. The following sections describe how concepts of space and time developed in Western philosophy and physics. We discuss them according to three epochs: (i) pre-Newtonian concepts, (ii) the Newtonian and classical concepts, and (iii) contemporary concepts of space and time.

13.2.1 Pre-Newtonian Concepts Of Space And Time

The concepts of space and time of this epoch are mainly dominated by the ideas of Greek philosophers about the logical conditions for things to change and the structure of the world in which change occurs.

HERACLITUS (around 500 BC) of Ephesos (western Turkey) studied the problem of change, i.e., how can the identity of things be preserved when they change. He stated that everything flows, nothing remains, and the only thing that really exists is change (processes). “Everything flows” and “We cannot step into the same river twice” are attributed to him.

At about the same time, PARMENIDES of Elea (southern Italy) developed a completely opposite philosophy of the non-existence of the void. He postulates (through deductive reasoning) that change does not exist and that the real world (the real being) is *plenum* (a solid complete compact being), immutable, without change and eternal. A void (or empty space, i.e., something non-existent) does not exist. What we perceive as change is a delusion of our senses. PARMENIDES’ ideas were further developed and “proven” by his student ZENO.

One of ZENO’s famous “proofs” that change and movement cannot exist is known as the race between Achilles and the tortoise. The tortoise gets a head-start and begins the race at point B, whereas ACHILLES starts from point A. When ACHILLES reaches point B, the tortoise already has moved on to point C. When ACHILLES reaches point C, the tortoise has moved on to another point, and so on. The lead of the tortoise gets smaller and smaller, *ad infinitum*. We get an infinite number of (smaller and smaller) leads.

The argument is now: In order to reach the tortoise, ACHILLES must catch up an infinite number of (finite in length) leads. It is impossible to run this infinite number of short distances, because ACHILLES would have to run infinitely far (or forever). Therefore, it is impossible for ACHILLES to catch the tortoise²⁹. Since we can easily catch a tortoise when we would run against it, we end up with a paradox. This proves that the assumption

²⁸ Human geographers, of course, might disagree.

²⁹ The solution of the paradox lies in the fact that an infinite series can converge to a finite value, i.e., in our case the point where ACHILLES passes the tortoise. This mathematical result was, however, not known until the 17th century.

that movement and change are real, leads to contradictions. Therefore, movement and change are impossible.

DEMOCRITUS (460–370 BC) did not accept the non-existence of change as postulated by PARMENIDES. Space is an absolute and empty entity existing independently from the atoms that fill the space. Atoms are indivisible real things; they are immutable and eternal, and have different size and weight. There is no empty space within atoms. An atom is a *plenum*. Objects are formed as a collection of atoms. The importance of the Atomist theory is evident in today's modern particle physics.

Greek mathematics was strongly dominated by the Pythagorean number theoretic approach. It was essentially arithmetic based on (philosophical) properties of numbers, counting, and the ratios between numbers. The discovery of irrational numbers such as $\sqrt{2}$ (the length of the diagonal in the unit square) shook the foundation of Greek mathematics that was based on counting in natural units. The need for a truly geometrical description of the world became apparent.

The great philosopher PLATO (427–347 BC) and one member of his school, EUCLID (at 300 BC), laid the foundation for a new geometric modeling of the real world. This geometric model of matter is based on right triangles as atoms and solids built from these triangles. Matter consists of four elements: earth, air, fire, and water. Each element is made of particles, i.e., solids (see Figure 80) that in turn are made from triangles. Transformations between the elements fire, air and water are possible through geometric transformations. Earth cannot be transformed.

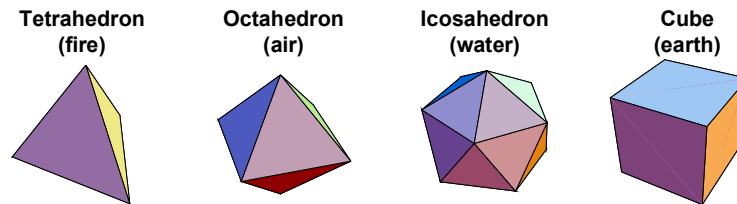


Figure 80. Platonic solids as building blocks of matter

In his book *The Elements*, EUCLID developed a mathematical theory of geometry that remained valid until the late 19th century. Euclidean geometry was considered a true description of our physical world until it was discovered that many consistent geometric systems are possible, some of them non-Euclidean, and that geometry is not a description of the world but yet another formal mathematical system with no necessary reference to real world phenomena.

13.2.2 Classical Concepts Of Space And Time

The time between the classical Greek period and the rise of modern science was dominated by the philosophy and teaching of ARISTOTLE (384–322 BC). According to his ideas, empty space is impossible, and time is the measure of motion with regard to what is earlier and later. Space is defined as the limit of the surrounding body towards what is surrounded.

Following from this approach space can be conceptualized in two possible ways:

Absolute space. Space as a set of places. It is an absolute real entity, the container of all things. Its structure is fixed and invariable. Generally, this is considered the space as described by the Euclidean geometry.

Relative space. Space is a system of relations. It is the set of all material things, and relations are abstracted from them. Space is a property of things or things have spatial properties.

The rise of modern science took shape in the works of Nikolaus COPERNICUS (heliocentric system stating that the sun is the center of our planetary system), Johannes KEPLER (mathematical foundation of the heliocentric system), and Galileo GALILEI (foundations of mechanics) in the 16th and 17th centuries.

Isaac NEWTON (1643 – 1727) was a brilliant scientist (dynamic theory) and philosopher. In his philosophy, he was an outspoken proponent of the concept of absolute space, although it strictly contradicts his dynamics theory. The concept of absolute space remained dominant until the late 19th century.

Gottfried Wilhelm LEIBNIZ (1646–1716) on the other hand sustained the concept of relative space. For him, space is a system of relationships between things. It is interesting to note that both NEWTON and LEIBNIZ are the founders of mathematical calculus.

One of the greatest philosophers, Immanuel KANT (1724–1804), claimed that space and time are not empirical physical objects or events. They are merely *a priori* true intuitions, not developed by experience, but used by us to relate and order observations of the real world. Space and time have *empirical reality* (they are absolute and *a priori* given) and *transcendental idealism* (they belong to our conceptions of things but are not part of the things). We cannot know anything about the things as such. In this regard, KANT can be seen as a proponent of absolute space, yet in a far more elaborate and sophisticated way than the previous philosophical approaches.

13.2.3 Contemporary Concepts Of Space And Time

The development of modern physics (field theories, theory of relativity, quantum theory) and mathematics (non-Euclidean geometries) lead to the conclusion that traditional Euclidean geometry (describing the three-dimensional space of our perception) is only an approximation to the real nature of the world.

The field theories (Michael FARADAY and James Clerk MAXWELL) lead to the assumption that space is not empty, but is filled with energy. Therefore, a material existence of space is strongly supported by these theories.

As a consequence of the special and general theory of relativity by Albert EINSTEIN (1879–1955), space and time cannot anymore be considered as two separate entities. We speak of space-time, which is considered a four-dimensional space that can only be described by non-Euclidean geometry. Quantum mechanics states the principle of uncertainty and the discrete character of matter and energy. It has become evident that the space of our perception is not necessarily identical with the microscopic (sub-atomic) space or the space of cosmic dimensions.

13.2.4 Concepts Of Space And Time In Spatial Information Systems

Spatial information is always related to geographic space, i.e., large-scale space. This is the space beyond the human body, space that represents the surrounding geographic world. Within such space, we move around, we navigate in it, and we conceptualize it in different ways. Physical geographic space is the space of topographic, cadastral, and other features of the geographic world. Geographic information system technology is used to manipulate objects in geographic space, and to acquire knowledge from spatial facts.

Geographic space is distinct from small-scale space, or tabletop space. In other words, objects that are smaller than us, objects that can be moved around on a tabletop, belong to small-scale space and are not subject of our interest.

The human understanding of space, influenced by language and cultural background, plays an important role in how we design and use tools for the processing of spatial data.

In the same way as spatial information is always related to geographic space, it relates to the time whose effects we observe in the changing geographic world around us. We are less interested in pure philosophical or physical considerations about time or space-time, but more in the observable spatiotemporal effects that can be described, measured and stored in information systems.

13.3 The Real World And Its Models

As mentioned in the previous sections we always create models of the real world in our minds. When we want to acquire, store, analyze, visualize, and exchange information about the real world, we use other media and means than just interpersonal communication. We need representations of our mind models, i.e., models of the real world that can be used to acquire, store, analyze and transfer information about real world data.

The most common of these models are—in historic sequence—maps and databases. Both have distinct characteristics, advantages and disadvantages. Whereas maps usually have been used to picture real world phenomena, databases can be used to represent real and virtual worlds.

Real worlds are subsets of the reality that we perceive. Virtual worlds are computer generated “realities” that exist only as potentialities with no counterpart in the real world. Yet, we can visualize them, navigate through them and perceive them as “real” (therefore the term virtual reality). There is no difference between real and virtual world models with regard to their representation. The only difference is that the former is a model of something that exists in the real world and the latter is a model of something that exists only in a virtual (physically non-existent) world.

13.3.1 Maps

The best known (conventional) models of the real world are maps. Maps have been used for thousands of years to represent information about the real world. We know maps from ancient Mesopotamia and Egypt, through the Roman times, the Medieval Ages until the present. Today, they are usually drawn on paper or other permanent material and function as data storage and visualization medium. Their conception and design has developed into an art and science with a high degree of scientific sophistication and artistic craftsmanship. Maps have proven to be extremely useful models of reality for many applications in various domains.

Yet, maps are two-dimensional (flat) and static. It is not easy to visualize three-dimensional dynamic features without considerable abstractions in the spatial and temporal domain. We distinguish between topographic and thematic maps. Other cartographic products that are not maps are often used to represent three-dimensional and dynamic phenomena. Such products are, for instance, block diagrams, animations, and panoramic views.

A disadvantage of maps is that they are restricted to two-dimensional static representations, and that they always are displayed in a given scale. The map scale determines the spatial resolution of the graphic feature representation. The smaller the scale, the less detail a map can show. The accuracy of the primary data, on the other hand, puts limits to the scale in which a map sensibly can be drawn. The selection of a proper map scale is one of the first and most important steps in a map conception.

A map is always a graphic representation at a certain level of detail, which is determined by the scale. The process to derive less detailed representations from a detailed one is called map generalization (or cartographic generalization). Map sheets have physical boundaries, and features spanning two map sheets have to be cut into pieces.

Cartography as the science and art of map making functions as an interpreter, translating real world phenomena (primary data) into correct, clear and understandable representations for our use. Maps also become a data source for other maps.

With the advent of computer systems, analog cartography became digital cartography. It is important to note that whenever we speak about cartography today, we implicitly assume digital cartography. The use of computers in map making is an integral part of modern cartography. The role of the map changed accordingly. Increasingly, maps lose their role as data storage. This role is taken over by databases. What remains is the visualization function of maps.

When we look back into the history of digital cartography and geographic information systems, we see that originally maps were considered the main data source for GIS databases. To transfer the contents of a map into a computer database was the major goal. We observe this also in the scientific literature of that time. The terms “map data model” and “map data structure” were widely used. It was not clearly understood that we actually want to store representations of real world phenomena (primary data) and not map data (secondary data). In short, to store map data into a database instead of primary data means to create the model of a model.

13.3.2 Databases

Spatial databases store representations of spatial phenomena in the real world to be used in a geographic information system. They are also called GIS databases or geodatabases. In the design of a database, we distinguish between three different levels of definition. A language that we use to define the database is called a *data model*; each level typically has its own data model. The data model used at the level closest to the end-users is called a *conceptual data model*. In our context, it is used for spatial data modeling. The intention is to define which concepts of interest exist in the application domain for which a database is being designed, and what their relationships are. Such a definition identifies the types of things relevant for a particular application, for example, a cadastral administration, or a landslide hazard analysis system. A commonly used conceptual data model is the *entity-relationship (ER) model*; it uses primitives like entity type to describe independently existing entities, relationship type to define relationships between entities, and attributes to describe characteristic values of entities and relationships. The complete database definition is called the (conceptual) *database schema*. It can be compared to a story written in a language that is the data model. Other, more implementation-oriented, data models will be discussed in Section 13.4.1.

The assumption for the design of a spatial database schema is that spatial phenomena exist in a two- or three-dimensional Euclidean space. All phenomena have various relationships among each other and possess spatial (geometric), thematic and temporal attributes. Phenomena are classified into thematic layers depending on the purpose of the database. This is usually described by a qualification of the database as, for example, cadastral, topographic, land use, or soil database.

The representations of spatial phenomena (i.e., spatial features) are stored in a *scaleless* and *seamless* manner. Scaleless means that all coordinates are world coordinates given in units that are normally used to reference features in the real world (geographic coordinates, meters, feet). From there, calculations can be easily performed and any (useful) scale can be chosen for visualization.

It must be noted, however, that scale plays a role when data are captured from maps as data source. Here, the scale of the source map determines the accuracy of the feature coordinates in the database. Likewise, the accuracy of measurements in field surveys determines the quality of the data. If the coordinates are given in units other than geographic coordinates, information concerning the spatial reference system should also be present.

A seamless database does not show map sheet boundaries or other partitions of the geographic space other than imposed by the spatial features themselves.

It is easy to query a database, and to combine data from different layers (spatial join or overlay). Spatiotemporal databases consider not only the spatial and thematic but also the temporal extent of the features they represent. Various spatial, temporal and spatiotemporal data models have been developed.

13.3.3 Space And Time In Real World Models

In modern physics, it is common to speak of space-time to express the close connection that exists between space and time according to the special and general theory of relativity. Here, we do not consider the physical characteristics of space and time, but the (simplified) ways of representing spatial phenomena in a GIS database.

In general, modeling can be described as creating a structure preserving mapping (morphism) from a domain to a co-domain. In our case, the domain is the real world, and the co-domain is a real world model. Such a mapping normally creates a ‘smaller’ (i.e., abstracted, generalized) image of the original. Structure preserving means that the elements of the co-domain (spatial features) behave in the same (however simplified or abstracted) way as the elements of the domain (spatial phenomena). Figure 81 illustrates the principle of spatial modeling.

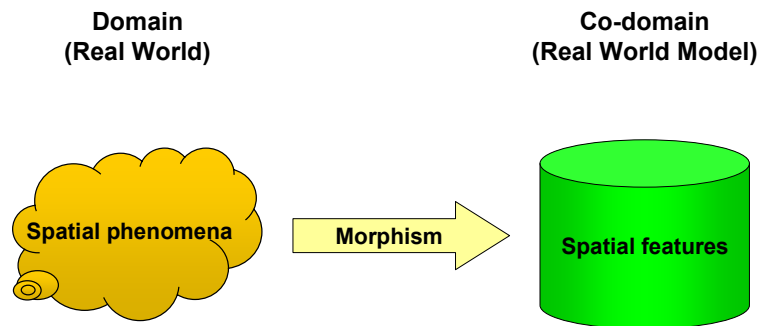


Figure 81. Spatial modeling is a structure preserving mapping from the real world to a spatial model.

As mentioned above, we consider space to be the three-dimensional Euclidean space of our common sense. All phenomena exist in this space and undergo changes, which we perceive as the passing of time. In this sense, time is modeled indirectly as changes in (spatial or thematic) attributes of the features.

We call a spatial data model that also considers time a spatiotemporal data model. Sometimes such a model is addressed as four-dimensional, giving the impression that time is the fourth dimension in addition to the three spatial dimensions. It is, however, better to call it spatiotemporal instead of four-dimensional. First, time is not of the same type as the spatial dimensions; it has a distinct different quality. Secondly, the term ‘four-dimensional’ only makes sense when we always would consider three spatial dimensions. In most of the cases, however, the spatial data model only considers two dimensions.

13.4 Real World Models And Their Representation

Spatial data are computer representations of spatial features. A modeling language for a GIS database is a spatial data model. It is used in the design of spatial databases. A spatial database holds a digital representation of the real world, sometimes called a digital landscape model (DLM).

A DLM is the basis for spatial analysis and manipulation of spatial data. The feature space for a spatial database is a geometric space in which we model features at various levels of detail. A good data model should allow for multiple representations of spatial features. The transformation of a representation from a detailed to a less detailed version is called model generalization. This is different from cartographic generalization where a graphic representation (digital or analog) at a smaller scale is derived from a large-scale data set or map under certain merely graphic constraints.

Database creation is a two-step process: first, the database is designed by defining a database schema that identifies types but no occurrences. Secondly, the database is filled with real data, thereby creating a digital landscape model. From the database (DLM), we derive graphic representations in digital or analogue form (cartographic model) that can result in a cartographic product. Figure 82 shows the different aspects of data modeling for database and cartographic design.

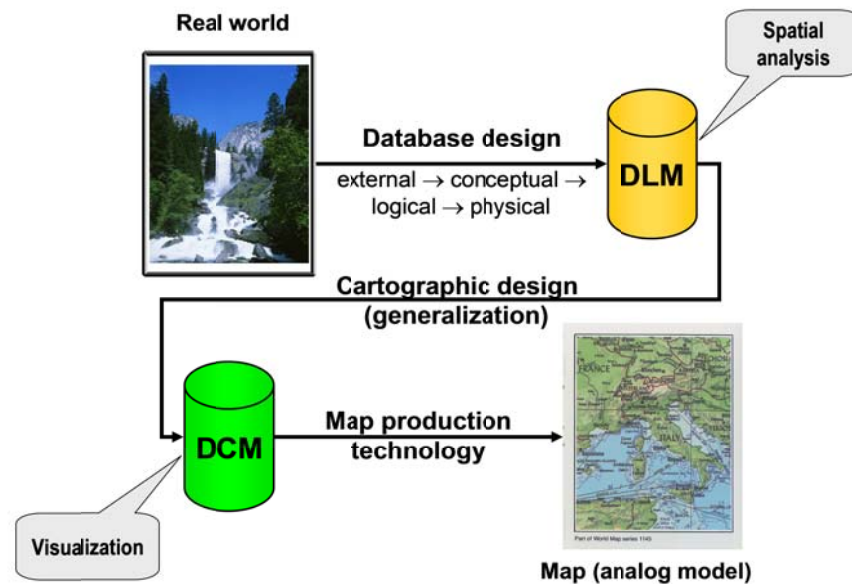


Figure 82. Data modeling from the real world to a database (digital landscape model), and from there to digital cartographic models and analogue products for visualization

We are particularly interested in properties of the geometric space that remain invariant under certain transformations. This is important to guarantee a consistent representation of the features in a database. These properties are related to topology.

Some features possess a degree of uncertainty in either their thematic or spatial extent. For example, soil types do not have crisp boundaries. Linguistic expressions of vagueness or uncertainty such as “moderate slope” or “close to Vienna” are often a better analysis approach than clear cut class boundaries. We have to devise proper means to take care of these uncertainties in our spatial data models. Fuzzy logic provides the tools.

Spatial data exist not only in space but also in time. They carry temporal characteristics that data models must be able to handle. Several spatiotemporal data models have been proposed.

13.4.1 Database Design

The design of a database consists of several phases. The result of every phase is a schema of the respective level. A schema is a representation of the database as described with a particular data model. This approach is called the ANSI/SPARC architecture; it comprises four levels.

A database is generally thought to serve multiple users or user groups. They may have different perceptions of the data stored. At this level, each user (group) is supported with its own external view of the database. We may define an external view as a personalized conceptual database schema. There will be as many external views as there are users and user groups. Mainly domain experts do spatial modeling at this level.

All external views are merged into a single conceptual schema of the database. This is usually done with high-level semantic data models such as the entity-relationship model (ER model), the extended ER model (EER), or object-oriented data models. The basic constructs of the ER model are entity types (e.g., country), attribute types (e.g., population) and relationship types (e.g., neighbor of). Instances of the types populate the database, e.g., ‘Austria’, ‘8 million’, and ‘Germany, Switzerland’ are instances of the entity type ‘country’, the attribute type ‘population’, and the relationship type ‘neighbor of’, respectively. A conceptual schema is implementation-independent and not related to any particular database management software. It provides an answer to the question what phenomena are represented in the database.

The conceptual schema is translated into a logical schema using one of the logical data models. Currently, the most popular one is the relational data model, which is based on relational algebra. Most commercial database implementations provide support for this model. It is easy to understand because it is based on relations—sets of records—that have a straightforward implementation as tables. The logical schema is meant to provide the definition of a redundancy-free data set.

A physical schema is the result of the implementation of the logical schema with particular database management software. Table 22 summarizes the ANSI/SPARC architecture.

Table 22. Data models and schemas in database design (the ANSI/SPARC architecture)

Schema	Models used to derive the schema
External views	Depending on different user perspectives, a subset of the real world is defined and described (spatial modeling).
Conceptual schema	A synthesis of external views to create a conceptual schema making use of semantic data modeling techniques such as the entity-relationship approach.
Logical schema	Transformation of the conceptual schema into a logical schema using the relational model. Emphasis is on redundancy removal.
Physical schema	This is the mapping of the logical schema into data structures and algorithms. It is normally not accessible or visible to the user. Its emphasis is on processing speed.

13.4.2 Spatial Data Models

Among spatial data models, we can distinguish two major types, field- and object-based models. *Field-based models* consider spatial phenomena to be of a continuous nature where in every point in space a value of the field can be determined. Examples of such phenomena are temperature, barometric pressure, or elevation. *Object-based models* consider space to be populated by well distinguishable, discrete, bounded objects with the space between objects potentially being empty. Examples are a cadastre with clearly identifiable objects like parcels and buildings.

Field versus object can be viewed as a manifestation of the philosophical conception of *plenum* versus *atomic space* (see Section 13.2.1), or as in modern physics, *wave* versus *particle* (see Section 13.2.3). In GIS, fields are usually implemented with a tessellation approach, objects with a (topological) vector approach. The following sections briefly illustrate the two different model types.

13.4.2.1 Field-based Models

The underlying space for a field-based model is usually taken as the two- or three-dimensional Euclidean space. A field is a computable function from a geometrically bounded set of positions (in 2D or 3D) to some attribute domain. Computable means that for every position within the geometric bounds a value can be determined by either measurement or by computation. A field-based model consists of a finite collection of such fields (Figure 83).

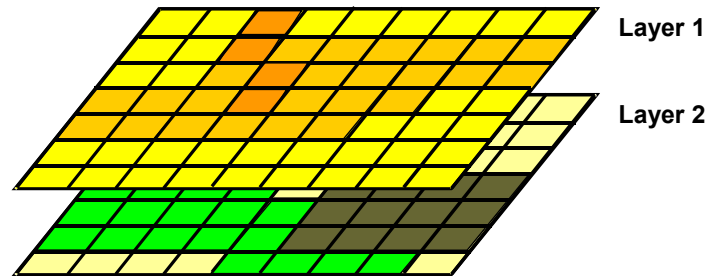


Figure 83. Two data layers in a field-based model

Fields can be discrete, continuous and differentiable. Discrete fields represent features with boundaries; continuous fields are used for features where the underlying function is continuous, such as for temperature, barometric pressure or elevation. If the field function is differentiable, we can even compute the slope at every position.

Though geometrically bounded, the domain of a field is still an infinite set of positions. Computers cannot easily represent field values for all these positions, so we must accept an approximation. The standard way to obtain this is to finitely represent the geometrically bounded space through a subdivision into a regular or irregular tessellation, consisting either of square (cubic) or triangular (tetrahedral) parts. These individual parts are called *locations*; points often approximate them.

Our finite approximation of positions into locations leads to some forms of interpolation. The field value at a location can be interpreted as one for the whole tessellation cell, in which case the field is discrete, not continuous or even differentiable. Some convention is needed to state which value prevails on cell boundaries; with square cells, this convention often says that lower and left boundaries belong to the cell.

Another option is to interpret the field value at a location as representative only for some position within the cell. Again that position is fixed by convention, and may be the cell centroid or, for instance, its left lower corner. Field values for positions other than these must be computed through some form of interpolation function, which will use one or more nearby field values to compute the value at the requested position. This allows representing continuous, even differentiable, functions.

To represent spatial features using a field-based approach we have to perform the following steps:

1. Define or use a suitable model for the underlying space (tessellation).
2. Find suitable domains for the attributes.
3. Sample the phenomena at the locations of the tessellation to construct the spatial field functions.
4. Perform analysis, i.e., compute with the spatial field functions.

13.4.2.2 Object-based Models

Object-based models decompose the underlying space into identifiable, describable objects that have spatial, thematic, and temporal attributes, as well as relationships among each other (Figure 84). The space outside the objects is empty. Examples of objects are buildings, cities, towns, districts or countries; attributes are, for instance, the number of floors, population, boundary or area.

In a GIS database implementation, objects are represented as a structured collection of geometric primitives (points, lines, polygons, and volumes) under geometric, thematic and topological constraints.

Object-based models are discrete models. Operations in the model always refer to the manipulation of individual objects or sets of objects. Manipulations concern the spatial, thematic, topological or temporal domain. Accordingly, they are realized through geometric, attribute manipulation or topological operations.

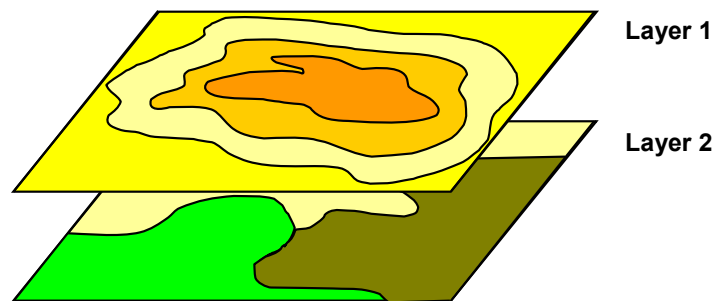


Figure 84. Layers in an object based model

Topology plays a major role in object-based models. It is the ‘language’ that allows us to specify and enforce consistency constraints for spatial databases. The majority of object-based models are two-dimensional. Recently, three-dimensional data models have been proposed. Their topology is more difficult to handle than in the standard two-dimensional cases.

13.4.3 Spatiotemporal Data Models

Beside geometric, thematic and topological properties, spatial data possess also temporal characteristics. It is, for instance, interesting to know who were the owners of a land parcel in 1980, or how did land use of a given piece of land change over the last 20 years.

Spatiotemporal data models are data models that can also handle temporal information in spatial data. Several models have been proposed. The most important ones will be discussed briefly. Before we describe the major characteristics of the spatiotemporal data models, we need a framework to describe the nature of time itself. Time can be characterized according to the following properties:

Time density. Time can be discrete or continuous. Discrete time is composed of discrete elements (seconds, minutes, hours, days, months, or years). In continuous time, for two points in time, there is always another point in between. We can also structure time by events (points in time) or states (time intervals). When we represent states by intervals bounded by nodes (events) we can derive temporal relationships between events and states such as “before”, “overlap”, “after”, etc.

Dimensionality of time. Valid time (or world time) is the time when an event really happened. Transaction time (or database time) is the time when the event was recorded in the database.

Time order. Time can be linear, extending from the past to the present, and into the future. We know also branching time (possible scenarios from a certain point in time onwards) and cyclic time (repeating cycles such as seasons or days of a week).

Measures of time. A chronon is the shortest non-decomposable unit of time that is supported by a database (e.g., a millisecond). The life span of an object is measured by a (finite) number of chronons. Granularity is the precision of a time value in a database (e.g., year, month, day, second, etc.). Different applications require different granularity. In cadastral applications, time granularity can be a day; in geological mapping, time granularity is more likely in the order of thousands of years.

Time reference. Time can be represented as absolute (fixed time) or relative (implied time). Absolute time marks points on the timeline where events happen (e.g., “6 July 1999 at 11:15 p.m.”). Relative time is indicated relative to other points in time (e.g., “yesterday”, “last year”, “tomorrow,” which are all relative to “now”, or “two weeks later,” which may be relative to an arbitrary point in time.).

In spatiotemporal data models, we consider changes of spatial and thematic attributes over time. In data analysis, we can keep the spatial domain fixed and look only at the attribute changes over time for a given location in space. We would, for instance, be interested how land cover changed for a given location or how the land use changed for a given land parcel over time, provided its boundary did not change.

On the other hand, we can keep the attribute domain fixed and consider the spatial changes over time for a given thematic attribute. In this case, we could be interested to see which locations were covered by forest over a given period.

Finally, we can assume both the spatial and attribute domain variable and consider how an object changed over time. This actually leads to notions of object motion, and these are a subject of current research, with two of the applications being traffic control and mobile telephony. But many more applications are on the horizon: think of wildlife tracking, disease control, and weather forecasting. Here, the problem of object identity becomes apparent. When does a change or movement cause an object to disappear and become a new one?

In the following, we describe the major characteristics of some popular spatiotemporal models.

13.4.3.1 Space-Time Cube Model

This model is based on a two-dimensional space (spanned by the x- and y-axis) whose features are traced through time (along the z-axis) thereby creating a space-time cube. The traces of objects through time create a worm-like trajectory in the space-time cube. This model potentially allows absolute, continuous, linear, branching and cyclic time. It supports only valid time. The attribute domain is kept fixed and the spatial domain variable.

13.4.3.2 Snapshot Model

In the snapshot model, layers of the same theme are time-stamped. For every point in time that we would like to consider, we have to store a layer and assign the time to it as an attribute. We do not have any information about the events that caused different states between layers. This model is based on a linear, absolute, discrete time. It supports only valid time and multiple granularity. The spatial domain is fixed (field-based) and the attribute domain is variable.

13.4.3.3 Space-Time Composite Model

The space-time composite model starts with a two-dimensional situation (plane or layer) at a given start time. Every change of features that happens later is projected onto the initial plane and intersected with the existing features, thereby creating an incrementally

built polygon mesh. Every polygon in this mesh has its attribute history stored with it. The space-time composite model is based on linear, discrete, relative time. It supports both valid and transaction time, and multiple granularity. It keeps the attribute domain fixed and the spatial domain variable.

13.4.3.4 Event-based Model

In an event-based model, we start with an initial state and record events (changes) along the time line. Whenever a change occurs, an entry is recorded. This is a time-based model. The spatial and thematic attribute domains are secondary. The model is based on discrete, linear, relative time, supports only valid time and multiple granularity.

13.4.3.5 Spatiotemporal Object Model

This model is based on spatio-temporal objects (ST-objects) that are a complex of ST-atoms (STA). Both objects and atoms have a spatial and a temporal extent. The model is based on discrete, absolute, linear time, and supports valid and transaction time as well as multiple granularity.

13.5 Summary

Geographical information systems process spatial information. The information is derived from spatial data in a database. To sensibly work with these systems, we need models of spatial information as a framework for database design. These models address the spatial, thematic and temporal dimensions of real world phenomena. An understanding of the principle concepts of space and time rooted in philosophy, physics and mathematics is a necessary prerequisite to develop and use spatial data models.


We know two major approaches to spatial data modeling, the analogue map approach, and spatial databases. Today, the function of maps as data storage (map as a database) is increasingly taken over by spatial databases. In databases, we store representations of phenomena in the real world. These representations are abstractions according to selected spatial data models. We know two fundamental approaches to spatial data modeling, field-based and object-based models. Both have their merits, advantages and disadvantages for particular applications.

Consistency is an important requirement for every model. Topology provides us with the mathematical tools to define and enforce consistency constraints for spatial databases, and to derive a formal framework for spatial relationships among spatial objects.

Spatial data not only possess spatial and thematic attributes, but extend also into the temporal domain. A model of time for spatial information is an important ingredient for any spatial data model, thus leading to what is called spatiotemporal data models.

Solutions of Exercises

This section contains solutions of the exercises mentioned in the text. The reader is advised to first try to solve the problems him/herself before consulting this section. For many of the problems a detailed solution is given; for some only the results are mentioned.



Chapter 2

Chapter 3

Chapter 4

Chapter 5

Exercise 16 $\overline{B} \cup C = \{b, d, e, f, g\}$, $\overline{C} \cap A = \{a, c, d\} = \overline{C}$, $B - C = \{a, c\}$, $\wp(B - C) = \{\{\}, \{a\}, \{c\}, \{a, c\}\}$

Chapter 6

Exercise 19 f is not symmetric because $\langle 1, 4 \rangle$ is in f but not $\langle 4, 1 \rangle$. f is not transitive because $\langle 2, 3 \rangle$ and $\langle 3, 2 \rangle$ in f but not $\langle 2, 2 \rangle$. g is not symmetric because $\langle 3, 1 \rangle$ in g but not $\langle 1, 3 \rangle$. g is transitive. h is not symmetric because $\langle 2, 1 \rangle$ in h but not $\langle 1, 2 \rangle$. h is not transitive because $\langle 2, 1 \rangle$ and $\langle 1, 4 \rangle$ in h but not $\langle 2, 4 \rangle$.

Exercise 20 f is not a function because 2 has more than one value! g is not a function because not all elements of the domain appear in the relation! h is a function! The fact that $\langle 2, 1 \rangle$ appears twice does not change the set of pairs.

Exercise 21 $R_1 R_2 = \{\langle a, d \rangle\}$, $R_2 R_1 = \{\langle b, d \rangle\}$, $R_1^2 = \{\langle a, a \rangle, \langle a, c \rangle, \langle a, d \rangle\}$,
 $R_2^3 = \{\langle b, b \rangle, \langle b, d \rangle, \langle b, c \rangle\}$

Exercise 22
$$\frac{f(x+h) - f(x)}{h} = \frac{(x+h)^2 - 3(x+h) + 2 - (x^2 - 3x + 2)}{h} =$$

$$\frac{x^2 + 2xh + h^2 - 3x - 3h + 2 - x^2 + 3x - 2}{h} = \frac{2xh + h^2 - 3h}{h} = 2x + h - 3$$

Chapter 7

Exercise 24

Exercise 26 (a) (11, 2, -5), (b) (1, 1, -9), (c) (7, 14, 21)

Exercise 27

Exercise 28 7

Chapter 8

Exercise 31 Both algebras are commutative groups. This can easily be verified by the usual laws for addition and negative numbers. The function $f(x) = 2x$ is surjective (all even numbers appear as values) and injective (distinct arguments produce distinct values), therefore bijective. Furthermore, we

have that $f(a+b) = 2(a+b) = 2a+2b = f(a) + f(b)$. $f(-a) = 2(-a) = -2a = -f(a)$. $f(0) = 2 \cdot 0 = 0$. Therefore the function is an isomorphism!

Exercise 32 This can be easily verified by substituting T and F into the axioms for a Boolean algebra and applying the rules for logical operators.

$$T \vee F = F \vee T$$

$$T \wedge F = F \wedge T$$

$$(T \vee F) \vee T = T \vee (F \vee T)$$

$$(T \wedge F) \wedge T = T \wedge (F \wedge T)$$

$$T \wedge (F \vee T) = (T \wedge F) \vee (T \wedge T)$$

$$T \vee (F \wedge T) = (T \vee F) \wedge (T \vee T)$$

$$T \vee F = T \quad F \vee F = F$$

$$T \wedge T = T \quad F \wedge T = F$$

$$T \vee \neg T = T \vee F = T \quad F \vee \neg F = F \vee T = T$$

$$T \wedge \neg T = T \wedge F = F \quad F \wedge \neg F = F \wedge T = F$$

Exercise 33 The function is surjective and maps every element of I to B. Because the function is surjective, it cannot be an isomorphism! To prove that the function maps the binary operation properly, we distinguish between four cases for $a+b$, with $a, b \in I$

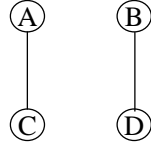
a	b	$f(a+b)$ with + as usual	f(a)	f(b)	$f(a)+f(b)$ with + as defined in the operation table
even	even	even + even = even, therefore 0	0	0	0
even	odd	even + odd = odd, therefore 1	0	1	1
odd	even	odd + even = odd, therefore 1	1	0	1
odd	odd	odd + odd = even, therefore 0	1	1	0

To prove that the unary operation maps correctly, we show that $f(-\text{even}) = -f(\text{even}) = -0 = 0$ and $f(-\text{odd}) = -f(\text{odd}) = -(1) = 1$. The constant maps as $f(0) = 0$, because 0 is even.

Chapter 9

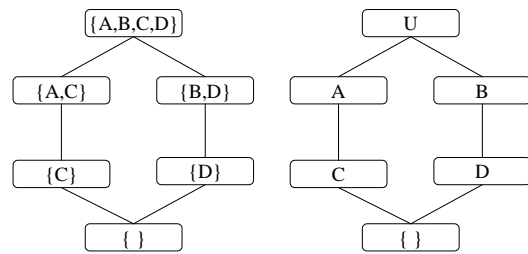
Chapter 10

Exercise 35 The order diagram of the poset looks like



The normal completion lattice is built as follows: $(\{A\})^* = \{A, C\}$, $(\{B\})^* = \{B, D\}$, $(\{C\})^* = \{C\}$, $(\{D\})^* = \{D\}$, $(\{A, B\})^* = \{A, B, C, D\}$, $(\{A, C\})^* = \{A, C\}$, $(\{A, D\})^* = \{A, B, C, D\}$, $(\{B, C\})^* = \{A, B, C, D\}$, $(\{B, D\})^* = \{B, D\}$, $(\{C, D\})^* = \{A, B, C, D\}$, $(\{A, B, C\})^* = \{A, B, C, D\}$, $(\{A, C, D\})^* = \{A, B, C, D\}$, $(\{A, B, D\})^* = \{A, B, C, D\}$, $(\{B, C, D\})^* = \{A, B, C, D\}$, $(\{A, B, C, D\})^* = \{A, B, C, D\}$, $(\{\})^* = \{\}$

This gives the following poset ordered by set inclusion and after the mapping of corresponding elements and renaming the universe:



Chapter 11

Chapter 12

References and Bibliography

- Aliev R.A., Aliev R.R., 2001, *Soft Computing and Its Applications*. World Scientific Publishing Co. Pte. Ltd., Singapore, London.
- Armstrong, M.A., 1983, *Basic Topology*. Springer Verlag, New York.
- Bonham-Carter G.F., 1994, *Geographic Information Systems for Geoscientists: Modelling with GIS*. Pergamon, Elsevier Science, Kidlington, U.K.
- Burrough P.A., McDonnell R.A., 1998, *Principles of Geographical Information Systems*. Oxford University Press.
- Burrough P.A., Frank A.U. (Eds.), 1996, *Geographic Objects with Indeterminate Boundaries. GISDATA II*, Taylor & Francis, London.
- Corbett, J.P. 1979. Topological Principles in Cartography. Technical paper – Bureau of the Census; 48.
- Davey, B.A., Priestley, H.A. 1990. *Introduction to Lattices and Order*. Cambridge: Cambridge University Press.
- Demicco R.V., Klir G.J., 2004, *Fuzzy Logic in Geology*. Elsevier Science (USA).
- Dubois D., Prade H., 1980, *Fuzzy Sets and Systems*. Academic Press, San Diego.
- Dubois D., Prade H., 2000, *Fundamentals of Fuzzy Sets*. Kluwer Academic Publishers, Boston, London, Dordrecht.
- Edgar W.J. 1989. *The Elements of Logic*. Macmillan Publishing Company, New York.
- Grimaldi R.P. 1989. *Discrete and Combinatorial Mathematics. An Applied Introduction*. Second Edition. Addison-Wesley Publishing Company.
- Hootsmans R., 1996, Fuzzy Sets and Series Analysis for Visual Decision Support in Spatial Data Exploration. PhD thesis, University Utrecht, The Netherlands. ISBN 90 6266 134 3.
- Jiang B., 1996, Fuzzy Overlay Analysis and Visualization in Geographic Information Systems. PhD Thesis, University Utrecht and ITC, The Netherlands. ISBN 90 6266 128 9.
- Kainz W., Egenhofer M., Greasley I. 1993, Modeling Spatial Relations and Operations with Partially Ordered Sets. *International Journal of Geographical Information Systems*, Vol. 7, No. 3, 215–229.
- Kainz W. 1995, Logical Consistency. In: S.C. Gupta and J.L. Morrison (eds.), *Elements of Spatial Data Quality*, Elsevier Science.
- Leung Y., 1997, *Intelligent Spatial Decision Support Systems*. Springer-Verlag, Berlin, Heidelberg.
- Petry F.E., Robinson V.B., Cobb M.A. (Eds.), 2005, *Fuzzy Modeling with Spatial Information for Geographic Problems*. Springer-Verlag, Berlin, Heidelberg

- Stanat D.F., McAllister D.F. 1977. *Discrete Mathematics in Computer Science*. Prentice Hall, Inc., Englewood Cliffs, N.J.
- Tanaka K., 1997, *An Introduction to Fuzzy Logic for Practical Applications*. Springer Verlag, New York.
- Tang X., 2004, Spatial Object Modelling in Fuzzy Topological Spaces with Applications to Land Cover Change. PhD thesis, University of Twente, Enschede, The Netherlands, ITC dissertation number 108.
- Zadeh L., 1965, Fuzzy sets. *Information and Control* 8; 338-353
- Zheng D., 2001, A Neuro-Fuzzy Approach to Linguistic Knowledge Acquisition and Assessment in Spatial Decision Making. PhD thesis, University Vechta and ITC, Enschede.
- Zimmermann H.J., 1987, *Fuzzy Sets, Decision Making, and Expert Systems*. Kluwer Academic Publishers, Dordrecht, The Netherlands.
- Zimmermann H.J., 2001, *Fuzzy Set Theory and Its Applications*, Fourth Edition. Kluwer Academic Publishers, Boston, Dordrecht, London.

Index

- absurdity, 9
- antisymmetrix, 34
- Arabs, 2
- argument
 - logical, 20
- assertion, 6
- Babylonians, 2
- bijection. *See* bijective
- bijective, 37
- CANTOR, 26
- cardinality, 26
- Cartesian product, 32
- Chinese, 2
- codomain, 33, 36
- complement, 27
- composite function, 37
- composite relation, 35
- conclusion, 20
- conjunction, 7
- constructive dilemma, 21
- contingency, 9
- contradiction, 9
- contrapositive, 8
- converse, 8
- cross product, 32, *See* Cartesian product
- destructive dilemma, 21
- difference, 27
- digraph, 33
- directed graph. *See* digraph
- disjunction, 7
- domain, 33, 36
- Egyptians, 2
- equivalence class, 35
- equivalence relation, 35
- EUCLID, 2
- exclusive or, 7
- existential*, 15
- existential generalization, 22
- existential instantiation, 22
- existential quantifier. *See* quantifier
- function, 2, 36
- hypotheses, 20
- implication, 8
- inclusive or, 7, *See* disjunction
- Indians, 2
- injection. *See* injective
- injective, 37
- intersection, 27
- inverse function, 40
- irreflexive, 34
- logic, 2
- logical and, 7, *See* conjunction
- map, 36, *See* function
- mapping*, 36, *See* function
- modus ponens*, 21
- modus tollens*, 21
- negation, 7
- one-to-one. *See* injective
- onto. *See* surjective
- operators, 7
- power set, 29
- predicate, 14
- premises, 20
- proposition, 6
- propositional forms, 6
- propositional variables, 6
- quantifier
 - existential, 15
 - universal, 15
- quotient set, 35
- reflexive, 34
- relation, 2, 33
- satisfiable, 15, 16
- set, 26
- set theory, 2
- structures, 2
- subset, 27
 - proper, 27
- Sumerians, 2
- superset, 27
- surjection. *See* surjective
- surjective, 37
- syllogism, 21
 - disjunctive, 21

hypothetical, 21
symmetric, 34
tautology, 9
transformation, 36, *See* function
transitive, 34
truth table, 7
union, 27
universal generalization, 22

universal instantiation, 22
universal quantifier. *See* quantifier
universe. *See* universe of discourse
universe of discourse, 14
unsatisfiable, 15, 16
valid, 16