An Introduction to the Application of Formal Theories to GIS

Thomas Bittner and Andrew U. Frank
Department of Geoinformation
Technical University of Vienna
Gusshausstrasse 27-29
A - 1040 Wien
{bittner, frank}@geoinfo.tuwien.ac.at

Abstract

This paper discusses the application of formal methods to the representation and modeling of geographic space. We consider Geographic Information Systems implementations of formal theories of geographic space. This paper explores preconditions for the application of formal methods to geographic space. We model GIS as formal systems of logic. This allows to apply methods of deductive science to geographic space and to abstract from implementation. The paper explains the basic notions of formal systems and formal theories. We use a running example from the context of GIS.

1 Introduction

In this paper we consider the formalization of geographic space (Egenhofer and Mark 1995) in Geographic Information Systems (GIS) (Laurini and Thompson 1994). Formalization is the representation of models of the world by means of a formal language. Formal languages, for example, are C (Kernighan and Plauger 1978), PROLOG (Kowalski 1979), HASKELL (Peterson, Hammond et al. 1996), … In order to abstract from particular formal languages and from particular implementation issues we consider GIS as formal systems (Curry, Feys et al. 1958), in particular as formal systems of logic (Tarski 1941), (Carnap 1958), (Barwise 1977). Formal systems of logic have two aspects that are important in this context:

1. Formal systems of logic are applications of the paradigm of the deductive science (Tarski 1941). The application of the paradigm of the deductive science "serves the purpose of securing for the knowledge acquired … the highest possible degree of clarity and certainty" (Tarski 1941, pg. 117). Within this paradigm we can apply methods and results from logic and mathematics.
2. Formal systems have a notion of proof.
   The notion of proof provides means to consider computation at the highest level of abstraction. It allows to abstract from particular programming languages and from implementing particular of the algorithms.

In this paper we use formal systems of logic to consider the following questions in an abstract manner:

- How to use formal systems for the formalization of models of geographic space, and
- How to represent computation in GIS by the notion of proof.

The paper is organized as follows: First we discuss the preconditions for the application of formal methods to geographic space (Section 2). Then we discuss basic notions of formal systems (Section 3) and formal theories (Section 4). In Section 5 we show that GIS can be considered as implementations of formal theories of (the geometry of) geographic space. In the end we draw the conclusion that GIS can

be abstractly considered and modeled as formal systems. We use a running example to introduce basic notions of formal systems and formal theories in the context of problems from GIS.

The Example

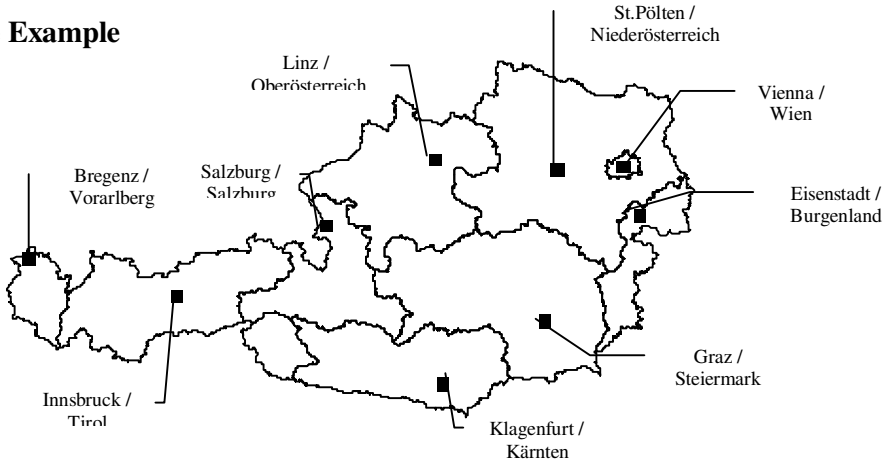

Figure 1: Political subdivision of Austria with provinces and their capitals

We use a representation of a political subdivision of Austria with a few cities as a running example. We use this running example from the field of GIS to bring the abstract notions of formal systems into the context of this paper. We point out that the example is very simplistic. Its purpose is to explain different aspects as simple as possible. It is not the purpose of the example to demonstrate the power of the formal method. In Figure 1 a map of the political subdivision of Austria is shown. Austria is divided into nine provinces. Each province has a capital. In Figure 1 first the name of the capital and then the name of the corresponding province is given. Vienna is the capital of Austria as well as a province. We use the term 'Wien' in order to refer to the province and 'Vienna' in order to refer to the capital. We do not explicitly distinguish between Vienna as the capital of Austria and Vienna as the capital of the province 'Wien'.

2 Preconditions for the Application of Formal Methods to Geographic Space

2.1 Conceptual Models, Formal Models, and Language Representations

Formal modeling and representing the world is based on abstract representations of phenomena in the world. The process of abstraction is critical since during this process the relevant aspects are kept and non relevant aspects are discarded. The process of abstraction results in a representation of the phenomena that has a finite structure, a model. A model consists of a *finite* number of *sets of individuals*, and a *finite* number of relations and operations that are defined on the individuals. Individuals in the model correspond to individuals or classes of individuals in the world. Relations

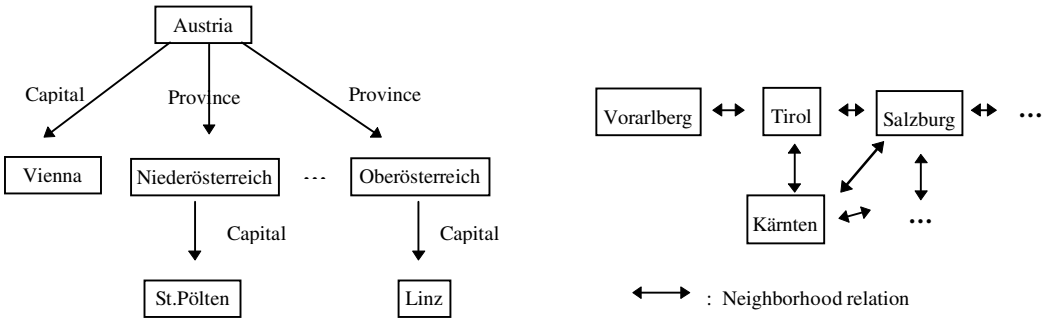

Figure 2: Conceptual model of the political subdivision of Austria in provinces and the neighborhood structure of the provinces

and operations of the model correspond to relations, properties, and processes in the world. This correspondence is established informally in the mind of the designer of a theory. These kinds of models are called *conceptual models*. The important point is that the structure of a conceptual model is finite, i.e., it has only a finite number of components. In Figure 2 we give an example of a conceptual model of the political subdivision of Austria into provinces with capitals and the neighborhood structure of the provinces.

On the formal level a model is considered to be an algebraic structure consisting of sets of abstract individuals, relations, and operations. The sets of individuals of the formal model may be infinite and continuous. An example for an algebraic structure with an infinite and continuous domain of individuals is the algebra of the real numbers. Formal models are used to represent conceptual models.

In Example 1 we consider a formal model of a finite domain of individuals corresponding to the conceptual model in Figure 2:

The Universe of Individuals:
{ Austria, Niederösterreich, Oberösterreich, Salzburg, Tirol, Vorarlberg, Kärnten, Steiermark, Burgenland, Vienna, Wien, St.Pölten, Linz, Salzburg_city, Salzburg, Innsbruck, Bregenz, Klagenfurt, Graz, Eisenstadt }
Unary Relations (Predicates):
Country = { Austria }
Province = { Niederösterreich, Oberösterreich, Salzburg, Tirol, Vorarlberg, Kärnten, Steiermark, Burgenland, Wien }
City = { Vienna, St.Pölten, Linz, Salzburg_city, Innsbruck, Bregenz, Klagenfurt, Graz, Eisenstadt }
Binary Relations:
Capital = { <Vienna, Austria>, <Vienna, Wien>, <Eisenstadt, Burgenland>, <Graz, Steiermark>, <Klagenfurt, Kärnten>, <Innsbruck, Tirol>, <Bregenz, Vorarlberg>, <Salzburg_city, Salzburg>, <Linz, Oberösterreich>, <St.Pölten, Niederösterreich> }
Neighbor = < Niederösterreich, Oberösterreich>, <Niederösterreich, Burgenland>, …
,<Tirol,Vorarlberg>,<Oberösterreich,Niederösterreich>, <Burgenland,Niederösterreich>, … , <Vorarlberg, Tirol> }
Part_Of = { <Niederösterreich, Austria>, <Oberösterreich, Austria>, … , <Burgenland, Austria>, <Vienna, Austria>, <Eisenstadt, Austria >, <Graz, Austria >, …, <St.Pölten, Austria >, <Vienna, Wien>, <Eisenstadt, Burgenland>, …, <St.Pölten, Niederösterreich>}

Example 1: A Formal Model of a Finite Domain of Individuals

Formal models can be represented by formal language. A formal language is a set of well-defined strings of symbols. Representations of formal models in terms of a formal language on a computer are finite and discrete. Discreteness is due to the symbolic character of the representation. Finiteness of the representation is precondition for computability in finite time with finite resources. Formalization is the process of the design of representations of potentially infinite and possibly continuous phenomena by a finite set of discrete symbols.

Formal theories are representations of formal models in terms of formal languages. Formal theories are possibly infinite sets of expressions of a formal language. Axiomatic theories are used to finitely represent formal theories (Section 4).

2.2 Ontology

The process of abstraction and discretization is not arbitrary. The definition of classes of individuals, relations and operations needs to be done in a way that preserves the properties of the domain to be represented and modeled. In particular *abstraction and discretization have to preserve the "… a priory distinctions among the entities of the world …*" (Guarino 1994, pg. 2).

In Example 1 we consider ontological different kinds of things: Austria, provinces of Austria, cities which are capitals of provinces or states, or both, administrative relations, spatial relations.
If we formalize the administrative structure of Austria we consider Austria as a state and provinces as administrative units. Between a

state and its administrative units administrative relations hold. In administrative relations many spatial aspects of the phenomena are irrelevant. Consequently we can abstract from them. A relevant spatial aspect which needs to be preserved is that the provinces partition Austria: Every place belongs to a province, no place is in two provinces.
From the point of view of administration capitals are in their administrative unit even if there location may be at the periphery (East-Berlin as the capital of East-Germany was always in East-Germany even if the region occupied by East-Berlin was topologically a tangential proper part of the region occupied by East-Germany).
If we formalize the spatial structure of the political subdivision of Austria provinces are not only considered as administrative units. The partition property might not be sufficient to capture relevant aspects. There might be cases in which the neighborhood structure is considered. Often the full geometry of the partition, i.e., the exact location of the boundaries is required to be represented.
Considering the spatial structure the city of 'Vienna' and the province 'Wien' occupy the same region. The terms 'Vienna' and 'Wien' refer to ontologically different phenomena.

Example 2: Ontological distinctions

2.3 Paradigms of Formalization

Finite and discrete representations of infinite and continuous domains are achieved by means of abstraction and discretization. One way of representing infinite and continuous domains of individuals is to represent classes of individuals rather than individuals. Individuals in the same class are considered to be equivalent. No distinctions between them are made. Equivalence classes of individuals partition the domain of individuals. Each individual belongs to one and only one class of equivalence. The difficult problem is *to define the classes of equivalence in such a way that the structural properties of the original domain of individuals are preserved.* This can be achieved if structures that *govern* the domain of individuals are used to *define* the classes of equivalent individuals.

There are classes of structures that can be used to define classes of equivalent individuals. The paradigm of formalization refers to the use of a particular class of structures. Such structures are for example:

- Properties and relations that remain invariant, i.e. unchanged, under particular classes of transformations, e.g., (Randell, Cui et al. 1992), (Egenhofer and Herring 1990).
- Frames of references and relations of order of individuals with respect to the frame, e.g. (Frank 1992), (Hernandez 1994);
- Landmark values (individuals which represent significant changes) (Kuipers 1994) and order relations between them which qualitatively structure the domain;

Paradigms of formalization which make use of those structures for example are:

- The *geometric* paradigm of formalization. It is based on definition of equivalence with respect to properties and relations that remain invariant under transformations (Klein 1872).
- The *analytic* paradigm of formalization. It is based on definition of equivalence based on relations of order with respect to a frame of reference.
- The *qualitative* paradigm of formalization. It is based on definition of equivalence with respect to landmark values and order relations between them (Kuipers 1994), (Schlieder 1996);

Often more than one paradigm of formalization is used depending on the structural properties of the class of intended models. In Example 3 we analyze the application of the geometric, analytic, and qualitative paradigm to the formalization of the concept of cardinal directions in geographic space:

The geometric component:
The geometric component of the paradigm of formalization refers to the use of *relations and properties of spatial order* that remain invariant, for example, under cartographic projection transformations, in order to characterize configurations of spatial individuals.
The analytic component:
The use of frames of reference is used to structure the infinite, continuous set of points in the plane using a *finite sets of relations of order of points with respect to an oriented line*. The equivalence classes of points are called left and right half-plane. They correspond to the relation left-of-the-line and right-of-the-line that hold between points and lines. This concept can be extended by using more than one directed lines in different configurations, e.g., (Frank 1992), (Freksa 1992), (Hernandez 1994).
The qualitative component:

Consider the domain of angles between two directed lines in the Euclidean plane. Landmark values, that are, values in which represent significant changes, are: 0, 90, 180, 270 degree. With respect to these landmark values the infinite and continuous domain of angles between directed lines is divided into four landmark values and four intervals between them. If one directed line is adjusted to the north pole the landmark values are named North, West, South, East, the intervals between them are named North-West, South-West, South-East, North-East.

Example 3: The application of the geometric, analytic, and qualitative paradigm of formalization to cardinal directions

3 Formal Systems

We use formal systems to consider the following questions in an abstract and mathematically correct manner:

- Which strings of symbols are well-formed formulas ? (Syntax)
- How do these formulas relate to the world, i.e., to formal models of the world ? (Semantic)
- How can the relation between formulas and the world (e.g. the relation of truth) and relations between formulas (e.g. the relation of consequence) be computed and which properties does the computation have ? (Proof theory)
- How correct computation can be replaced by the notion of proof ? (Declarative programming)

3.1 Syntax

A formal language is a set of well-formed formulas (abbreviation WFFs). Well-formed formulas are defined by the syntax of a formal language. The syntax of a formal language is determined by an alphabet and rules of grammar.

The alphabet of a formal language is the set of primitive symbols. These symbols are used to compose well-formed formulas. One usually distinguishes between the alphabet of the base language and the alphabet of a particular theory. In the case of mathematical logic as base language the base-alphabet consists of variable symbols $x_0, x_1, \ldots, y_0, y_1, \ldots$, the logical constants `'True'` and `'False'`, the logical connectivities $\wedge, \vee, \neg, \rightarrow$ (read as 'and', 'or', 'not', 'implies'), logical quantors $\forall, \exists$ (read as 'for all', 'there exists'), and additional parenthesis. Non logical constants denote individuals, relations of functions specific to the theory of a particular domain (e.g. the theory of political subdivision of Austria in Example 4).

The grammar is a set of rules for creating syntactic correct expressions from the alphabet of the language. The formal definition of the syntax of a formal language allows to formally decide for each string of symbols of the alphabet of a formal language whether it is a well-formed formula, or not.

An alphabet of a formal language in which one could talk, i.e., make assertions, about aspects of the political subdivision of Austria shown in Figure 1, should contain the following set of symbols:

Non-logical constant symbols

"Austria", "Niederösterreich", "Oberösterreich", "Salzburg", "Tirol", "Vorarlberg", "Kärnten", "Steiermark", "Burgenland", "Wien", "Vienna", "St.Pölten", "Linz", "Salzburg_city", "Innsbruck", "Bregenz", "Klagenfurt", "Graz", "Eisenstadt", "Countries", "Province", "City", "Capital", "Neighbor", "Part_Of".

Well-Formed Formula

Well-formed formulas of a formal language of logic are either atomic or complex. Complex formulas are atomic formulas with negation, quantification or atomic formulas connected via logical connectivities.

Atomic formulas: "Country" ("Austria"). "City" ("Vienna").

"Neighbor" ("Tirol", "Burgenland"). "City"("Austria").

Complex formulas:

"Capital" ("Vienna", "Austria") ∧ "Capital" ("Vienna", "Wien").

∃x["Province"(x)∧∃y["City"(y)∧"Capital"(y,x)]].

Not well-formed formulas are for example:

"Neighbor"("Vienna"). "Country""Capital"(((.

Example 4: Alphabet and well-formed formulas of a formal language of political subdivision of Austria.

3.2 Semantic

The meaning of a well-formed formula of a formal language is defined in terms of relations to formal models. Formal models are represented by algebraic structures. An algebraic structure is a set of individuals and a finite number of relations and functions over the domain of the individuals. The relation between a WFF and an algebraic structure is established in a two-step process. First an interpretation must be established. An interpretation is a mapping that assigns to every non-logical symbol of the WFF a relation, operation, or individual of the algebraic structure. Then assignments from individuals of the set of individuals of the algebraic structure to the variables in the well-formed formula must be established. A WFF is satisfiable if an assignment can be found which makes the formula true. A WFF is valid in an algebraic structure if it is true under arbitrary assignments. An algebraic structure is a model of a WFF if the WFF is valid in this algebraic structure.

Consider the formal model in Example 1 and the fragment of a formal language in Example 4:

Satisfiability

of "Province"("Tirol")∧ "City"(y) ∧ "Capital"(y,"Tirol").

1. Interpretation: { ("Province" ↦ Province), ("City" ↦ City), ("Capital" ↦ Capital), "Tirol" ↦ Tirol, **...**}

2. Assignment: { (y ↦ Innsbruck) }

"Province"("Tirol")∧ "City"(y) ∧ "Capital"(y,"Tirol") is satisfied because:

(Tirol ∈ Province) AND (Innsbruck ∈ City) AND (<Innsbruck, Tirol> ∈ Capital)

Validity :

To prove the validity of an existential formula it is sufficient to provide one satisfying assignment.

∃y["Province"("Tirol")∧ "City"(y) ∧ "Capital"(y,"Tirol")] (in English: Tirol is a province which has a particular city as its capital) is valid because

"Province"("Tirol")∧ "City"(y) ∧ "Capital"(y,"Tirol") can be satisfied.

Example 5: Satisfiability and Validity of Formula

3.3 Proof Procedures

A proof procedure is a well-defined sequence of operations which can be used to decide whether a well-formed formula is valid. There are two main classes of proof procedures: model based (semantic) and calculus based (syntactic) procedures.

In model based proof procedures it is checked whether or not under a particular interpretation and assignments a formula is satisfied or not. Model based proof procedures depend on the existence of

finite models that are sufficient to provide the proof. A well-formed formula is proven to be valid if it can be satisfied under arbitrary assignments or if its negation cannot be satisfied at all.

Calculus based proof procedures are based on syntactic manipulations of formulas that preserve validity (rules of inference). They are based on the existence of well-formed formulas that are assumed to be valid. These special sentences are called axioms. A theorem is a well-formed formula that can be derived (proven) from the axioms by means of the rules of inference.

Properties of Proof Procedures

A proof procedure is sound if and only if every provable formula is valid. A proof procedure is complete if every valid formula is provable. Assume soundness and completeness of a given proof procedure. A proof procedure is decidable if and only if for *every* well-formed formula of the language it can be decided whether it is valid or not.

Assume a proof procedure that is sound, complete, decidable, and implemented on a computer. These proof procedures can be further characterized by evaluating the complexity of the algorithm processed by the procedure. The complexity is expressed in terms of the amount of resources, space (storage), and time required to perform the algorithm.

Most purely syntactic proof procedures require an amount of time to perform the prove that is too large for many practical purposes. On computers often model based proof procedures are implemented which are sufficient for a particular class of languages and problems. Often these proof procedures are incomplete.

Query processing in relational databases is an example for model based reasoning implemented on a computer (Example 6).

Relational databases can be considered as representations of formal models. Tuples of individuals are organized in tables. Tables are representations of the extension of a particular relation.

Cities
“Vienna”
“Eisenstadt”
…
“Graz”

Province
“Burgenland”
“Steiermark”
…
“Wien”

Capital	
“Vienna”	“Wien”
“Graz”	“Steiermark”
…	…
“Eisenstadt”	“Burgenland”

In order to perform reasoning, i.e., perform query processing, algebraic operations on the rows and columns in the table are performed. The query ‘What city is the capital of the province Steiermark’ can be expressed as statement of the language SQL: ‘SELECT City FROM Capital WHERE Province = “Steiermark”‘. The result of this query is either empty or consists of one or more than one tuples in which the component “Province” has the value “Steiermark”.
The non-empty result of the SQL query can be considered as a model based proof of the logical sentence
$\exists x[\text{“City”}(x) \wedge \text{”Province”}(\text{“Steiermark”}) \wedge \text{”Capital”}(x,\text{”Steiermark”})]$.

Example 6: Relational algebra as proof procedure

3.4 Declarative Programming Languages

Declarative programming languages are implementations of formal systems on a computer. Examples are the logical programming language PROLOG (Kowalski 1979) or the functional programming language HASKELL (Peterson, Hammond et al. 1996). PROLOG is based on predicate logic. HASKELL is based on the lambda calculus (Church 1941). Declarative programming languages have a well-defined formal semantic. They have implemented a proof procedure that is sound for the corresponding class of languages.

The important advantages of declarative programming languages are

- their declarative style of programming,
- their formal semantic, and
- their notion of proof.

The declarative style of programming allows to formalize problems abstractly in terms of logical formula. The meaning of these formulas is well defined within the corresponding semantic. Computation is replaced by the notion of proof and is guaranteed to be correct with respect to the underlying semantic.

Consider the declarative programming language PROLOG. A declarative program is a set of logical formulas $\{\Phi_1, \ldots, \Phi_n\}$. Computation is performed by proving whether a logical formula Φ_{n+1} is a consequence of the logical program $\Phi_1 \wedge \ldots \wedge \Phi_n$. This proof is performed by proving that the formula $\Phi_1 \wedge \ldots \wedge \Phi_n \wedge \neg\Phi_{n+1}$ cannot be satisfied in any model. Representing problems in terms of logical formulas allows to concentrate on the problem rather than to handle pointers, array indices, procedure calls, …

Of course declarative programming has also disadvantages. First: The complexity of their proof procedures is due to its generality rather high. Declarative programs are rather slow. Second: Declarative programming is not strictly logical and not totally independent of the computation (Kowalski 1979), (Peterson, Hammond et al. 1996).

The field of application of declarative programming is in the domain of problem specification, problem solving, prototyping rather than in final-product-programming (Car and Frank 1995), (Frank 1995), (Kowalski 1979).

4 On Using Formal Theories

4.1 Formal Theories

Formal theories are sets of well-formed formulas that characterize or represent formal models. The alphabet of the language of a formal theory contains symbols which are specific to the domain the well-formed formulas refer to. A formal theory of a model is the set of all well-formed formulas, which are valid in this model. The larger the theory the less models it will have, i.e., the better it characterizes the remaining models.

A formal theory is characterized by the following properties: A formal theory is *consistent* if it has at least one model. A formal theory is *finite* if it consists of a finite number of sentences. A theory is *complete* if for each well-formed formula Φ either Φ or $\neg\Phi$ belongs to the theory. A theory is *decidable* if for every formula Φ either Φ or $\neg\Phi$ can be proven. If there is a contradiction in the theory, i.e., the theory is inconsistent, i.e., it contains a sentence Φ and its negation $\neg\Phi$, then it does not have a model.

For our purposes consistent finite theories of finite models with sound and effectively decidable proof procedures are relevant.

Well-formed formulas which belong to the theory of the model in Example 1 are:
"Country"("Austria"), "Province"("Niederösterreich"), "Province"("Oberösterreich"),…, "City"("Vienna"), "City"("St.Pölten"),…,
"Capital"("Vienna","Austria"), "Capital"{"Vienna", "Wien"),…, "Neighbor"("Niederösterreich","Oberösterreich"),
"Neighbor"("Niederösterreich","Burgenland"),…, $\exists$x["Province"(x)$\wedge\exists$y["City"(y)$\wedge$"Capital"(y,x)]],
$\forall$x["Province"(x)$\rightarrow\exists$y["City"(y)$\wedge$"Capital"(y,x)]],
$\forall$x$\forall$y ["Capital"(x,y) $\rightarrow$ "Part_Of"(x,y)]
$\forall$x$\forall$y["Neighbor"(x,y)$\rightarrow$"Neighbor"(y,x)], $\forall$x $\neg$"Neighbor"(x,x), …

…
The theory (fragment) is consistent because it has Example 1 as model.

Example 7: Well-formed formulas which belong to the formal theory of the model of Example 1

4.2 Axiomatic Theories

Formal theories are possibly infinite sets of well-formed formulas. They can be represented by axiomatic systems. "By an axiomatic system we understand the representation of a theory in such way that certain sentences of this theory (the axioms) are placed in the beginning, and from them further sentences (the theorems) are derived by means of logical deduction ... " (Carnap 1958, pg. 171).

Axioms are special sentences of a formal theory. Their validity is assumed without a proof. They characterize the properties of the axiomatic primitive constants. Axiomatic primitive constants are constant, predicate, or function symbols of the alphabet of the theory which are given without any definition, e.g. the symbols "Part_Of", "Province", "Austria", "City", "Vienna",… in Example 4. A formal theory, i.e. the set of all WFFs valid in all models of the theory, can be derived from the axioms by means of syntactic or semantic proof procedures.

Consider the following sets of axioms postulating a property of "Part_Of", the relation between "Capital" and "Part_Of", and pieces of knowledge about particular cities and provinces (1..4), the syntactic rule of inference modus ponens (MP). The theorems 5 and 6 can be derived. The axioms are assumed to be valid, and hence provable by definition.
Property of transitivity of "Part_Of"
(1) $\forall x \forall y \forall z$ ["Part_Of"(x,y)$\wedge$ "Part_Of"(y,z) $\rightarrow$ "Part_Of"(x,z)]
Relation between "Capital" and "Part_Of"
(2) $\forall x \forall y$ ["Capital"(x,y) $\rightarrow$ "Part_Of"(x,y)]
Knowledge about Cities and provinces of Austria
(3) "Capital"("Vienna","Wien")
(4) "Part_Of"("Wien","Austria")
The rule of inference
(MP) if p is provable, and it is provable that p implies q ($p \rightarrow q$), then
q is provable (p and q are variables that range over sentences)
Theorems:
(5) "Part_Of("Vienna","Wien"). (3+2+MP)
(6) "Part_Of"("Vienna", "Austria") (5+4+1+MP)
It is obvious that from this set of axioms not the whole theory of the model in Example 1 can be derived. The set of axioms is incomplete. It represents only a part of the whole theory.

Example 8: Axiomatic Theories

There are, for example, axiom systems for the natural number (Peano axioms) (McCoy and Berger 1977), elementary geometry (Tarski 1959), the topology of point sets (Alexandroff 1961), …

4.3 The Method of Deductive Science

The essence of the method of deductive science is to formally represent a scientific discipline in terms of an axiomatic theory (Tarski 1941), (Carnap 1958). A small number of axiomatic primitives are assumed without (formally) explaining their meaning. The meaning of these primitives is assumed to immediately understandable. These primitives denote ontologically primitive individuals, relations, or operations (section 2.2). Based on these primitives further primitives are defined. The choice of axiomatic primitives also depends on the paradigm of formalization, e.g. the choice of points *and* lines in Example 3. The paradigm of formalization determines which derived primitives are defined in terms of the axiomatic primitives, e.g. classes of equivalent individuals (`left-of-the-line` and `right-of-the-line`,…) in Example 3. The paradigm of formalization also determines how the definitions are stated, e.g. in term of equal location (to be *located* left of the line). Properties of the

axiomatic primitives which are doubtless true are formalized in terms of axioms. Axioms are sentences which validity is assumed without any proof.

A scientific discipline is represented by the set of theorems that can be proven from the axioms within a sound logical and mathematical environment. Proofs of the *general laws of the domain* are often per formed on an abstract and purely formal basis only assuming the axioms and theorems of the formal theory, axioms and theorems of logic, and axioms and theorems of mathematics. Due to the character of the deductive method we can be sure that the results we get are valid under the assumption that the axioms from which they are based on are valid.

4.4 Model Based Reasoning

Proofs about *particular configurations of individuals* (for example stored in a database) are often performed by model proof procedures. Model based reasoning is performed by model construction and by model inspection. Model inspection means to check the validity or satisfiability of a formula in a particular model. Model construction means to construct a model in which a given set of formulas can be satisfied. In order to be representable on a computer the representation of the models necessarily need to be *finite and discrete* (Section 2).

Model based reasoning in finite representations is based on three assumptions that were first investigated in the context of logical considerations of relational databases (Reiter 1984):

1. "The *closed world assumption* (CWA), …, which states that facts not known to be true are assumed to be false …
2. The *unique name assumption*, which states that individuals with different names are different.
3. The *domain closure assumption*, which states that there are no other individuals than those in the database." (Gallaire, Minker et al. 1984, pg. 158)

Under these assumptions *finite* representations can be considered as formal models of formal theories and the validity or satisfiability of expressions of the language of the theory can be proved semantically by model inspection.

A consequence of the CWA is that the reasoning process becomes non-monotonic. Monotony of the proof procedure means that all proofs remain valid if new knowledge is added to a representation. Non monotony has the consequence that all former proofs need to be revised in the case new knowledge is added. This is also a well-known problem in spatial databases.

Consider the formal model in Example 1 as representation of the political subdivision of Austria in Figure 1. The unique name assumption ensures that different cities and provinces have different names, i.e., are represented by different individuals in the model. The domain closure assumption ensures that all cities and provinces are represented in the model. With the help the closed world assumption we can derive ¬"Capital"("Vienna","Steiermark") because a tuple <Vienna,Steiermark> does not exist in the relation Capital.

Imagine that we did not know about the province Vorarlberg and its capital Bregenz. Consequently it would not occur in Figure 1. The corresponding individuals would not belong to the set of individuals and the tuple <Bregenz,Vorarlberg> would not belong to the relation capital of the model in Example 1. Consequently we could derive ¬"Capital"("Bregenz","Vorarlberg"). As soon as we get knowledge about Bregenz and Vorarlberg and incorperate it into the model the former valid conclusion ¬"Capital"("Bregenz","Vorarlberg") becomes invalid. Consequently model based reasoning is non-monotone. If the invalid conclusion is not removed from the theory the whole theory becomes inconsistent.

Example 9: Model Based Reasoning

5 GIS as Implementations of Formal Theories of Geographic Space

Theories of geometry are formal theories of spatial aspects of configurations of individuals that remain invariant under classes of transformations (Klein 1872). There exist different theories of geometry. Theories of geometry can be classified:

- concerning their formalized primitives, e.g. points and lines, points and sets of points, regions;
- concerning the class of transformations under which the formalized properties remain invariant, e.g. invariance under continuous transformation, rotation, translation, ….

Theories of geometry have different properties and are used, i.e. implemented, in GIS for different purposes.

Theories of the geometry of points and sets of points are the basis of vector systems. Spatial phenomena are modeled as "geometrical figures definable in elementary geometry, i.e., first-order logic over real numbers with addition and multiplication. These figures are called semi-*algebraic sets* in real algebraic geometry … A formula in the first-order logic of the reals , a *real formula* for short, is built from *atomic real formula* using boolean operators and quantification over real variables" (Vandeurzen, Gyssens et al. 1995, pg. 16). The geometric figures represented by these formula are the sets of points which satisfy a given *real formula*:

$F = \{ (x_1,\ldots,x_m) \mid \Phi(x_1,\ldots,x_m) \}$

F is a set of points in m-dimensional space. Each point is denoted by a m-ary coordinate tuple $(x_1,\ldots,x_m)$. Φ is a *real formula* with the real variables $x_1,\ldots,x_m$.

In vector systems in particular linear figures, i.e., semi linear sets, are used. Semi linear sets are defined by means of linear formula. "Most spatial data types … are sub-types of semi-linear sets … the data types point, line, and polygon … can be seen as 0-, 1-, and 2-dimensional polytropes" (Vandeurzen, Gyssens et al. 1995, pg. 17). Properties of geometric figures and relations between them remain invariant under linear transformations, e.g. translation, rotation.

In (Egenhofer and Herring 1990, Egenhofer, 1993 #440) geometric relations between figures that are identified with topologically defined sets of points were formalized. This approach is based on the mathematical theory of point set topology (Alexandroff 1961). In point set topology between the interior and the boundary subsets of a set of points can be distinguished. The topological relations between figures are defined in terms of the content of the intersection of the interior and boundary subsets of the corresponding point-sets. Topological relations remain invariant under continuous transformations. Applications of this approaches the maintenance of the topological consistence of spatial databases (Egenhofer and Sharma 1993) and the evaluation of the meaning of spatial predicates in spatial queries (Mark and Egenhofer 1994).

Theories of geometry of configurations of regional primitives were proposed in (Randell, Cui et al. 1992), (Pratt and Schoop 1997).

Spatial databases are representations of formal models of formal theories of the geometry of geographic space. Vector databases are finite and discrete representations of polytropes. Polytropes are representations of infinite and continuous semi-linear sets of points. The semantic of spatial databases, i.e., the relations between data items stored in a database and formal theories of geometry is subject of ongoing research (Kuijpers, Paredaens et al. 1995), (Vandeurzen, Gyssens et al. 1995), (Paredaens, Van den Bussche et al. 1994), (Pratt and Lemon 1997).

GIS operations (Laurini and Thompson 1994) can either be considered as proofs, e.g., query processing (Paredaens, Van den Bussche et al. 1994), (Afrati, Cosmadakis et al. 1994), or need to be provable consistent with the laws of the corresponding geometry. For example: Topological relations between individuals of a spatial configuration must remain invariant under affine transformation that are applied to the database representation of the configuration (Frank and Kuhn 1986).

The abstract view on GIS as implementations of formal theories of the geometry of geographic space allows to consider:

- Spatial databases as representations of formal models of formal theories of geographic space.
- The underlying logical concepts encoded in the program code and the data structures as a theory of geographic space.
- GIS operations as model based proof procedures.

Consequently we can apply notions of

- completeness and consistency to characterize the underlying theory of geographic space,
- formal semantic to characterize spatial databases,
- soundness and completeness in order to characterize GIS operations.

6 Conclusions

In this paper we explained important notions of formal systems and logic. We discussed the application of formal methods to phenomena in geographic space. We considered geographic information systems (GIS) as implementations of formal theories of geographic space. We showed that there are aspects of GIS that can be formalized in formal systems.

We argued that to formalize aspects of GIS as formal systems of logic has the following advantages:

- To axiomize a theory of a domain means to make the laws that govern the domain explicit.
- Due to the character of the deductive method we can be sure that the results we get are valid if the assumptions on which they are based are valid.
- The notion of proof allows to abstract from questions specific to the implementation (declarative programming).
- Considering GIS as a formal system provides a sound meta-theoretical view of database, query processing, and operational aspects of GIS.

To apply computer based formal methods to phenomena in geographic space infinite and continuous domains need to be represented by finite and discrete models. This transition has to preserve the domain properties that are relevant for the particular purpose. This is achieved in a process of abstraction and discretization. This process is controlled by the ontology and the paradigms of formalization.

The paradigm of formalization controls the transition from infinite and continuous domains of individuals to equivalence classes of individuals. Correspondence between both domains can be preserved if the definition of the equivalence classes is consistent with the ontological significant structures and distinctions.

We point out that the application of formal methods is rather in the prototypical design and the specification and evaluation of GIS applications rather than in implementation. This is due to the high complexity of the proof component. In the process of design and specification this method allows to concentrate on relevant aspects of the formalization.

The following open questions in the application of formal methods to GIS need to be considered:

- A theory of paradigms of formalization is needed.
- Different formal theories can be formulated to characterize the same model. The suitability of a particular formal theory for a particular purpose needs to be evaluated.
- The process of design of a formal theory consisting of
    1. the choice of an ontology,
    2. the choice of paradigms of formalization,
    3. the choice of formal languages and reasoning techniques

    needs to be investigated.

7 References

Afrati, F., S. S. Cosmadakis, et al. (1994). Linear vs. Polynomial Constraints in Database Query Languages. PPCP 1994.

Alexandroff, P. (1961). Elementary Concepts of Topology. New York, NY, Dover Publications.

Barwise, J. (1977). An Introduction to First-Order Logic. Handbook of Mathematical Logic. J. Barwise.

Car, A. and A. U. Frank (1995). "Formalization of Conceptual Models for GIS using Gofer." Comput. Environ. and Urban Systems **19**(2): 89-98.

Carnap, R. (1958). Introduction to Symbolic Logic and its Applications. New York, Dover Publications, Inc.

Church, A. (1941). The Calculi of Lambda-Conversation. Princeton, NY, Princeton University Press.

Curry, H. B., R. Feys, et al. (1958). Formal Systems. Combinatory Logic. H. B. Curry, R. Feys and W. Carig, North Holland.

Egenhofer, M. and J. R. Herring (1990). A Mathematical Framework for the Definition of Topological Relationships. 4th International Symposium on Spatial Data Handling, Zurich, Switzerland, IGU.

Egenhofer, M. and J. Sharma (1993). Topological Consistency. Proceedings of the 5th International Symposium on Spatial Data Handling, Charleston, IGU Commission of GIS.

Egenhofer, M. J. and D. M. Mark (1995). Naive Geography. Spatial Information Theory, A Theoretical Basis for GIS, Springer.

Frank, A. (1992). "Qualitative Spatial Reasoing about Distances and Directions in Geographic Space." Journal of Visual Languages and Computing **3**: 343-371.

Frank, A. U. (1995). Specifying Open GIS with Functional Languages. SSD'95 Proceedings, Springer-Verlag.

Frank, A. U. and W. Kuhn (1986). Cell Graphs: A Provable Correct Method for the Storage of Geometry. Second International Symposium on Spatial Data Handling, Seattle, WA, IGU.

Freksa, C. (1992). Using Orientation Information for Qualitative Spatial Reasoning. Theories and Methods of Spatio-Temporal Reasoning in Geographic Space. A. U. Frank, I. Campari and U. Formentini, Springer.

Gallaire, H., J. Minker, et al. (1984). "Logic and Databases: A Deductive Approach." Computing Surveys **16**(2).

Guarino, N. (1994). The Ontological Level. Philosophy and the Cognitive Science. R. Casati, B. Smith and G. White. Vienna, Hoelder-Pichler-Tempsky.

Hernandez, D. (1994). Qualitative Spatial Reasoning, Springer.

Kernighan, B. W. and P. J. Plauger (1978). The C Programming Language, Prentice Hall Software Series.

Klein, F. (1872). Vergleichende Betrachtungen über neuere geometrische Forschungen. Erlangen, Germany.

Kowalski, R. (1979). Logic for Problem Solving, North Holland.

Kuijpers, B., J. Paredaens, et al. (1995). Semantics in Spatial Databases. Semantics in Databases. B. Thalheim, Springer.

Kuipers, B. (1994). Qualitative Reasoning, MIT Press.

Laurini, R. and D. Thompson (1994). Fundamentals of Spatial Information Systems, Academic Press.

Mark, D. M. and M. J. Egenhofer (1994). Calebrating the Meanings of Spatial Predicates from Natural Languages: Line-Region Relations. SDH '94, Edinburgh, Taylor & Francis.

McCoy, N. H. and T. R. Berger (1977). Algebra; Groups, Rings, and other Topics, Allyns and Bacon, Inc.

Paredaens, J., J. Van den Bussche, et al. (1994). Towards a Theory of Spatial Database Queries. Thirteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Minneapolis.

Peterson, J., K. Hammond, et al. (1996). The Haskell Report.

Pratt, I. and O. Lemon (1997). Ontologies for Plane, Polygonal Merotopology. Manchester, Department of Computer Science, University of Manchester.

Pratt, I. and D. Schoop (1997). A complete axiom system for polygonal merotopology of the real plane. Manchester, Department of Computer Science.

Randell, D. A., Z. Cui, et al. (1992). A Spatial Logic based on Regions and Connection. 3rd Int. Conf on Knowledge Representation and Reasoning, Boston.

Reiter, R. (1984). Towards a Logical Reconstruction of Relational Database Theory. On Conceptual Modeling. M. Brodie, J. Mylopoulos and J. W. Schmidt, Springer.

Schlieder, C. (1996). Qualitative Shape Representation. Geographic Objects with Indeterminate Boundaries. P. Burrough and A. U. Frank, Taylor & Francis.

Tarski, A. (1941). Introduction to Logic and to the Methodology of Deductive Sciences, Oxford University Press.

Tarski, A. (1959). What is Elementary Geometry. The Axiomatic Method, with special Reference to Geometry and Physics. L. Henkin, P. Suppes and A. Tarski. Amsterdam, North Holland Publishing Company.

Vandeurzen, L., M. Gyssens, et al. (1995). On the Desirability and Limitations of Linear Spatial Database Models. Fourth International Symposium on Large Spatial Databases - SSD95, Portland, ME, Springer Verlag, Heidelberg.