Spatial Data Types: Conceptual Foundation for the Design and Implementation of Spatial Database Systems and GIS

Markus Schneider

FernUniversität Hagen
Praktische Informatik IV
D-58084 Hagen
Germany
markus.schneider@fernuni-hagen.de

Abstract

Spatial database systems and Geographic Information Systems as their most important application aim at storing, retrieving, manipulating, querying, and analysing *geometric data*. Research has shown that special data types are necessary to model geometry and to suitably represent geometric data in database systems. These data types are usually called *spatial data types*, such as *point*, *line*, and *region* but also include more complex types like *partitions* and *graphs* (*networks*). Spatial data types provide a fundamental abstraction for modeling the geometric structure of objects in space, their relationships, properties and operations. Their definition is to a large degree responsible for a successful design of spatial data models and the performance of spatial database systems and exerts a great influence on the expressive power of spatial query languages. This is true regardless of whether a DBMS uses a relational, complex object, object-oriented, or some other kind of data model. Hence, the definition and implementation of spatial data types is probably the most fundamental issue in the development of spatial DBMS. Consequently, their understanding is a prerequisite for an effective construction of important components of a spatial database system (like spatial index structures, optimizers for spatial data, spatial query languages, storage management, and graphical user interfaces) and for a cooperation with extensible DBMS providing spatial type extension packages (like spatial data blades and cartridges).

The goal of this tutorial is to present the state of the art in the design and implementation of spatial data types. First, we summarize the modeling process for phenomena in space in a three-level model and categorize the treatment of spatial data types with regard to this model. Then we pose design criteria for the types and analyse current proposals for them according to these criteria. Furthermore, we classify the proposed types and the operations defined on them from different perspectives. Our main interest is directed towards approaches which provide a formal definition of the semantics of spatial data types and which offer methods for their numerically and topologically robust implementation.

Contents

1 What are Spatial Data Types (SDTs)?

Spatial data types

- ... are special data types needed to model geometry and to suitably represent geometric data in database systems
- Examples: *point*, *line*, *region*; *partitions* (*maps*), *graphs* (*networks*)
- ... provide a *fundamental abstraction* for modeling the geometric structure of *objects* in space, their relationships, properties, and operations
- ... are an important part of the data model and the implementation of a spatial DBMS

The definition of SDTs

- ... is to a large degree responsible for a successful design of spatial data models
- ... decisively affects the performance of spatial database systems
- ... exerts a great influence on the expressiveness of spatial query languages
- ... should be independent from the data model used by a DBMS

Conclusions

- An understanding of SDTs is a prerequisite
  - for an effective construction of important components of a spatial database system
    $\rightarrow$ spatial index structures, optimizers for spatial data, spatial query languages, storage management, graphical user interfaces
  - for a cooperation with extensible DBMS providing spatial type extension packages
    $\rightarrow$ spatial data blades, cartridges
- *The definition and implementation of spatial data types is probably the most fundamental issue in the development of spatial database systems.*

Focus of this tutorial: present the state of the art in the design and implementation of spatial data types

Contents of this tutorial

2 Foundations of Spatial Data Modeling

3 Spatial Data Models and Type Systems

4 Formal Definition Methods

5 Tools for Implementing SDTs: Data Structures and Algorithms

6 Other Interesting Issues and Researchs Trends

Tutorial based on the book:

Markus Schneider, *Spatial Data Types for Database Systems - Finite Resolution Geometry for Geographic Information Systems*, LNCS 1288, Springer Verlag, 1997.

2 Foundations of Spatial Data Modeling

2.1 What Needs to Be Represented?

2.2 A Three-Level Model for Phenomena in Space

2.3 Design Criteria for Modeling Spatial Data Types

2.4 Closure Properties and Geometric Consistency

2.5 Organizing the Underlying Space: Euclidean Geometry versus Discrete Geometric Bases

2.6 ADTs in Databases for Supporting Data Model Independence

2.7 Integrating Spatial Data Types into a DBMS Data Model

2.1 What Needs to be Represented?

Two views of spatial phenomena:

- objects in space (entity-oriented / feature-based view)

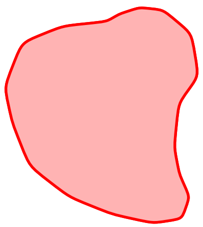
  $\rightarrow$ vector data, spatial database systems

- space itself (space-oriented / position-based view)

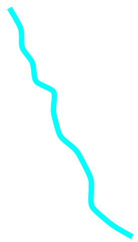
  $\rightarrow$ raster data, image database systems

Objects in space

city Berlin, pop = 4000000, ..., area =



highway A45, ..., route =



Space

Statement about every point in space

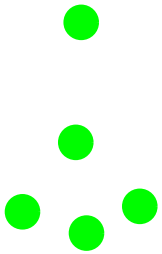
- land use maps (“thematic maps”)
- partitions into states, districts, ...

We consider:

- modeling single, self-contained objects
- modeling spatially related collections of objects

Fundamental abstractions for modeling single, self-contained objects

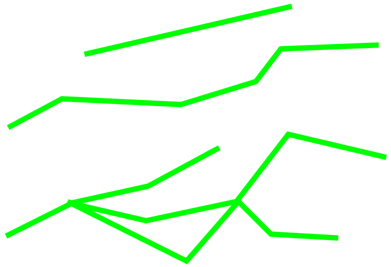
point



city
castle
lighthouse
church

(location of object in space but not its extent relevant)

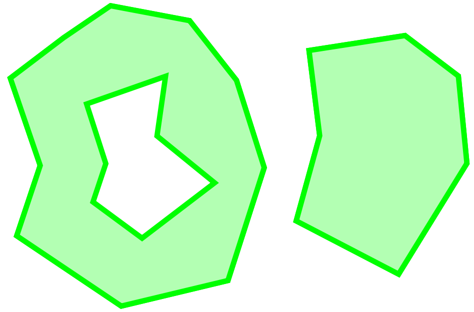
line



highway
river
cable
route

(connections in space, movement through space)

region

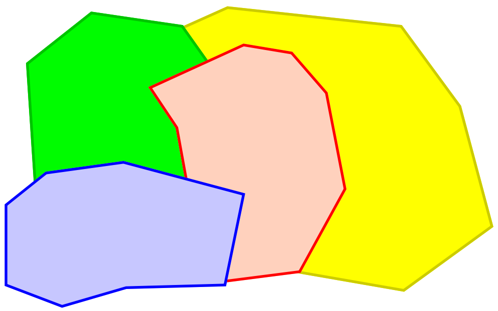


city
lake
district
forest
cornfield

(extent of an object relevant)

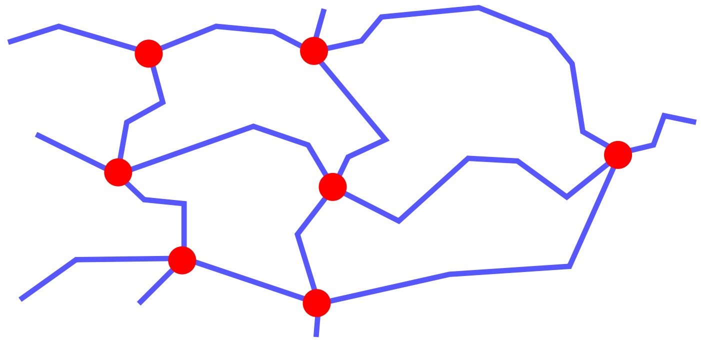
Fundamental abstractions for modeling spatially related collections of objects

partition



land use
districts
wards
countries
speech areas

Spatially embedded *network* (graph)



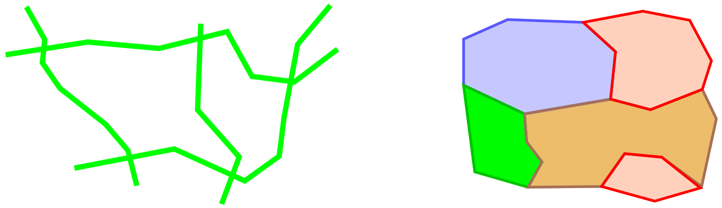
highways
railways
rivers
electricity
phone

Others: nested partitions, digital terrain models

2.2 A Three-Level Model for Phenomena in Space

Structure objects

Structure modeling

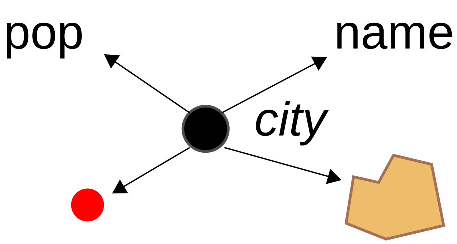


Structure types: sets, sequences, partitions, networks

Operations: overlay, shortest_path

Spatially-referenced objects

Object modeling

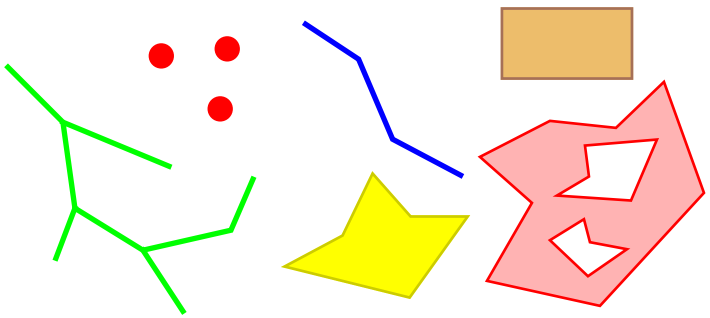


Object types: city, state, river

Operations: lies_in: city → state, flow: river → line)

Spatial objects

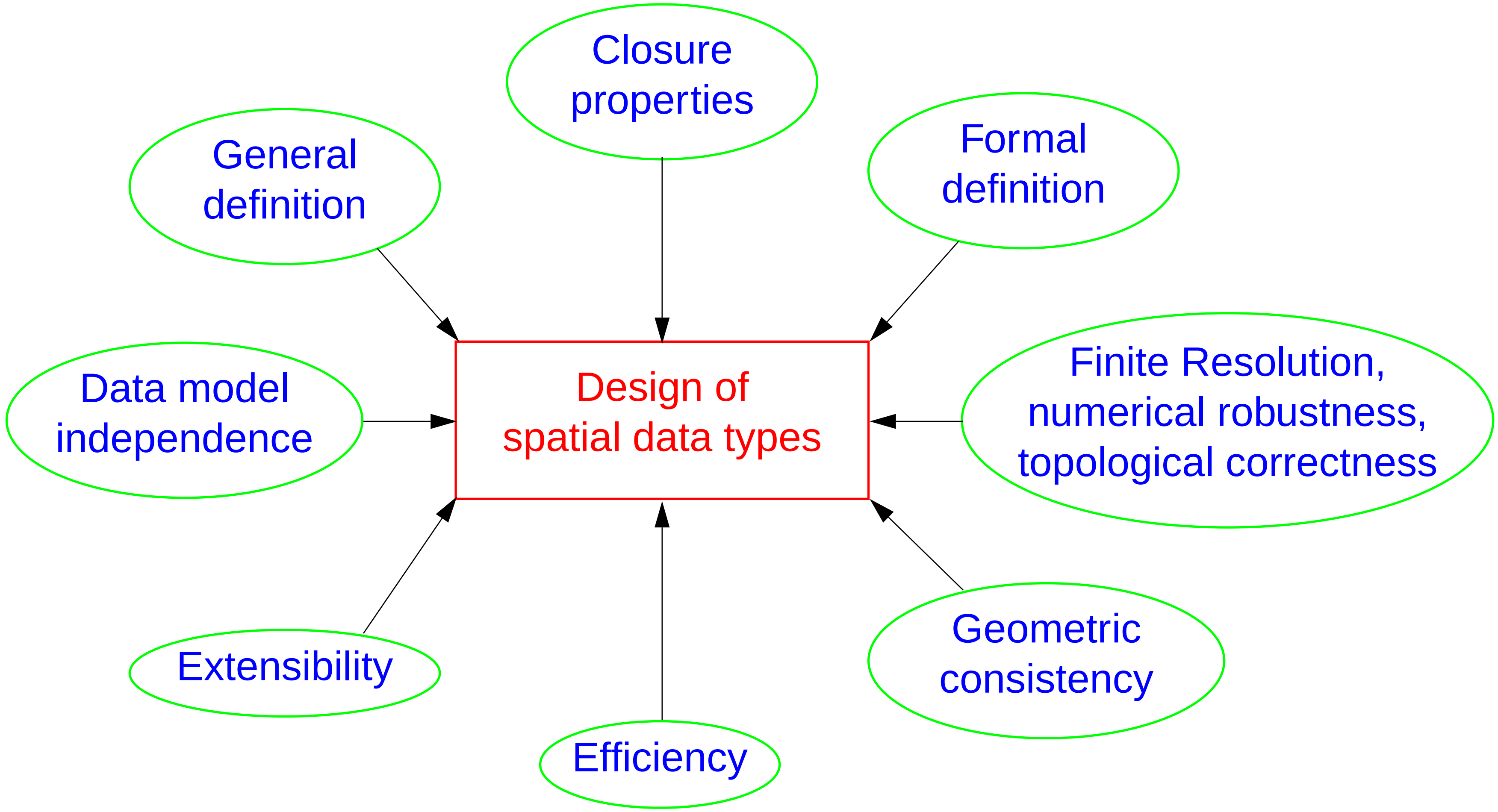
Spatial modeling



Spatial data types: point, line, polygon

Operations: point-in-polygon test, intersection

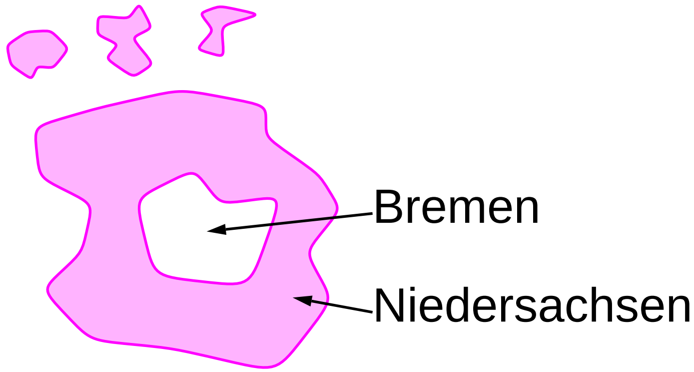
2.3 Design Criteria for Modeling Spatial Data Types



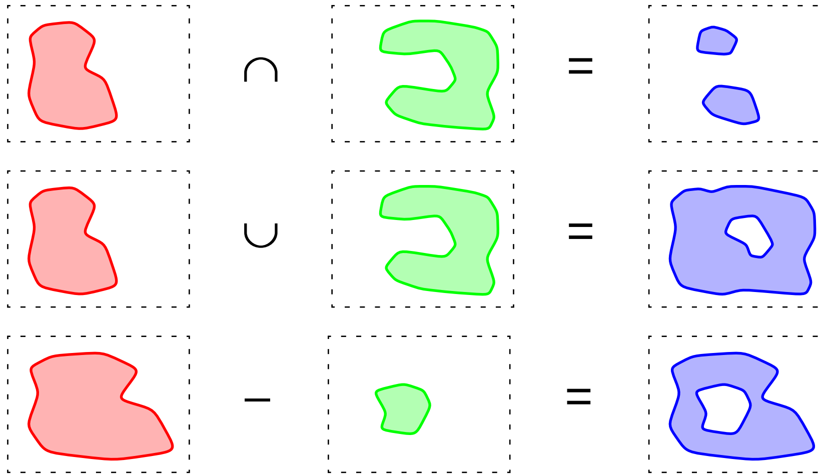
2.4 Closure Properties and Geometric Consistency

General definition/structure of spatial objects

application-driven requirements



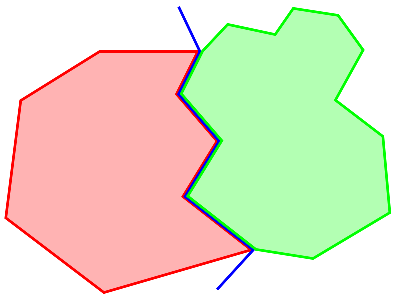
formal requirements



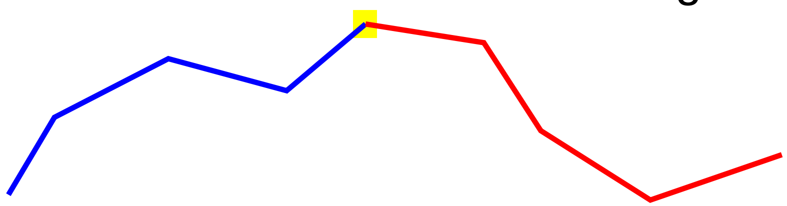
→ spatial objects must be closed under set operations on the underlying point sets

Support of geometric consistency constraints for spatially related objects

adjacent regions



meeting lines



→ SDT definition must offer facilities to enforce such consistency contraints

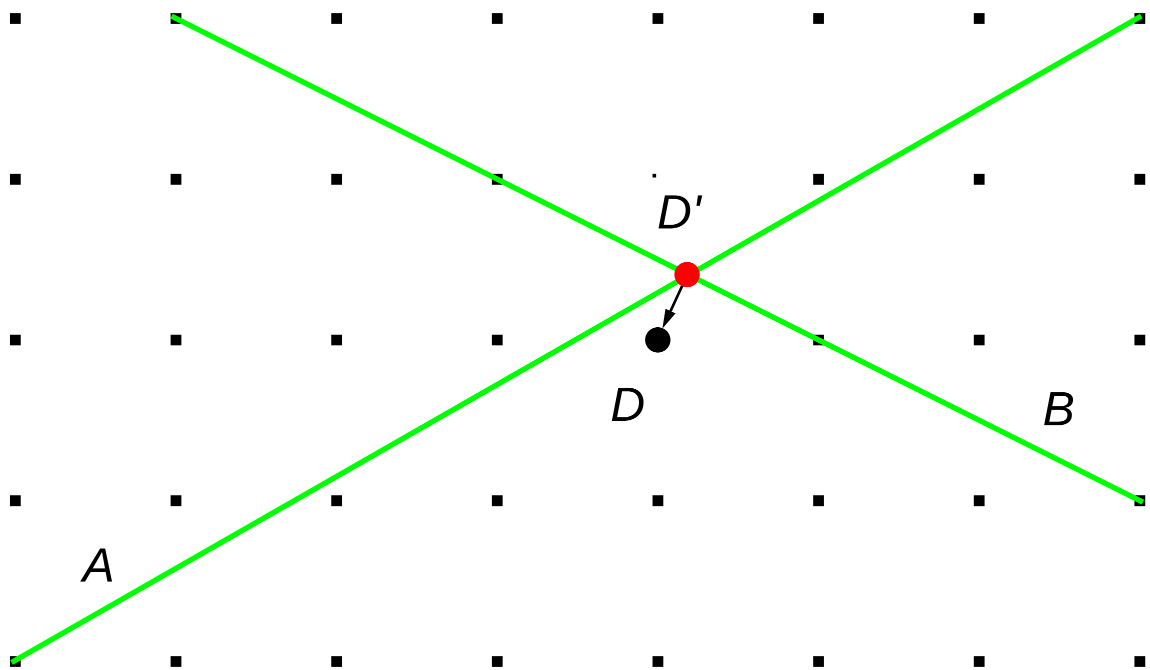
2.5 Organizing the Underlying Space: Euclidean Geometry versus Discrete Geometric Bases

Euclidean space is continuous ($p = (x, y) \in \mathbb{R}^2$)

- basis of Computational Geometry algorithms

But: computer numbers are finite and discrete ($p = (x, y) \in \texttt{real} \times \texttt{real}$)

- $\rightarrow$ numerical rounding errors $\rightarrow$ topological inconsistencies and degeneracies



Is *D* *on* *A*?

Is *D* *properly contained* in the area below *A* and *B*?

What happens if there is a segment *C* between *D* and *D'*?

$\rightarrow$ formal SDT definitions must bear in mind the finite representations available in computers

Solution: avoid computation of any new intersection points within geometric operations

Definition of spatial types and operations

Treatment of numerical problems upon updates on the geometric basis

Two approaches:

- *Simplicial complexes*

  Frank & Kuhn 1986

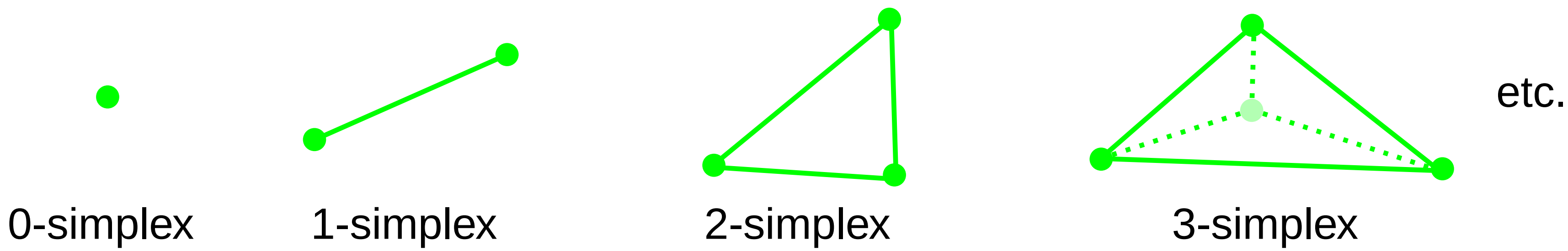
  Egenhofer, Frank & Jackson 1989
- *Realms*

  Güting & Schneider 1993

  Schneider 1997

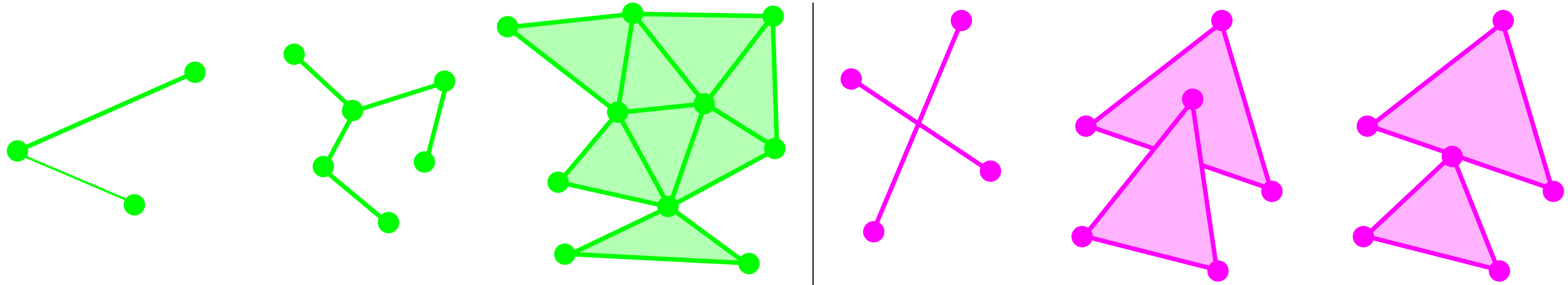
Simplicial Complexes

- topological relations are separately recorded and independent of metric positions
- use of *k-simplices* for representing minimal spatial objects of dimension *k*
  - construction rule: *k*-simplex consists of *k*+1 simplices of dimension *k*–1
  - component of a simplex is called *face*



- two completeness principles
  - completeness of incidence: the intersection of two *k*-simplices is either empty or a face of both simplices
    $\rightarrow$ no line intersection at points which are not start or end points of the lines, no two geometric objects may exist at the same location (geometry only recorded once)
  - completeness of inclusion: every *k*-simplex is a face of a (k+1)-simplex
    $\rightarrow$ all point are end points of lines, all lines are boundaries of triangles, etc.; no isolated points, no lines which are not part of a boundary

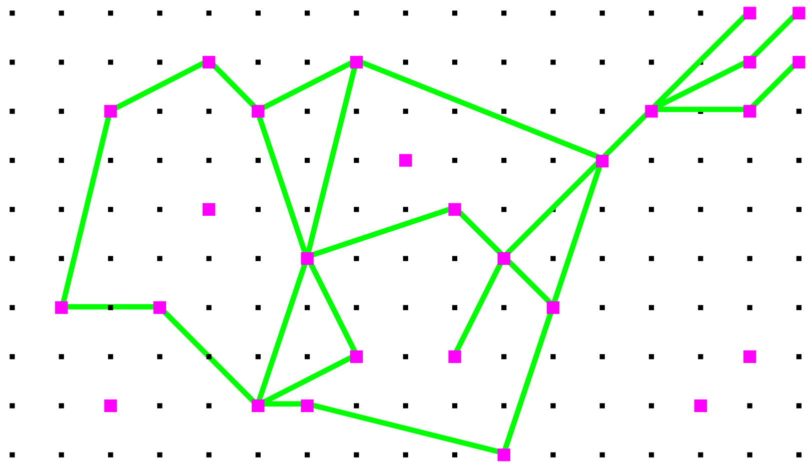
- *Simplicial complex*: finite set of simplices such that the intersection of any two simplices is empty or a face

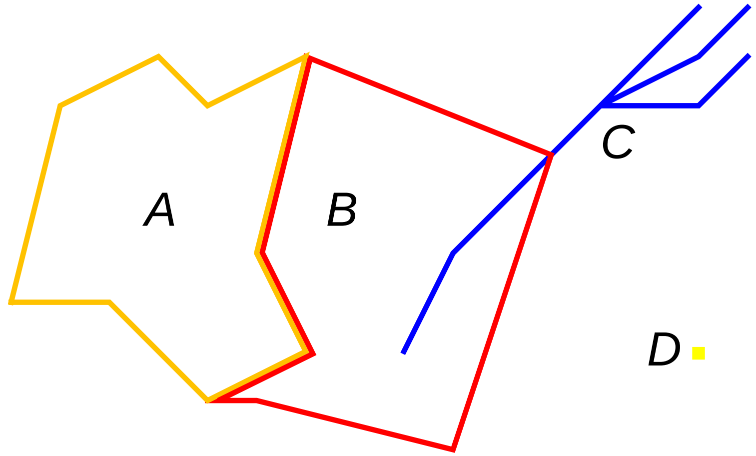


correct simplicial complexes — no simplicial complexes

- Advantages
  - maintenance of topological consistency
  - approach fulfils closure properties
- Drawbacks
  - unfortunately: no spatial algebra has been defined on top of this approach
  - triangulation of space would lead to space-consuming representations of spatial objects
  - no treatment of numerical problems: additional data structures needed to realize (at least imprecise) metric operations

Realms

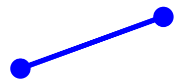




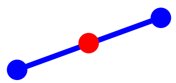
Realm (intuitive notion): description of the complete underlying geometry (all points and lines) of an application or application space

Realm (formally): A finite set of points and *non-intersecting* line segments defined over a grid such that

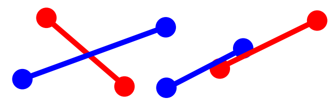
- each point and each end point of a segment is a grid point
- each end point of a segment is also a point of the realm



- no realm point lies within a segment



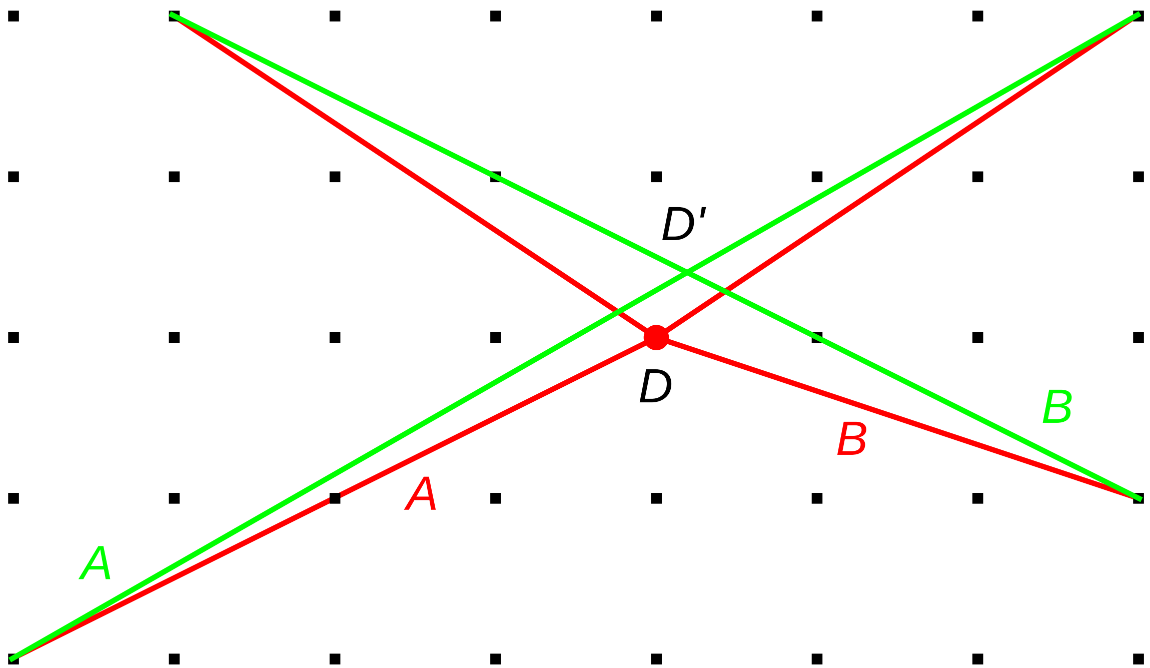
- any two distinct segments do neither properly intersect nor overlap



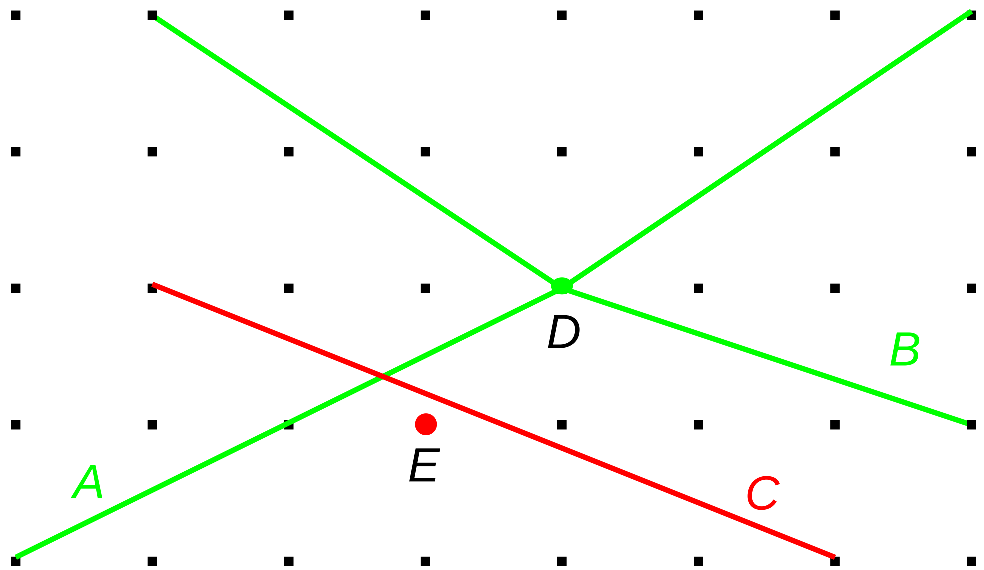
→ A realm is a *spatially embedded planar graph*

All numerical problems are treated *below* the realm layer:

- input: application data that are sets of points and *intersecting* line segments
- output: “realmified” data that have become acquainted with each other
- basic idea: slightly distort/perturb both segments



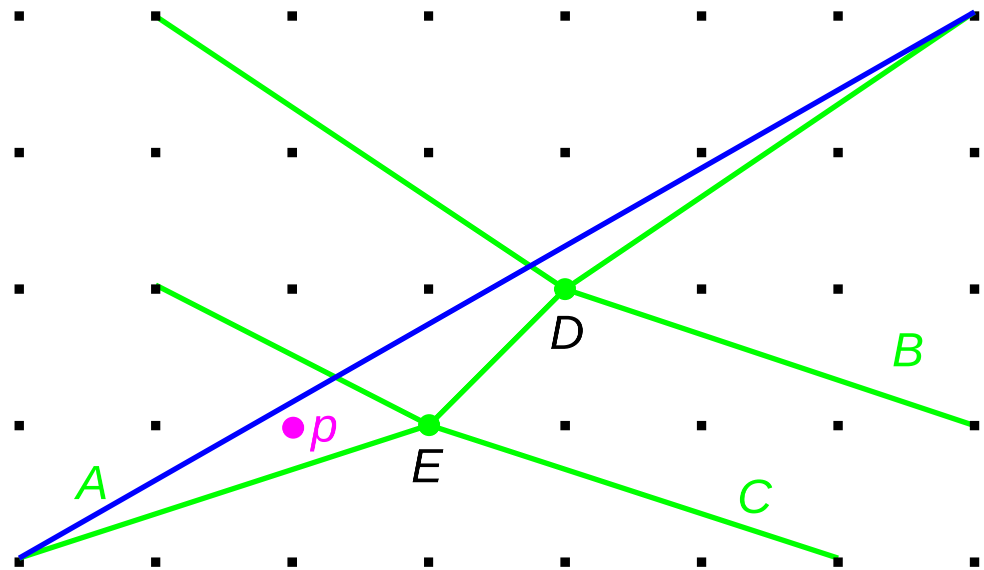
Good solution?



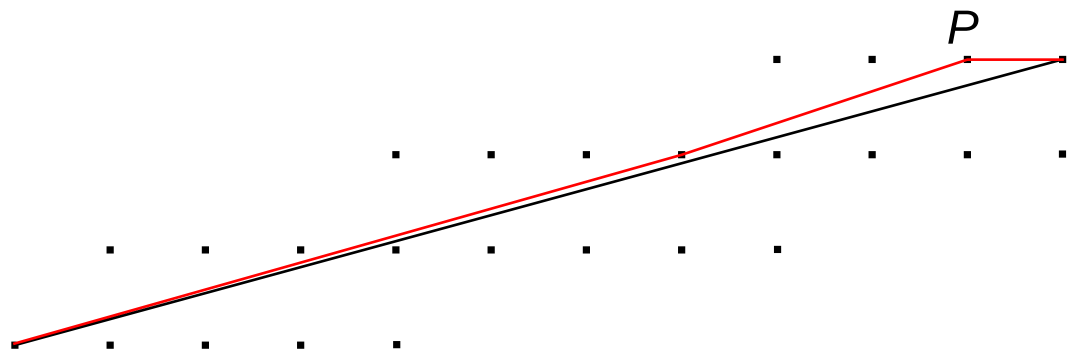
Intersection of segment *A* with some further segment *C*

Observations

- Segments can move far away from their original position by iterated intersections!
- Topological errors can occur: point *p* is now on the wrong side of *A*!



Solution: *redrawing* of segments within their envelope (Greene & Yao 1986)



Segments

- are "caught" within their envelope
- can never cross a grid point

Advantages of the realm concept

- definition of distinct SDTs on a common domain, guarantee of closure properties
- protection of geometric computation in query processing from problems of numerical robustness and topological correctness
- enforcement of geometric consistency of related spatial objects

Disadvantages of the realm concept

- no SDT operations possible that create new geometries (leave the realm closure), e.g., *convex_hull*, *voronoi*
- integration of realms into database systems somewhat difficult, propagation of realm updates to realm-based attribute values in database objects

2.6 ADTs in Databases for Supporting Data Model Independence

Modeling aspects

- Separation of DBMS data model and application-specific data types/algebras
- Modularity, conceptual clarity
- Reusability of ADTs for different DBMS data models
- Extensibility of DBMS data models

Implementation aspects

- Modularity, information hiding, exchange of implementations
- Employment of specialized methods (e.g. Computational Geometry for SDTs)
- Efficiency of data structures for data types and algorithms for operations

2.7 Integrating Spatial Data Types into a DBMS Data Model

Integration of single, self-contained spatial objects

- can be realized in a data model independent way ($\rightarrow$ ADTs)
- Basic concept: represent "spatial objects" (i.e., points, lines, regions) by *objects* of the DBMS data model *with at least one SDT attribute*
- DBMS data model must be open for new, user-defined types

  $\rightarrow$ ADT support, $\rightarrow$ data model independence, $\rightarrow$ extensibility
- Example for the relational model:

```
relation states(sname: string, area: region, spop: integer)
relation cities( cname: string, center: point, extent: region,
                 cpop: integer)
relation rivers(rname: string, route: line)
```

Integration of spatially related collections of objects

- not data model independent

- partitions
  - set of database objects with *region* attribute?
  - loss of information: disjointedness or adjacency of regions cannot be modeled
  - Güting 1988: SDT *area*, but: no support of this integrity constraint by the DBMS

- networks
  - not much research on *spatially embedded networks*
  - e.g., Güting 1994: GraphDB with *explicit graphs* integrated into an OO model

3 Spatial Data Models and Type Systems

3.1 What Has to Be Modeled from an Application Point of View?

3.2 Classification

3.3 Examples of Spatial Type Systems for Single Spatial Objects

3.4 Partitions

3.1 What Has to Be Modeled from an Application Point of View?

Spatial data types

- single, self contained objects: points, lines, regions
- spatially related collections of objects: partitions, networks

Spatial operations

- *spatial predicates* returning boolean values
  - *topological relationships*
    e.g., *equal*, *unequal*, *disjoint*, *adjacent* (*neighboring*), *intersect* (*overlap*), *meet* (*touch*), *inside* (*in*), *outside*, *covered_by*, *contains*
  - metric relationships
    e.g., *in_circle*, *in_window*
  - spatial order and strict order relationships
    e.g., *behind / in_front_of*, *above / below*, *over / under*, *inside / contains*
  - directional relationships
    e.g., *north / south*, *left / right*

Spatial Operations (*continued*)

- spatial operations returning numbers
  e.g., *area*, *perimeter*, *length*, *diameter*, *dist*, *mindist*, *maxdist*, *direction*, *components* (*cardinality*)
- spatial operations returning new spatial objects
  - object construction operations
    e.g., *union*, *intersection*, *difference*, *convex_hull*, *center*, *boundary* (*border*), *box*
  - object transformation operations
    e.g., *extend*, *rotate*, *translate*
- spatial operations on sets of spatially related objects
  - general operations
    e.g., *voronoi*, *closest*, *compose*, *decompose*
  - operations for partitions
    e.g., *overlay*, *superimposition*, *fusion*, *cover*, *windowing*, *clipping*
  - operations for networks
    e.g., *shortest_path*

3.2 Classification

Concrete Models

- point-based models, e.g.
  - Güting 1988 (geo-relational algebra)
  - Worboys & Bofakos 1993 (complex regions with holes)
  - Egenhofer & Herring 1990, Egenhofer & Franzosa 1991, ... (topological relationships)
  - Belussi, Bertino & Catania 1997, Grumbach, Rigaux & Segoufin 1998 (linear constraint approach)
- discrete models
  - Güting & Schneider 1995 (ROSE algebra)
  - Frank & Kuhn 1986, Egenhofer, Frank & Jackson 1989 (simplex-based model)

Abstract Models

- logic (pointless, axiomatic) models, e.g.
  - Cui, Cohn & Randell 1993, ... (spatial logic)

3.3 Examples of Spatial Type Systems for Single Spatial Objects

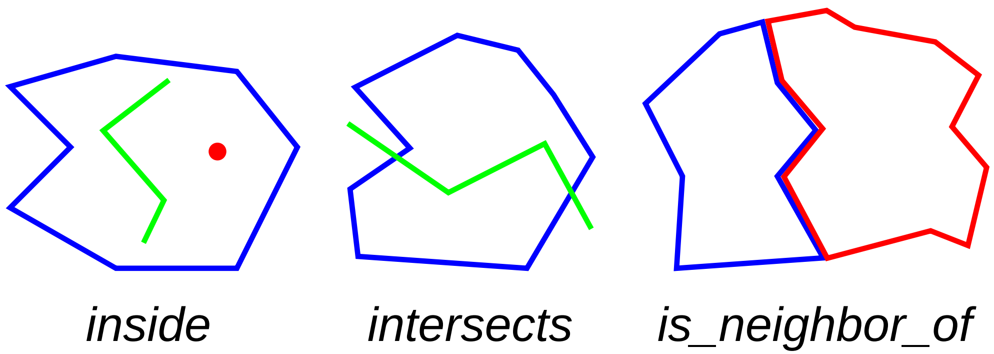
(1) Güting 1988 (*geo-relational algebra*) (based on *point set theory*)

- Relational algebra viewed as a *many-sorted algebra* (relations + atomic data types)
- Sorts: *rel*; *int*, *real*, *string*, *bool*; *point*, *line*, *pgon*, *area*
- example relation: *states*(*sname*: *string*, *extent*: *area*, *cpop*: *int*)
- a *point* value is a single point, a *line* value is a polyline, a *pgon* value is a polygon without holes
- special type *area* for modeling *partitions*

  $\rightarrow$ but: partition constraints are not maintained by the system but by the user
- generalizations: *reg* = {*pgon*, *area*}, *ext* = {*line*, *reg*}, *geo* = {*point*, *ext*}

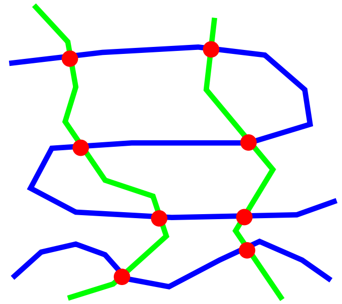
Geometric predicates

=, ≠:	$geo_i \times geo_i$	$\rightarrow bool$
inside:	$geo \times reg$	$\rightarrow bool$
intersects:	$ext \times ext$	$\rightarrow bool$
is_neighbor_of:	$area \times area$	$\rightarrow bool$

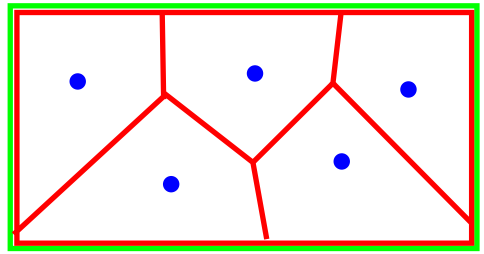


Geometric relation operations

intersection:	*line** $\times$ *line**	$\to$ *point**
	*line** $\times$ *reg**	$\to$ *line**
	*pgon** $\times$ *reg**	$\to$ *pgon**
overlay:	*area** $\times$ *area**	$\to$ *area**
vertices:	*ext**	$\to$ *point**
voronoi:	*point** $\times$ *reg*	$\to$ *area**
closest:	*point** $\times$ *point*	$\to$ *rel*



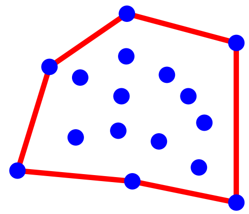
intersection



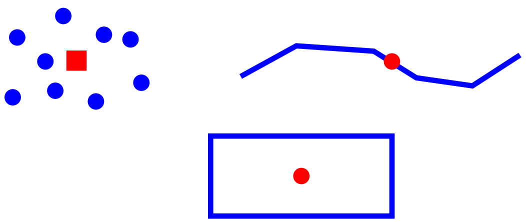
voronoi

Operations returning atomic geometric objects

convex_hull:	*point**	$\to$ *pgon*
center:	*point**	$\to$ *point*
	ext	$\to$ *point*



convex_hull



center

Operations returning numbers

dist:	*point* $\times$ *point*	$\rightarrow$ *real*
mindist, *maxdist*:	*geo* $\times$ *geo*	$\rightarrow$ *real*
diameter:	*point**	$\rightarrow$ *real*
length:	*line*	$\rightarrow$ *real*
perimeter, *area*:	*reg*	$\rightarrow$ *real*

Comparison to design criteria

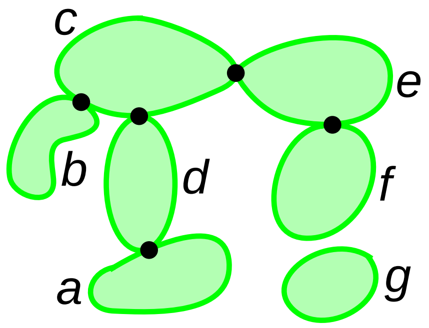
• general definition, closure properties	–
• formal definition	+
• finite precision arithmetic	–
• support for geometric consistency	(–)
• efficiency	+
• extensibility	+
• data model independence	–

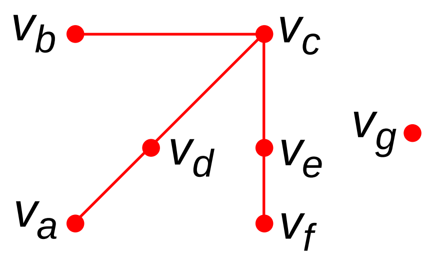
Remarks

- only simple polygons
  $\rightarrow$ no union, difference of polygons
- forming the intersection of two spatial objects must be embedded in a relation operation
- no numerically critical operations included
- simple data structures + algorithms

(2) Worboys & Bofakos 1993 (based on *point set topology*)

- complex spatial regions with holes and islands within holes to any finite level
- *atom*: subset of IR^2 that is topologically equivalent to a closed disc
- *base area*: aggregation of atoms whose structure is described by a *skeleton graph*

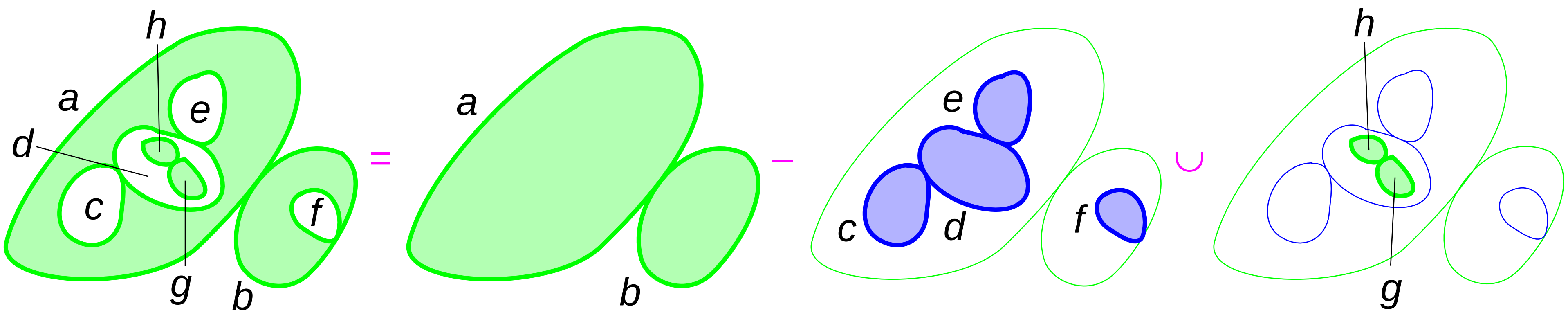


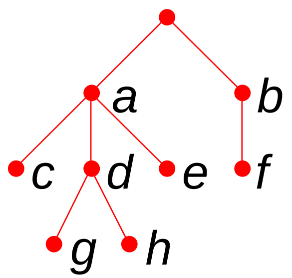


Constraints

- the intersection of any two distinct atoms is either empty or a singleton set
- composite object has no holes

- *generic area*: recursive construction of complex spatial regions





Constraints

- for each vertex their successors form a base area
- for each vertex $v \neq$ root and each successor w of v holds: (a) $w \subset v$, (b) the intersection of w and the boundary of v has finite cardinality

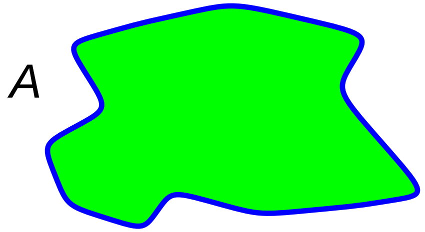
- operations: e.g., *equals*, *intersection*, *union*, *difference*, *boundary*, *adjacent*, *centroid*, *area*, *perimeter*, *cardinality*, *components*, *connected*

Comparison to design criteria

• general definition, closure properties	+	(only for regions)
• formal definition	+	
• finite precision arithmetic	–	
• support for geometric consistency	–	
• efficiency	?	
• extensibility	?	
• data model independence	+	

(3) Egenhofer & Herring 1990, Egenhofer & Franzosa 1991 (based on *point set topology*)

- goal: a "*complete*" collection of *topological relationships* between two spatial objects
- topological relationships are invariant under translation, rotation, and scaling
- originally: topol. relationships between two simple, connected regions without holes



boundary ≡ ∂A

interior ≡ A°

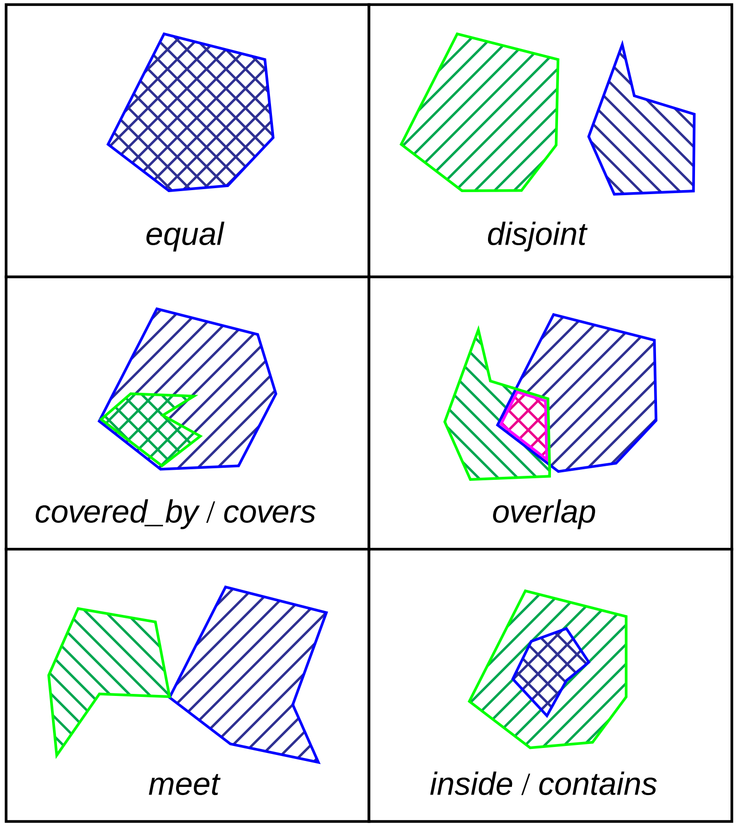
- *4-intersection model*: 4 intersection sets between boundaries and interiors of two objects

Extensions

- *9-intersection model* (Egenhofer 1991): consider also intersections of ∂A and A° with the exterior / complement A^- ($\rightarrow 9^2 = 81$ combinations, 8 are valid)
- include point and line features (Egenhofer & Herring 1992, de Hoop & van Oosterom 1992)

- *4-intersection model*

$\partial A \cap \partial B$	$\partial A \cap B°$	$A° \cap \partial B$	$A° \cap B°$	relationship name
$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	*A* and *B* are *disjoint*
$\varnothing$	$\varnothing$	$\varnothing$	$\neq\varnothing$	
$\varnothing$	$\varnothing$	$\neq\varnothing$	$\varnothing$	
$\varnothing$	$\varnothing$	$\neq\varnothing$	$\neq\varnothing$	*A contains B* / *B inside A*
$\varnothing$	$\neq\varnothing$	$\varnothing$	$\varnothing$	
$\varnothing$	$\neq\varnothing$	$\varnothing$	$\neq\varnothing$	*A inside B* / *B contains A*
$\varnothing$	$\neq\varnothing$	$\neq\varnothing$	$\varnothing$	
$\varnothing$	$\neq\varnothing$	$\neq\varnothing$	$\neq\varnothing$	
$\neq\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	*A* and *B meet*
$\neq\varnothing$	$\varnothing$	$\varnothing$	$\neq\varnothing$	*A* and *B* are *equal*
$\neq\varnothing$	$\varnothing$	$\neq\varnothing$	$\varnothing$	
$\neq\varnothing$	$\varnothing$	$\neq\varnothing$	$\neq\varnothing$	*A covers B* / *B covered_by A*
$\neq\varnothing$	$\neq\varnothing$	$\varnothing$	$\varnothing$	
$\neq\varnothing$	$\neq\varnothing$	$\varnothing$	$\neq\varnothing$	*A covered_by B* / *B covers A*
$\neq\varnothing$	$\neq\varnothing$	$\neq\varnothing$	$\varnothing$	
$\neq\varnothing$	$\neq\varnothing$	$\neq\varnothing$	$\neq\varnothing$	*A* and *B overlap*



4^2 = 16 combinations, 8 are valid

- *dimension extended method* (Clementini, Felice & van Oosterom 1993): consider dimension of the intersection (empty, 0D, 1D, 2D in 2D space)

  $\rightarrow$ 4^4 = 256 combinations for each relationship group (area / area, line / area, point / area, line / line, point / line, point / point), totally 52 are valid

  too many relationships to be remembered!

  alternative: five basic relationships *touch*, *in*, *cross*, *overlap*, *disjoint* plus three operators *b*, *f*, *t* to obtain boundaries

  one can prove:

  - 5 relationships are mutually exclusive
  - 5 relationships plus 3 boundary operators can distinguish all 52 configurations

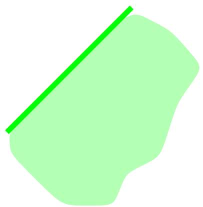
- consider regions with holes (Egenhofer, Clementini & Di Felice 1994)

- consider composite regions (Clementini, Di Felice & Califano 1995)

(4) Belussi, Bertino & Catania 1997, Grumbach, Rigaux & Segoufin 1998

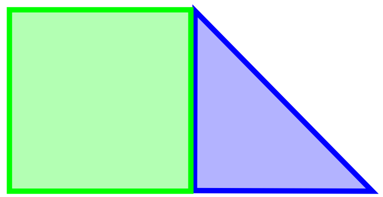
(based on a *linear constraint approach*)

- basic idea of the spatial constraint model: represent spatial objects finitely as infinite collections of points satisfying first-order formulas
- a convex polygon is the intersection of a finite set of half planes, i.e., a conjunction of the inequalities defining each half plane
- a non-convex polygon is the union (logical disjunction) of a finite set of convex polygons (disjunctive normal form (DNF))



$\{(x, y) \mid y - x \leq 0\}$

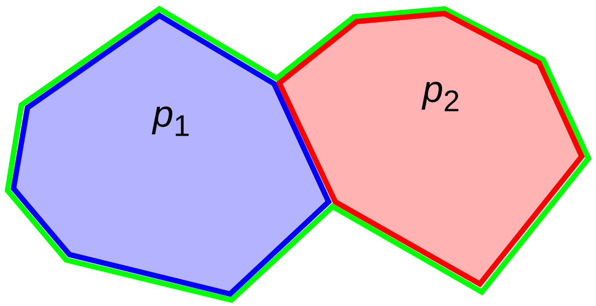
half plane representation



$\{(x, y) \mid x \leq 1 \wedge x \geq -1 \wedge y \leq 1 \wedge y \geq -1\}$

$\{(x, y) \mid x \geq -1 \wedge y \geq -1 \wedge x + y - 2 \leq 0\}$

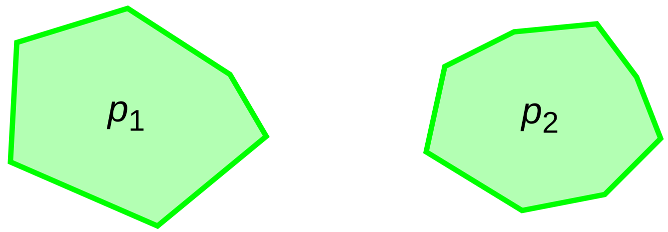
representation of two convex polygons



convexification of a polygon

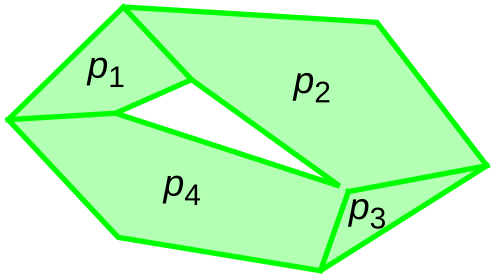
DNF repr.:

$p_1 \vee p_2$



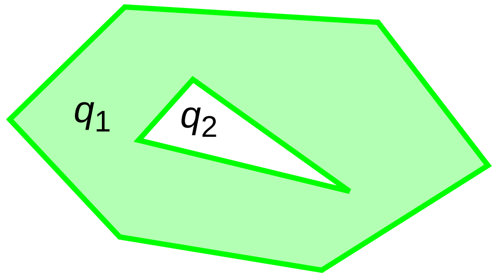
polygon with disjoint components

- CHNF (convex with holes normal form) is a generalization of DNF



DNF repr.:

$p_1 \vee p_2 \vee p_3 \vee p_4$



CHNF repr.:

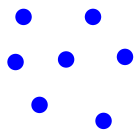
$q_1 - q_2$

Comparison to design criteria

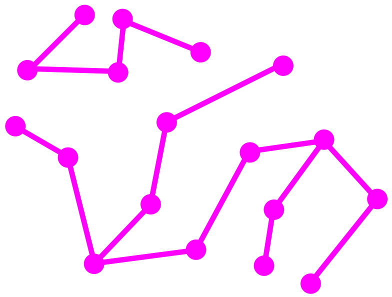
• general definition, closure properties	+
• formal definition	+
• finite precision arithmetic	(+)
• support for geometric consistency	+
• efficiency	?
• extensibility	?
• data model independence	–

(5) Güting & Schneider 1995 (ROSE algebra) (based on a *discrete geometric basis*)

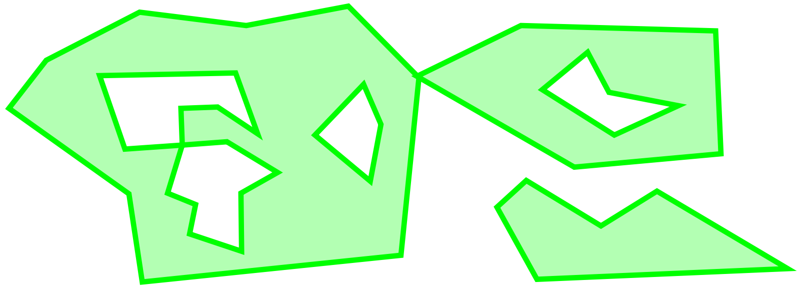
- *ROSE* = *RO*bust *S*patial *E*xtension
- system of realm-based spatial data types (*points*, *lines*, *regions*) whose objects are composed of realm elements (points and segments over a discrete geometric domain)



a *points* value



a *lines* value



a *regions* value

- ROSE algebra offers a comprehensive collection of precisely defined operations for manipulating such values, e.g. (let EXT = {*lines*, *regions*}, GEO = {point} $\cup$ EXT)

$\forall$ geo, geo_1, $geo_2 \in$ GEO, $\forall$ ext, ext_1, $ext_2 \in$ EXT, $\forall$ obj $\in$ OBJ

inside:	geo $\times$ *regions*	$\rightarrow$ *bool*
edge-/vertex-inside:	*regions* $\times$ *regions*	$\rightarrow$ *bool*
area-/edge-disjoint:	*regions* $\times$ *regions*	$\rightarrow$ *bool*

ROSE algebra also contains operations which are usually numerically critical, e.g.

on_border_of:	*points* $\times$ ext	$\rightarrow$ *bool*
border_in_common:	$ext_1 \times ext_2$	$\rightarrow$ *bool*

Closure properties are fulfilled for intersection, union, and difference due to general definition of spatial data types

intersection:	*points* $\times$ *points*	$\rightarrow$ *points*	
intersection:	*lines* $\times$ *lines*	$\rightarrow$ *points*	(no embedding into e.g.
intersection:	*regions* $\times$ *regions*	$\rightarrow$ *regions*	relation operations needed)
intersection:	*regions* $\times$ *lines*	$\rightarrow$ *lines*	
plus, *minus*:	geo $\times$ geo	$\rightarrow$ geo	

Spatial operations for manipulating sets of spatially related objects (i.e., database objects) defined by a general "*object model interface*"

sum:	*set*(obj) $\times$ (obj $\rightarrow$ geo)	$\rightarrow$ geo
closest:	*set*(obj) $\times$ (obj $\rightarrow geo_1) \times geo_2$	$\rightarrow$ *set*(obj)

(also operations for *overlay*, *fusion*, and *decompose*)

Other operations

vertices:	ext	$\rightarrow$ *points*
contour:	*regions*	$\rightarrow$ *lines*
interior:	*lines*	$\rightarrow$ *regions*
no_of_components:	geo	$\rightarrow$ *int*

Comparison to design criteria

• general definition, closure properties	+
• formal definition	+
• finite precision arithmetic	+
• support for geometric consistency	+
• efficiency	+
• extensibility	+
• data model independence	+

(6) Cui, Cohn & Randell 1993 (based on *logic*)

- pointless approach: *regions* are the basic entities, no points, no lines
- axiomatic approach to representing and reasoning about topological spatial data
- basic binary relation $C(x, y)$: "*x* connects with *y*"

  $\forall x\, C(x, y)$ reflexitivity of C
  $\forall xy\, [C(x, y) \rightarrow C(y, x)]$ symmetry of C

- axiomatic formulation of topological relationships

$DC(x, y)$	$\equiv_{def}$	$\neg\, C(x, y)$	"*x* is disconnected from *y*"
$P(x, y)$	$\equiv_{def}$	$\forall z\, [C(z, x) \rightarrow C(z, y)]$	"*x* is a part of *y*"
$x = y$	$\equiv_{def}$	$P(x, y) \wedge P(y, x)$	"*x* is identical with *y*"
$O(x, y)$	$\equiv_{def}$	$\exists z\, [P(z, x) \wedge P(z, y)]$	"*x* overlaps *y*"
$PO(x, y)$	$\equiv_{def}$	$O(x, y) \wedge \neg\, P(x, y) \wedge \neg\, P(y, x)$	"*x* partially overlaps *y*"
$EC(x, y)$	$\equiv_{def}$	$C(x, y) \wedge \neg\, O(x, y)$	"*x* is externally connected with *y*"
$TPP(x, y)$	$\equiv_{def}$	$PP(x, y) \wedge \exists z\, [EC(z, x) \wedge EC(z, y)]$	"*x* is a tangential proper part of *y*"
$NTTP(x, y)$	$\equiv_{def}$	$PP(x, y) \wedge \neg \exists z\, [EC(z, x) \wedge EC(z, y)]$	"*x* is a nontangential proper part of *y*"

- topological model to interpret the theory: C(*x*,*y*) holds when the topological closures of regions *x* and *y* share a common point
- similar results like Egenhofer *et al.*: 8 mutually exhaustive and pairwise disjoint relations

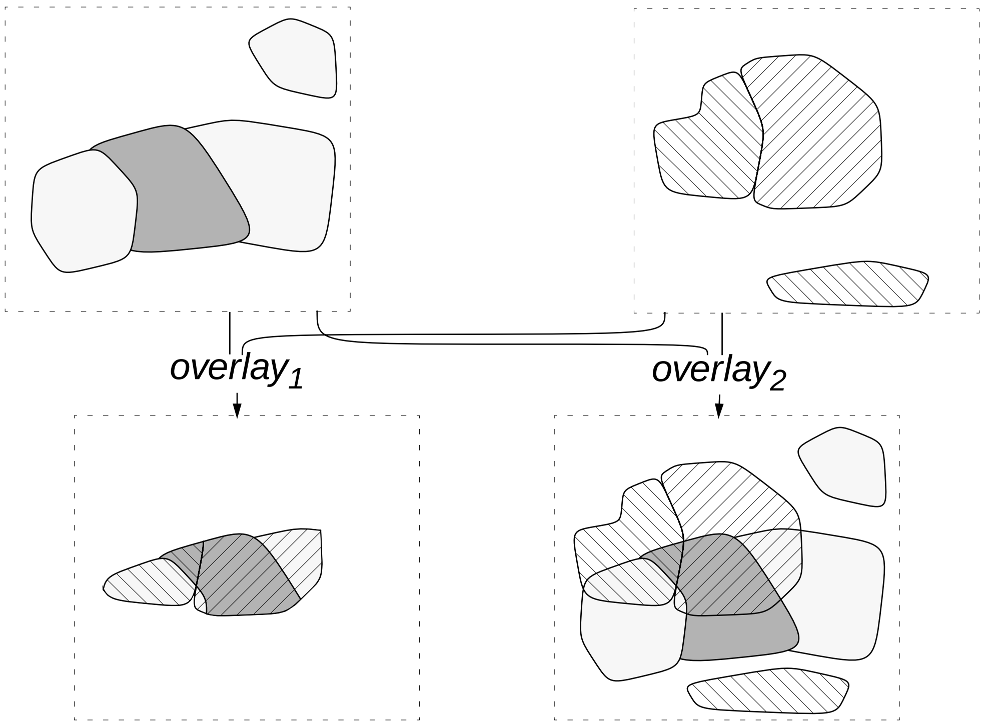
DC (*disjoint*)	TPP (*covered_by* / *covers*)	NTTP (*inside* / *contains*)
EC (*meet*)	TPP^{-1} (*covers* / *covered_by*)	$NTPP^{-1}$ (*contains* / *inside*)
PO (*overlap*)		
= (*equal*)		

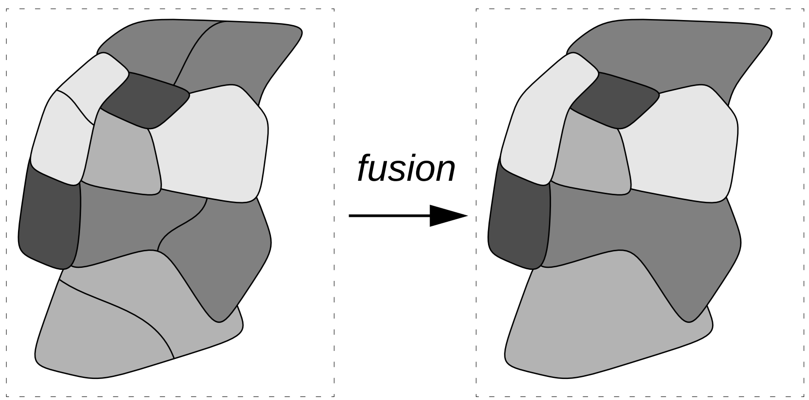
Comparison to design criteria

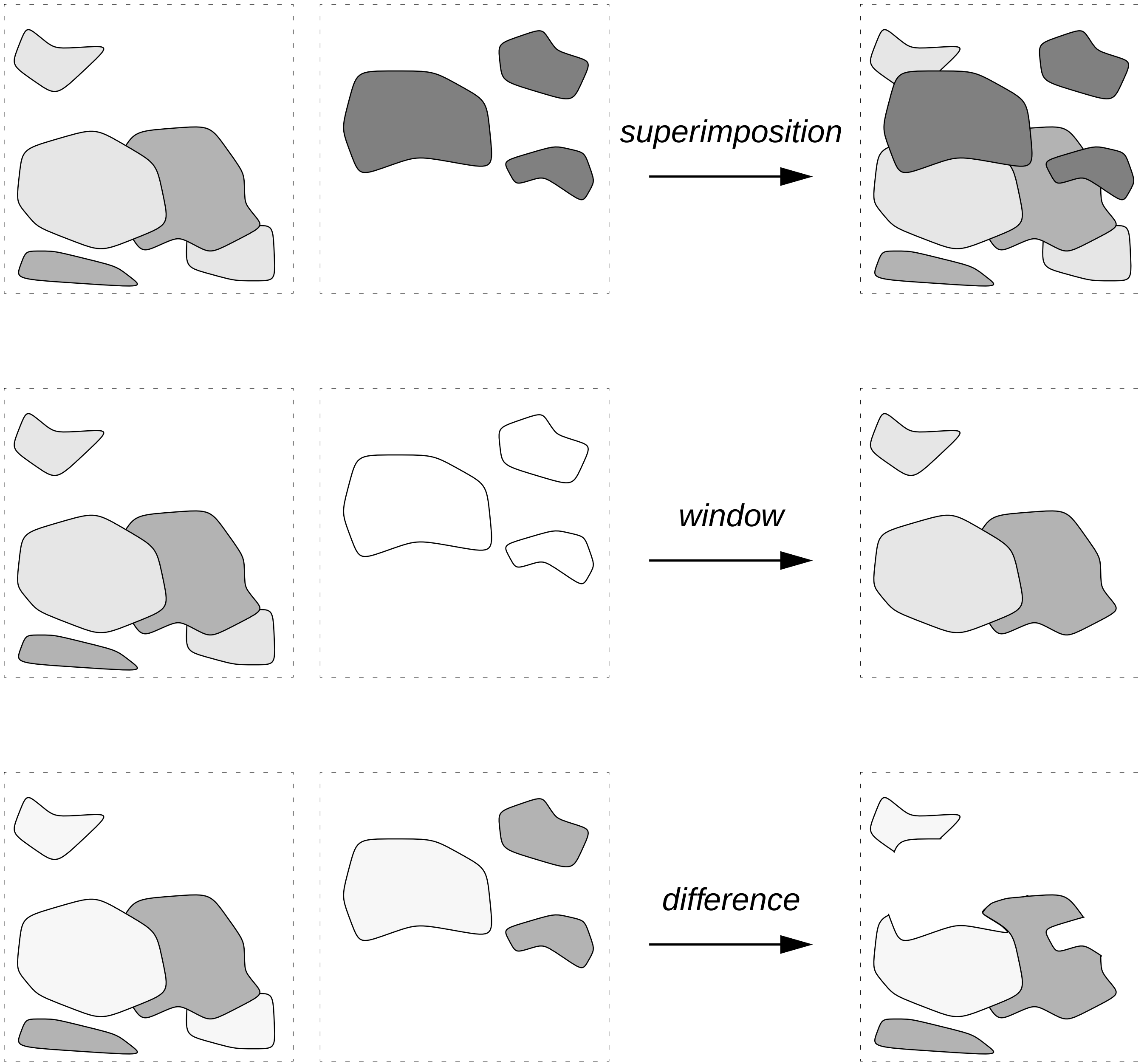
- general definition, closure properties –
- formal definition +
- finite precision arithmetic –
- support for geometric consistency –
- efficiency ?
- extensibility ?
- data model independence +

3.4 Partitions

- *partition*: subdivision of the plane into pairwise disjoint regions where each region is associated with an attribute having a simple or even complex structure
- partition implicitly models topological relationships
  - neighborhood of different regions which may have common boundaries
  - disjointedness of different regions (except for boundaries)
- application-specific operations






superimposition
window
difference

other operations: reclassify, cover, clipping

- Scholl & Voisard 1989
  - identification of application-specific operations on maps
  - complex object algebra extended by a data type for regions plus some operations on regions (union, intersection, difference)
  - a map is a set of tuples with a region attribute
  - elementary region: single polygon, region: set of polygons
  - problems: region type not closed under union operation, no control of partition constraints through the model, deeply data model dependent

- Erwig & Schneider 1997
  - formal definition of spatial partitions
  - basic idea: a partition is a mapping from IR^2 to some *label type*, i.e., regions of a partition are assigned single labels, adjacent regions have different labels in their interior, a boundary is assigned the pair of labels of both adjacent regions

– three powerful operations that are closed under partitions and that are sufficient to express all known (generalized) application-specific operations

intersection: compute the geometric intersection of all regions of two partitions and produce a new spatial partition; each resulting region is assigned the pair of labels of the original two intersecting regions; labels on the boundaries are derived correspondingly

relabel: change the labels of the regions of a partition either by renaming the label of each region or by mapping distinct labels of two or more regions to a new label; adjacent regions in the result partition are fused

refine: look with finer granularity on regions and reveal and enumerate the connected components of regions

- other approaches: e.g., Frank 1987, Huang, Svensson & Hauska 1992, Tomlin 1990

4 Formal Definition Methods

4.1 Why do We Need Formal Definitions?

4.2 Point Set Theory

4.3 Point Set Topology

4.4 Finite Set Theory

4.5 Other Formal Approaches

4.1 Why do We Need Formal Definitions?

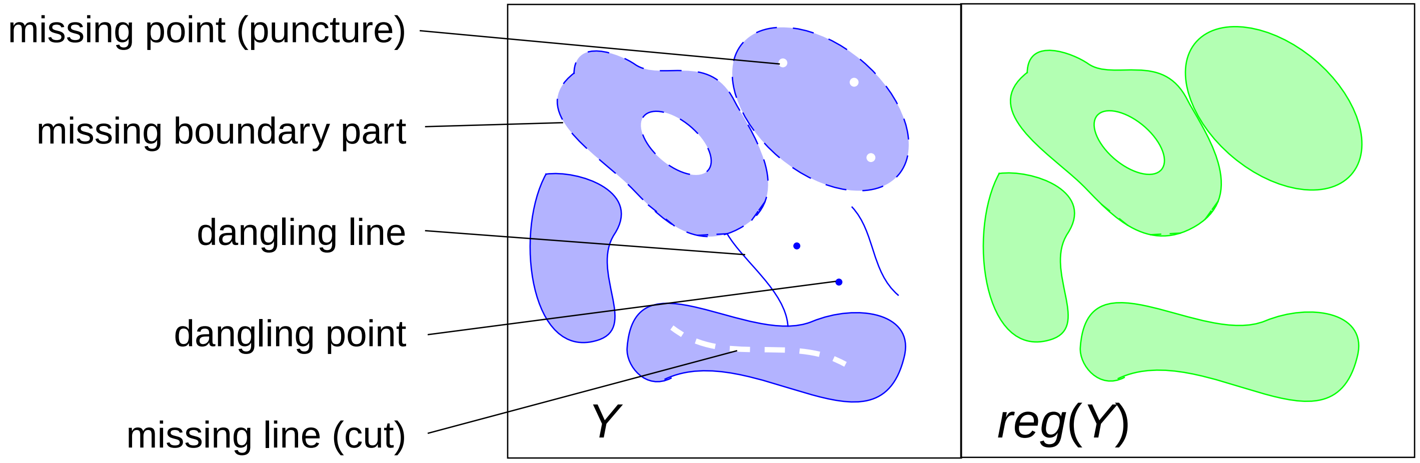
- better understanding of the complex semantics of spatial objects and operations at the designer's level
- formal definition of SDTs should be directly usable for a formal definition of corresponding spatial operations
- clarity and consistency at the user's level
- consideration of the finiteness of computers and the problems of numerical robustness and topological correctness
- a first step towards a standardization of spatial data types
- formal specification of SDTs for a possible realization at the implementation level

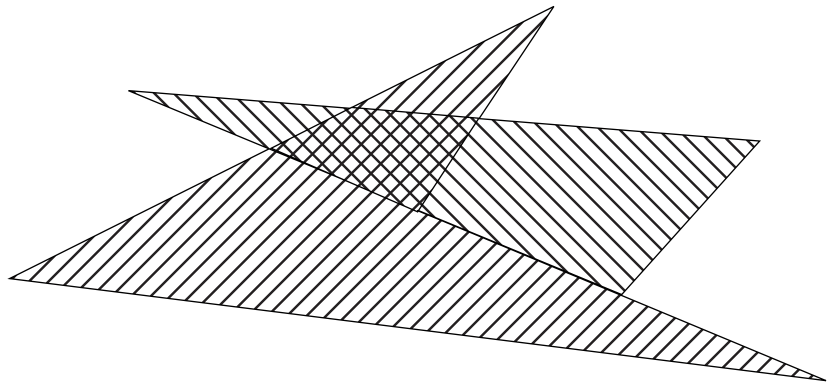
4.2 Point Set Theory

- basic assumption: space is composed of infinitely many *points* and contains a set of spatial objects

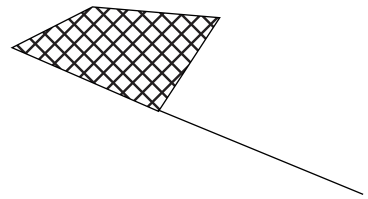
  → each spatial object can be regarded as the point set occupied by that object

- analytical geometry is used to represent structures like points, lines, regions, etc. by numbers and relations between these structures by equations
- use of set operations ∪, ∩, – for constructing new objects
- topological properties are deduced from analytical geometry by numerical computation
- two main problems
  - possible anomalies (shown first by Tilove 1980)

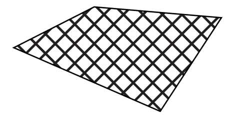




Two intersecting *Regions* objects



Conventional intersection



Regularized intersection

– ambiguities when defining topological relationships

$$x = y \ := \ points(x) = points(y)$$

$$x \ inside \ y \ := \ points(x) \subseteq points(y)$$

$$x \ intersects \ y \ := \ points(x) \cap points(y) \neq \emptyset$$

the definitions of = and *inside* are both covered by the definition of *intersects*

- e.g., Güting 1988, Pullar 1988

4.3 Point Set Topology

- ... has the same basic assumptions as point set theory and investigates topological structures of a point set

  boundary (∂Y), *interior* (Y°), *closure* ($\overline{Y}$), *exterior* (Y^-)

  $\overline{Y} = Y^\circ \cup \partial Y$ $\quad Y^\circ \cap \partial Y = \varnothing$ $\quad Y^- \cap \partial Y = \varnothing$ $\quad Y^\circ \cap Y^- = \varnothing$ $\quad \mathrm{IR}^2 = Y^\circ \cup \partial Y \cup Y^-$

- ... investigates properties that are independent of an underlying distance or coordinate measure (metric) and that are preserved under continuous topological transformations
- *regularization* of point sets to avoid anomalies which leads to spatial objects as *regular closed sets*

  Y is a *regular closed* set if $Y = \overline{Y^\circ}$ $\qquad reg(Y) := \overline{Y^\circ}$

  effect of *interior*: elimination of dangling points, dangling lines and boundary parts

  effect of *closure*: elimination of cuts and punctures by appropriately adding points plus adding boundary points

- geometric operations are equated with regular set operations (A, B regular closed sets)

  $A \cup_r B := reg(A \cup B)$ $\quad A \cap_r B := reg(A \cap B)$ $\quad A -_r B := reg(A - B)$ $\quad \neg_r A := reg(\neg A)$

- e.g., Worboys & Bofakos 1993, Egenhofer & Herring 1990, Egenhofer & Franzosa 1991

4.4 Finite Set Theory

Example: Definition layers of *Realms* and *ROSE Algebra* (Güting & Schneider 1993, 1995)

ROSE Algebra Operations	Objects: *points*, *lines*, *regions* Operations: **=**, **≠**, **inside**, **edge_inside**, **vertex_inside**, **area_disjoint**, **edge_disjoint**, **disjoint**, **intersects**, **meets**, **adjacent**, **encloses**, **on_border_of**, **border_in_common**, **intersection**, **plus**, **minus**, **common_border**, **vertices**, **contour**, **interior**, **count**, **dist**, **diameter**, **length**, **area**, **perimeter**, **sum**, **closest**, **decompose**, **overlay**, **fusion**
Spatial Data Types and Spatial Algebra Primitives	Objects: *points*, *lines*, *regions* Operations: ***union***, ***intersection***, ***difference***, (***area-***)***inside***, ***edge-inside***, ***vertex-inside***, ***area-disjoint***, ***edge-disjoint***, (***vertex-***)***disjoint***, ***meet***, ***adjacent***, ***intersect***, ***encloses***, ***on_border_of***, ***border_in_common***
Realms, Realm-Based Structures, and Realm-Based Primitives	Objects: *R*-point, *R*-segment; *R*-cycle, *R*-face, *R*-unit, *R*-block Operations: *on*, *in*, *out*, (*area-*)*inside*, *edge-inside*, *vertex-inside*, *area-disjoint*, *edge-disjoint*, (*vertex-*)*disjoint*, *adjacent*, *meet*, *encloses*, *intersect*, *dist*, *area*
Robust Geometric Primitives	Objects: *N*-point, *N*-segment Operations: =, ≠, meet, overlap, intersect, disjoint, on, in, touches, intersection, parallel, aligned
Integer Arithmetic	Objects: integers in the range $[-2n^3, 2n^3]$ (*n* integer grid size) Operations: +, -, ∗, **div**, **mod**, =, ≠, <, ≤, ≥, >

Example: definition of a *region* object

$N := \{0, ..., n\}$, n finite and representable

$P_N := N \times N$ — *N*-points

$S_N := P_N \times P_N$ — *N*-segments

$P \subseteq P_N$ — *R*-points

$S \subseteq S_N$ — *R*-segments

Realm properties

(i) $\forall\, s \in S : s = (p, q) \Rightarrow p \in P \wedge q \in P$

(ii) $\forall\, p \in P\ \forall\, s \in S : \neg\, (p\ \underline{\text{in}}\ s)$

(iii) $\forall\, s, t \in S, s \neq t : \neg\, (s \text{ and } t\ \underline{\text{intersect}})$
$\wedge \neg\, (s \text{ and } t\ \underline{\text{overlap}})$

An *R-cycle* c is a set of *R*-segments $S(c) = \{s_0, ..., s_{m-1}\}$, such that

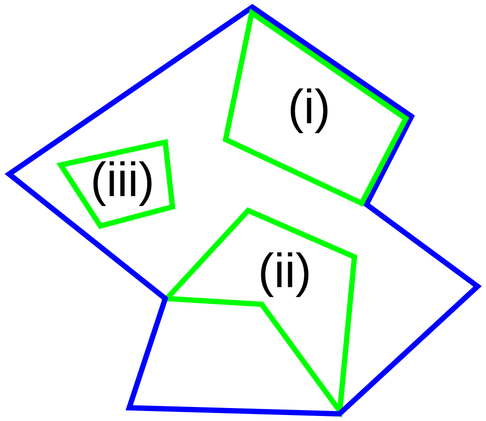
(i) $\forall\, i \in \{0, ..., m–1\} : s_i\ \underline{\text{meets}}\ s_{(i+1) \bmod m}$

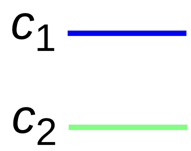
(ii) $\forall\, i \in \{0, ..., m–1\} : deg(s_i) = 2$

c_2 is

- (*area-*)*inside* (i, ii, iii)
- *edge-inside* (ii, iii)
- *vertex-inside* (iii)

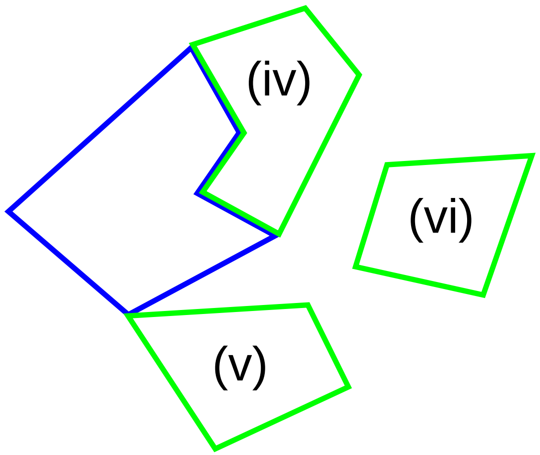
c_1.





c_1 and c_2 are

- *area-disjoint* (iv, v, vi)
- *edge-disjoint* (v, vi)
- (*vertex-*)*disjoint* (vi)



c_1 (*area-*)*inside* c_2 :⇔ $P(c_1) \subseteq P(c_2)$

c_1 *edge-inside* c_2 :⇔ c_1 *area-inside* c_2 $\wedge\, S(c_1) \cap S(c_2) = \varnothing$

c_1 *vertex-inside* c_2 :⇔ c_1 *edge-inside* c_2 $\wedge\, P_{on}(c_1) \cap P_{on}(c_2) = \varnothing$

c_1 and c_2 are *area-disjoint* :⇔ $P_{in}(c_1) \cap P(c_2) = \varnothing \wedge P_{in}(c_2) \cap P(c_1) = \varnothing$

c_1 and c_2 are *edge-disjoint* :⇔ c_1 and c_2 are *area-disjoint* $\wedge S(c_1) \cap S(c_2) = \varnothing$

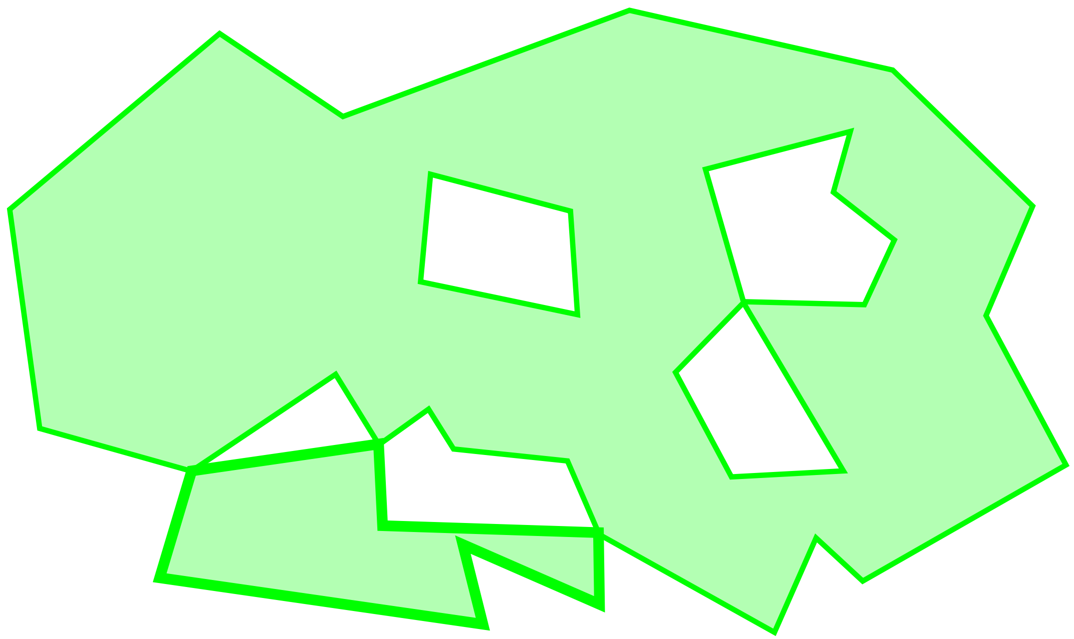
c_1 and c_2 are (*vertex-*)*disjoint* :⇔ c_1 and c_2 are *edge-disjoint* $\wedge P_{on}(c_1) \cap P_{on}(c_2) = \varnothing$

An *R-face* f is a pair (c, H) where c is an R-cycle and $H = \{h_1, ..., h_m\}$ is a (possibly empty) set of R-cycles such that:

(i) $\forall\, i \in \{1, ..., m\} : h_i$ <u>*edge-inside*</u> c

(ii) $\forall\, i, j \in \{1, ..., m\}, i \neq j : h_i$ and h_j are <u>*edge-disjoint*</u>

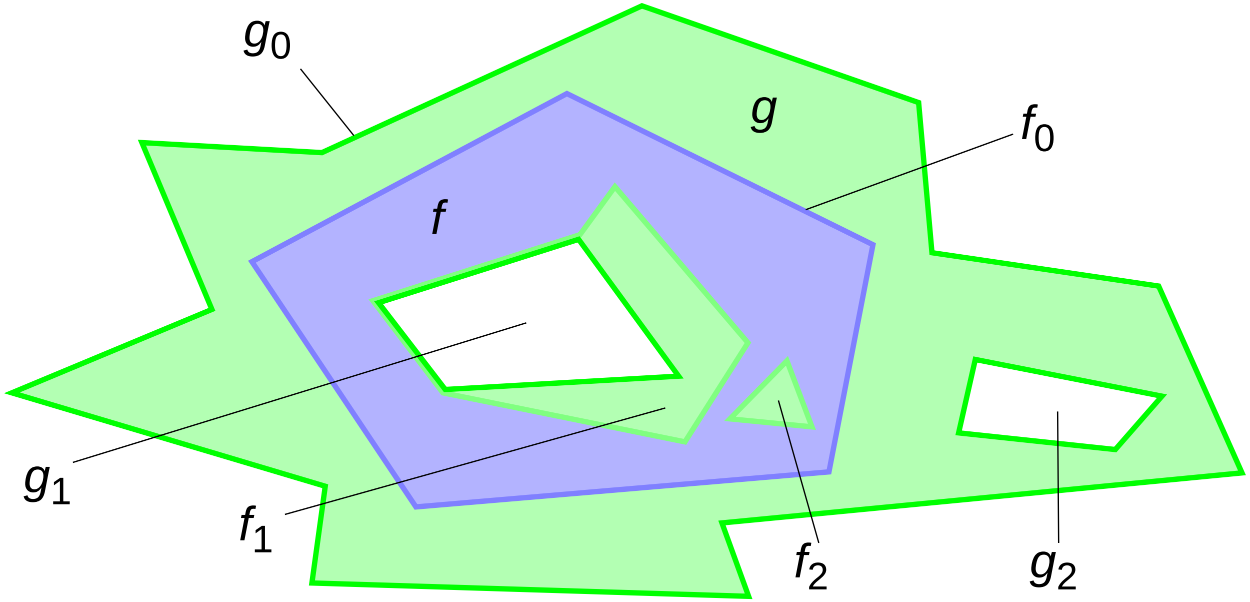
(iii) "no other cycle can be formed from the segments of f"



Let $f = (f_0, F)$ and $g = (g_0, G)$ be two *R*-faces. Then

f (area-)*inside* *g* $:\Leftrightarrow$ f_0 *area-inside* g_0

$\wedge\ \forall\ g' \in G : (g'$ *area-disjoint* $f_0 \vee \exists\ f' \in F : g'$ *area-inside* $f')$



A *regions* value *F* is a set of edge-disjoint *R*-faces.

Let *F*, *G* be two *regions* values.

F (area-)*inside* *G* $:\Leftrightarrow$ $\forall\ f \in F\ \exists\ g \in G : f$ *area-inside* g

4.5 Other Formal Approaches

- Algebraic Topology
  - ... describes the structure of a (topological) space by an algebraic system
  - ... is not based on general set theory
  - ... uses properties that are invariant under topological transformations
  - topological properties are explicitly recorded (*simplices*, *simplicial complexes*)
  - concepts of *boundary* and *interior* (different from point set topology)
  - e.g., Frank & Kuhn 1986, Egenhofer, Frank & Jackson 1989, Egenhofer 1989
- Order Theory, Lattice Theory
  - ... allows the comparison of two or more elements of a set and can be used to answer queries of inclusion and containment
  - *strict order* for modeling a hierarchy of elements of a set: subdivision of space into regions (e.g., political subdivisions), perspectives (e.g., left / right, front of / behind)
  - *partial order* for combining several hierarchies of space: e.g., relationship between districts and cultivation areas
  - e.g., Kainz 1988, 1989, 1990, Kainz, Egenhofer & Greasley1993, Saalfeld 85

- Constraint Approach
  - first order logic with a point set interpretation
  - *constraints* are linear equations and inequalities of the form $\sum_{i=1}^{p} a_i x_i \Theta a_0$
  - e.g., Belussi, Bertino & Catania 1997, Grumbach, Rigaux & Segoufin 1998

- Spatial Logic
  - pointless approach, regions as basic entities
  - Clarke's calculus of individuals based on connection, Allen's interval logic
  - e.g., Randell, Cui & Cohn 1992, Cui, Cohn & Randell 1993

5 Tools for Implementing SDTs: Data Structures and Algorithms

5.1 Representing SDT Values

5.2 Implementing Atomic SDT Operations

5.1 Representing SDT Values

Goals: Implementation of a spatial type system (spatial algebra) so that it can be integrated into a DBMS (query processing, storage management, user interface, etc.), fulfillment of the design criteria:

- representations for the types ($\rightarrow$ data structures)
- algorithms for the operations ($\rightarrow$ algorithms)

DBMS view of SDT values

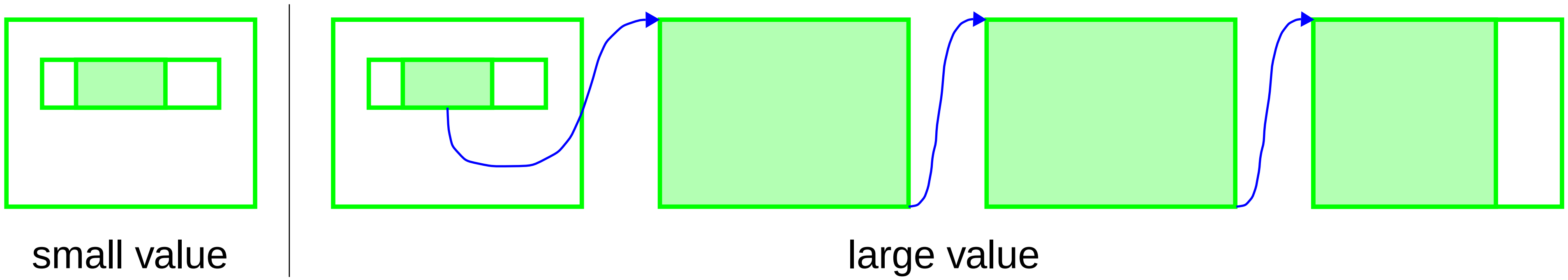
- treatment like values of other types w.r.t. generic operations (access, use in schemas, bulk loading, data exchange, user interface)
- values of varying and possibly large size
- persistent storage on disk in one or more pages
- efficient loading into main memory (value of a pointer variable there)

Algebra view of SDT values

- some (possibly) complex data structure
- use as a value of a programming language type
- support of computational geometry algorithms
- no special support for each single operation (no most efficient algorithm, no sophisticated data structure): reconcile the various requirements of different algorithms within a data structure for each type

Support of the DBMS view

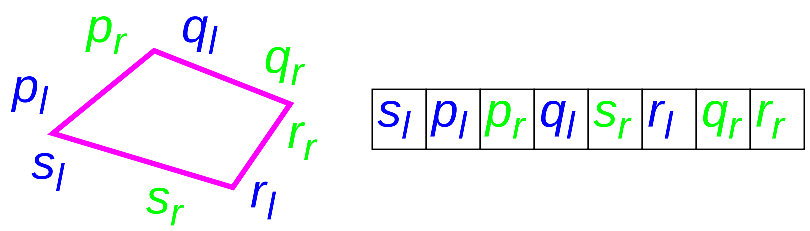
- no use of pointer data structures in main memory, use of a page-oriented data structure accommodating with DBMS support for large attribute values or long fields



- separation of an SDT value into an *info part of small, fixed size* and the *exact geometry part of possibly large, varying size*

Support of the algebra view

- representation contains *approximations* (e.g., bounding box) in the info part
- representation contains stored values of unary functions (e.g., area, perimeter, length, number of components, etc.) in the info part
- representation contains *plane sweep sequence* in the geometric part



5.2 Implementing Atomic SDT Operations

General remarks

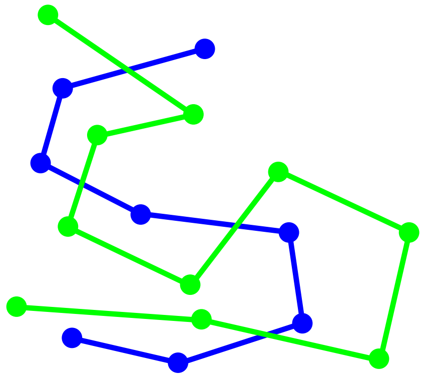
- in general: use efficient algorithms from Computational Geometry
- single steps
  - check approximations (filter condition)
  - look up stored function values
  - use plane sweep
- e.g., Güting, de Ridder & Schneider 1995, Schneider 1997, Chan & Ng 1997

Special case: implementation of realm-based SDTs

- all spatial objects have been acquainted with each other when they were entered into the realm ("*realmification*")

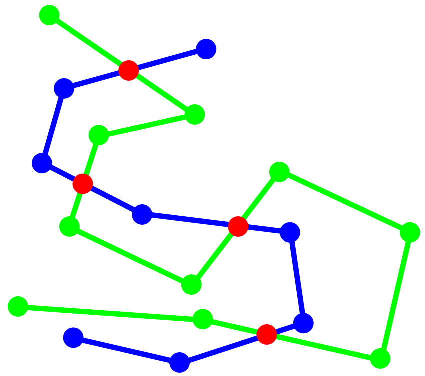
  → no new intersection points have to be computed, all are known in advance and occur in both objects
- often a parallel scan of two SDT values is sufficient where otherwise a plane sweep has to be used

- example: *intersection*: *lines* × *lines* → *points*



“classical” plane sweep needed (complex)

O(*n* log *n* + *k*)



parallel scan on two sequences of halfsegments (simple)

O(*n* + *k*)

- plane sweep is also simpler than usual: only static sweep-event structure is needed, no preceding sorting phase

- Güting, de Ridder & Schneider 1995

6 Other Interesting Issues and Research Trends

6.1 Other Interesting Issues not Covered in this Tutorial

6.2 Current Research Trends

6.1 Other Interesting Issues not Covered in this Tutorial

- data types and operations for image database systems and raster data management
- *multi-scale modeling* / *cartographic generalization*

  e.g., Puppo & Dettori 1995, Rigaux & Scholl 1995
- *three-dimensional spatial data modeling*

  e.g., Pigot 1992, Oosterom, Vertegaal, Hekken & Vijlbrief 1994
- *spatially embedded graphs* (*networks*)

  e.g., Erwig 1994, Erwig & Güting 1994, Güting 1991, Güting 1994

6.2 Current Research Trends

- combination of space and time

  spatio-temporal databases, *moving objects databases*

  e.g., Worboys 1994, Sistla, Wolfson, Chamberlain & Dao 1997, Erwig, Güting, Schneider & Vazirgiannis 1999, Erwig & Schneider 1999

  $\rightarrow$ European research project CHOROCHRONOS

- combination of space and uncertainty / vagueness

  spatial objects with imprecise / indeterminate / broad boundaries, *vague objects*, *fuzzy objects*

  e.g., Clementini & Di Felice 1996, Cohn & Gotts 1996, Erwig & Schneider 1997, Schneider 1999

References

Alves D., A Data Model for Geographic Information Systems. *4th Int. Symp. on Spatial Data Handling*, 2, 879-887, 1990.

Aref W.G. & H. Samet, Extending a DBMS with Spatial Operations. *2nd Int. Symp. on Advances in Spatial Databases* (*SSD'91*), Springer-Verlag, LNCS 525, 299-318, 1991.

Belussi A., E. Bertino & B. Catania, Manipulating Spatial Data in Constraint Databases. *5th Int. Symp. on Advances in Spatial Databases* (*SSD'97*), LNCS 1262, 115-141, 1997.

Bennis K., B. David, I. Quilio & Y. Viemont, GeoTropics Database Support Alternatives for Geographic Applications. *4th Int. Symp. on Spatial Data Handling*, 599-610, 1990.

Bentley J.L. & T. Ottmann, Algorithms for Reporting and Counting Geometric Intersections. *IEEE Transactions on Computers*, C-28, 643-647, 1979.

Birkhoff G., *Lattice Theory*. American Mathematical Society Colloquium Publications, 25, 1967.

Burrough P.A., Natural Objects with Indeterminate Boundaries. *Geographic Objects with Indeterminate Boundaries*, GISDATA Series 2, Taylor & Francis, 3-28, 1996.

Burton F.W., V.J. Kollias & J.G. Kollias, Consistency in Point-in-Polygon Tests. *The Computer Journal*, 27(4), 375-376, 1984.

Burton W., Logical and Physical Data Types in Geographic Information Systems. *Geo-Processing*, 1, 167-181, 1979.

Chan E.P.F. & R. Zhu, QL/G - A Query Language for Geometric Data Bases. 1st Int. Conf. on GIS in Urban and Environmental Planning, 271-286, 1996.

Clementini E. & P. di Felice, An Algebraic Model for Spatial Objects with Indeterminate Boundaries. *Geographic Objects with Indeterminate Boundaries*, GISDATA Series, 2, Taylor & Francis, 153-169, 1996.

Clementini E., P. di Felice & G. Califano, Composite Regions in Topological Queries, *Information Systems*, 20, 579-594, 1995.

Clementini E., P. di Felice & P. van Oosterom, A Small Set of Formal Topological Relationships Suitable for End-User Interaction. *3rd Int. Symp. on Advances in Spatial Databases* (*SSD'93*), LNCS 692, 277-295, 1993.

Claire R.W. & S.C. Guptill, Spatial Operators for Selected Data Structures. *Auto-Carto 5*, 189-200, 1982.

Cohn A.G. & N.M. Gotts, The 'Egg-Yolk' Representation of Regions with Indeterminate Boundaries. *Geographic Objects with Indeterminate Boundaries*, GISDATA Series, 2, Taylor & Francis, 171-187, 1996.

Cox R.J., B.K. Aldred & D.W. Rhind, A Relational Data Base System and a Proposal for a Geographical Data Type. *Geo-Processing*, 1, 217-229, 1980.

Croom F.H., *Basic Concepts of Algebraic Topology*. Springer-Verlag, 1978.

Cui Z., A.G. Cohn & D.A. Randell, Qualitative and Topological Relationships in Spatial Databases. *3rd Int. Symp. on Advances in Spatial Databases* (*SSD'93*), LNCS 692, 296-315, 1993.

Dayal U., F. Manola, A. Buchmann, U. Chakravarthy, D. Goldhirsch, S. Heiler, J. Orenstein & A. Rosenthal, Simplifying Complex Objects: The PROBE Approach to Modelling and Querying Them. *BTW* (Datenbanksysteme in Büro, Technik und Wissenschaft), Springer-Verlag, 17-37, 1987.

Dobkin D. & D. Silver, Applied Computational Geometry: Towards Robust Solutions of Basic Problems. *Journal of Computer and System Sciences*, 40, 70-87, 1990.

Edelsbrunner H. & E.P. Mücke, Simulation of Simplicity: A Technique to Cope with Degenerate Cases in Geometric Algorithms. *ACM Symp. on Computational Geometry*, 118-133, 1988.

Egenhofer M.J., A Formal Definition of Binary Topological Relationships. *3rd Int. Conf. on Foundations of Data Organization and Algorithms*, LNCS 367, 457-472, 1989.

Egenhofer M.J., Reasoning about Binary Topological Relations. *2nd Symp. on Advances in Spatial Databases* (*SSD'91*), LNCS 525, 143-160, 1991.

Egenhofer M.J., Categorizing Topological Relationships - Quantitive Refinements of Qualitative Spatial Information. Technical Report, Department of Surveying Engineering, University of Maine, 1991.

Egenhofer M.J., E. Clementini & P. di Felice, Topological Relations between Regions with Holes. *Int. Journal of Geographical Information Systems*, 8(2), 129-142, 1994.

Egenhofer M.J., A. Frank & J.P. Jackson, A Topological Data Model for Spatial Databases. *1st Int. Symp. on the Design and Implementation of Large Spatial Databases* (*SSD'89*), LNCS 409, 271-286, 1989.

Egenhofer M.J. & R.D. Franzosa, Point-Set Topological Spatial Relations. *Int. Journal of Geographical Information Systems*, 5(2), 161-174, 1991.

Egenhofer M.J. & R.D. Franzosa, On the Equivalence of Topological Relations. *Int. Journal of Geographical Information Systems*, 9(2), 133-152, 1995.

Egenhofer M.J. & J. Herring, A Mathematical Framework for the Definition of Topological Relationships. *4th Int. Symp. on Spatial Data Handling*, 803-813, 1990.

Egenhofer M.J. & J. Herring, Categorizing Binary Topological Relationships between Regions, Lines, and Points in Geographic Databases. Technical Report, Department of Surveying Engineering, University of Maine, 1992.

Erwig M., *Graphs in Spatial Databases*. Doctoral Thesis, FernUniversität Hagen, 1994.

Erwig M. & R.H. Güting, Explicit Graphs in a Functional Model for Spatial Databases. *IEEE Transactions on Knowledge and Data Engineering*, 6(5), 787-804, 1994.

Erwig M., R.H. Güting, M. Schneider & M. Vazirgiannis, Abstract and Discrete Modeling of Spatio-Temporal Data Types. *ACM Symp. on Geographic Information Systems* (*ACM GIS*), 131-136, 1998.

Erwig M. & M. Schneider, Vague Regions. *5th Int. Symp. on Advances in Spatial Databases* (*SSD'97*), LNCS 1262, Springer-Verlag, 298-320, 1997.

Erwig M. & M. Schneider, Partition & Conquer. *3rd Int. Conf. on Spatial Information Theory* (*COSIT*), LNCS 1329, 389-408, 1997.

Erwig M. & M. Schneider, Developments in Spatio-Temporal Query Languages. *IEEE Int. Workshop on Spatio-Temporal Data Models and Query Languages*, 1999, to appear.

Erwig M. & M. Schneider, The Honeycomb Model of Spatio-Temporal Partitions. *Int. Workshop on Spatio-Temporal Database Management*, 1999, to appear.

Fernandes A.A.A., A. Dinn, N.W. Paton & M.H. Williams, Extending a Deductive Object-Oriented Database System with Spatial Data Handling Facilities. *Information and Software Technology*, 41, 483-497, 1999.

Forrest A.R., Computational Geometry in Practice. *Fundamental Algorithms for Computer Graphics*, Springer-Verlag, 707-723, 1985.

Frank A., MAPQUERY: Data Base Query Language for Retrieval of Geometric Data and their Graphical Representation. *Computer Graphics*, 16, 199-207, 1982.

Frank A., Overlay Processing in Spatial Information Systems. *8th Int. Symp. on Computer-Assisted Cartography, AUTOCARTO 8*, 16-31, 1987.

Frank A. & W. Kuhn, Cell Graphs: A Provable Correct Method for the Storage of Geometry. *3rd Int. Symp. on Spatial Data Handling*, 411-436, 1986.

Freeman J., The Modelling of Spatial Relations. *Computer Graphics and Image Processing*, 4, 156-171, 1975.

Gaal S., *Point Set Topology*. Academic Press, 1964.

Gardarin G., J.P. Cheiney, G. Kiernan, D. Pastre & H. Stora, Managing Complex Objects in an Extensible Relational DBMS. *15th Int. Conf. on Very Large Databases*, 33-44, 1989.

Gargano M., E. Nardelli & M. Talamo, Abstract Data Types for the Logical Modeling of Complex Data. *Information Systems*, 16(5), 1991.

Greene D. & F. Yao, Finite-Resolution Computational Geometry. *27th IEEE Symp. on Foundations of Computer Science*, 143-152, 1986.

Grumbach S., P. Rigaux, M. Scholl & L. Segoufin, Dedale: A Spatial Constraint Database. *Int. Workshop on Database Programming Languages*, 1997.

Erwig M., R.H. Güting, M. Schneider & M. Vazirgiannis, Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases. *GeoInformatica*, 3(3), 1999, to appear.

Grumbach S., P. Rigaux & L. Segoufin, The DEDALE System for Complex Spatial Queries. *ACM SIGMOD Conf.*, 213-224, 1998.

Güting R.H., Geo-Relational Algebra: A Model and Query Language for Geometric Database Systems. *Int. Conf. on Extending Database Technology*, 506-527, 1988.

Güting R.H., Modeling Non-Standard Database Systems by Many-Sorted Algebras. Fachbereich Informatik, Universität Dortmund, Report 255, 1988.

Güting R.H., Gral: An Extensible Relational Database System for Geometric Applications. *15th Int. Conf. on Very Large Databases* (*VLDB*), 33-44, 1989.

Güting R.H., Extending a Spatial Database System by Graphs and Object Class Hierarchies. *Int. Workshop on Database Management Systems for Geographical Applications*, 103-124, 1991.

Güting R.H., GraphDB: A Data Model and Query Language for Graphs in Databases. *20th Int. Conf. on Very Large Databases* (*VLDB*), 297-308, 1994.

Güting R.H., An Introduction to Spatial Database Systems. *VLDB Journal (Special Issue on Spatial Database Systems)*, 3(4), 357-399, 1994.

Güting R.H., T. de Ridder & M. Schneider, Implementation of the ROSE Algebra: Efficient Algorithms for Realm-Based Spatial Data Types. *4th Int. Symp. on Advances in Spatial Databases* (*SSD'95*), LNCS 951, 216-239, 1995.

Güting R.H. & M. Schneider, Realms: A Foundation for Spatial Data Types in Database Systems. *3rd Int. Symp. on Advances in Spatial Databases*, LNCS 692, 14-35, 1993.

Güting R.H. & M. Schneider, Realm-Based Spatial Data Types: The ROSE Algebra, *VLDB Journal*, 4, 100-143, 1995.

Guibas L., D. Salesin & J. Stolfi, Epsilon-Geometry: Building Robust Algorithms from Imprecise Computations. *SIAM Conf. on Geometric Design*, 208-217, 1989.

Guttman A., R-Trees: A Dynamic Index Structure for Spatial Searching. *ACM SIGMOD Conf.*, 47-57, 1984.

Herring J.R., The Definition and Development of a Topological Spatial Data System. *Photogrammetry and Land Information Systems*, 57-70, 1990.

Herring J.R., The Mathematical Modeling of Spatial and Non-Spatial Information in Geographic Information Systems. *Cognitive and Linguistic Aspects of Space*, 1991.

Hoffmann C.M., J.E. Hopcroft & M.S. Karasick, Towards Implementing Robust Geometric Computations. *ACM Symp. on Computational Geometry*, 106-117, 1988.

Huang Z., P. Svensson & H. Hauska, Solving Spatial Analysis Problems with GeoSAL, A Spatial Query Language. *6th Int. Working Conf. on Scientific and Statistical Database Management*, 1992.

Kainz W., Application of Lattice Theory in Geography. *3rd Int. Symp. on Spatial Data Handling*, 135-142, 1988.

Kainz W., Order, Topology, and Metric in GIS. *ASPRS/ACSM Annual Convention*, 4, 154-160, 1989.

Kainz W., Spatial Relationships - Topology Versus Order. *4th Int. Symp. on Spatial Data Handling*, 814-819, 1990.

Kainz W., M.J. Egenhofer & I. Greasley, Modelling Spatial Relations and Operations with Partially Ordered Sets. *Int. Journal of Geographical Information Systems*, 7(3), 215-229, 1993.

Kemper A. & M. Wallrath, An Analysis of Geometric Modeling in Database Systems. *ACM Computing Surveys*, 19(1), 47-91, 1987.

Larue T., D. Pastre & Y. Viémont, Strong Integration of Spatial Domains and Operators in a Relational Database System. *3rd Int. Symp. on Advances in Spatial Databases* (*SSD'93*), LNCS 692, 53-72, 1993.

Lee D.T. & F.P. Preparata, Computational Geometry - A Survey. *IEEE Transactions on Computers*, C-33(12), 1072-1101, 1984.

Lipeck U. & K. Neumann, Modelling and Manipulating Objects in Geoscientific Databases. *5th Int. Conf. on the Entity-Relationship Approach*, 67-86, 1987.

Manola F. & J.A. Orenstein, Toward a General Spatial Data Model for an Object-Oriented DBMS. *12th Int. Conf. on Very Large Data Bases*, 328-335, 1986.

Mantey P.E. & E.D. Carlson, Integrated Geographic Data Bases: The GADS Experience. *Data Base Techniques for Pictorial Applications*, Springer-Verlag, 173-190, 1980.

Mehlhorn K. & S. Näher, LEDA: A Platform for Combinatorial and Geometric Computing. *Communications of the ACM*, 38(1), 1995.

Milenkovic V., Verifiable Implementations of Geometric Algorithms Using Finite Precision Arithmetic. *Artifical Intelligence*, 37, 377-401, 1988.

Müller V., N.W. Paton, A.A.A. Fernandes, A. Dinn & M.H. Williams, Virtual Realms: An Efficient Implementation Strategy for Finite Resolution Spatial Data Types. *7th Int. Symp. on Spatial Data Handling* (*SDH'96*), Taylor & Francis, 697-710, 1997.

Ooi B.C., R. Sacks-Davis & K.J. McDonell, Extending a DBMS for Geographic Applications. *IEEE Int. Conf. on Data Engineering*, 590-597, 1989.

Oosterom P. van, Vertegaal W., Hekken M. van & T. Vijlbrief, Integrated 3D Modelling within a GIS. *Int. Workshop on Advanced Geographic Data Modelling* (*Spatial Data Modelling and Query Languages for 2D and 3D Applications*), 80-95, 1994.

Osborn S.L. & T.E. Heaven, The Design of a Relational Database System with Abstract Data Types for Domains. *ACM Transactions on Database Systems*, 11(3), 357-373, 1986.

Ottmann T., G. Thiemt & C. Ullrich, Numerical Stability of Geometric Algorithms. *3rd ACM Symp. on Computational Geometry*, 119-125, 1987.

Overmars M.H., Computational Geometry on a Grid: an Overview. *Theoretical Foundations for Computer Graphics and CAD*, Springer-Verlag, 167-184, 1988.

Peuquet D.J., The Use of Spatial Relationships to Aid Spatial Database Retrieval. *2nd Int. Symp. on Spatial Data Handling*, 459-470, 1986.

Peuquet D.J., Toward the Definition and Use of Complex Spatial Relationships. *3rd Int. Symp. on Spatial Data Handling*, 211-223, 1988.

Pigot S., Topological Models for 3D Spatial Information Systems. *5th Int. Symp. on Spatial Data Handling*, 344-360, 1992.

Preparata F.P. & M.I. Shamos, *Computational Geometry*. Springer Verlag, 1985.

Pullar D.V. & M.J. Egenhofer, Toward Formal Definitions of Topological Relations Among Spatial Objects. *3rd Int. Symp. on Spatial Data Handling*, 225-241, 1988.

Pullar D., Data Definition and Operators on a Spatial Data Model. *ACSM-ASPRS Annual Convention*, 197-202, 1988.

Puppo E. & G. Dettori, Towards a Formal Model for Multi-Resolution Spatial Maps. *4th Int. Symp. on Advances in Spatial Databases* (*SSD'95*), 152-169, 1995.

Randell D., Z. Cui & A.G. Cohn, A Spatial Logic Based on Regions and Connection. *3rd Int. Conf. on the Principles of Knowledge Representation and Reasoning*, 1992.

Ridder T. de, The ROSE System. Modula-2 Program System (Source Code). Fernuniversität Hagen, Praktische Informatik IV, Software Report 1, 1995. Available as a LaTeX file for printing and/or as a compressed collection of ASCII files.

Rigaux P. & M. Scholl, Multi-Scale Partitions: Application to Spatial and Statistical Databases. *4th Int. Symp. on Advances in Spatial Databases* (*SSD'95*), 170-183, 1995.

Saalfeld A., Lattice Structures in Geography. *7th Int. Symp. on Computer-Assisted Cartography, AUTOCARTO 7*, 482-489, 1985.

Schneider M., *Spatial Data Types in Database Systems – Finite Resolution Geometry for Geographic Information Systems*, LNCS 1288, Springer-Verlag, 1997.

Schneider M., Modeling Spatial Objects with Undetermined Boundaries Using the Realm/ROSE Approach. *Geographic Objects with Indeterminate Boundaries*, GISDATA Series 2, Taylor & Francis, Chapter 10, 141-152, 1996.

Schneider M., Uncertainty Management for Spatial Data in Databases: Fuzzy Spatial Data Types. *6th Int. Symp. on Advances in Spatial Databases* (*SSD'99*), 1999, to appear.

Scholl M. & A. Voisard, Thematic Map Modeling. *1st Int. Symp. on the Design and Implementation of Large Spatial Databases* (*SSD'89*), LNCS 409, 167-190, 1989.

Sistla A.P., O. Wolfson, S. Chamberlain & S.D. Dao, Modeling and Querying Moving Objects. *13th Int. Conf. on Data Engineering*, 422-432, 1997.

Smyth C.S., A Reference Data Model for Spatial and Geographic Applications. *4th Int. Symp. on Spatial Data Handling*, 869-878, 1990.

Stonebraker M., B. Rubenstein & A. Guttmann, Application of Abstract Data Types and Abstract Indices to CAD Databases. *ACM/IEEE Conf. on Engineering Design Applications*, 107-113, 1983.

Stonebraker M., Inclusion of New Types in Relational Data Base Systems. *2nd Int. Conf. on Data Engineering*, 262-269, 1986.

Svensson P. & Z. Huang, Geo-SAL: A Query Language for Spatial Data Analysis. *2nd Int. Symp. on Advances in Spatial Databases* (*SSD'91*), LNCS 525, 119-140, 1991.

Tilove R.B., Set Membership Classification: A Unified Approach to Geometric Intersection Problems. *IEEE Transactions on Computers*, C-29, 874-883, 1980.

Tomlin C.D., *Geographic Information Systems and Cartographic Modeling*. Prentice Hall, 1990.

Wagner D., A Method of Evaluating Polygon Overlay Algorithms. *ACSM-ASPRS Annual Convention*, 173-183, 1988.

Waugh T.C. & R.G. Healey, The GEOVIEW Design - A Relational Data Base Approach to Geographic Data Handling. *Int. Journal of Geographical Information Systems*, 1(2), 101-118, 1987.

Worboys M.F., A Generic Model for Planar Geographical Objects. *Int. Journal of Geographical Information Systems*, 4(4), 369-385, 1992.

Worboys M.F., A Unified Model for Spatial and Temporal Information. *The Computer Journal*, 37(1), 1994.

Worboys M.F. & P. Bofakos, A Canonical Model for a Class of Areal Spatial Objects. *3rd Int. Symp. on Advances in Spatial Databases* (*SSD'93*), LNCS 692, 36-52, 1993.

Wilms P.F., P.M. Schwarz, H.-J. Schek & L.M. Haas, Incorporating Data Types in an Extensible Database Architecture. *3rd Int. Conf. on Data and Knowledge Bases*, 180-192, 1988.

Yao F.F.. Computational Geometry. Algorithms and Complexity, *Handbook of Theoretical Computer Science*, A, Elsevier Science Publishers B.V., 343-389, 1992.