Chapter 13

REAL ALGEBRAIC GEOMETRY AND
CONSTRAINT DATABASES

Floris Geerts
Hasselt University, Transnational University of Limburg & University of Edinburgh

Bart Kuijpers
Hasselt University & Transnational University of Limburg

Second Reader
Peter Revesz
University of Nebraska–Lincoln

1. From the relational database model to the constraint
database model

The constraint database model can be seen as a generalization of the classical
relational database model that was introduced by Codd in the 1970s to deal with
the management of alpha-numerical data, typically in business applications
(Codd, 1970). A relational database can be viewed as a finite collection of
tables or relations that each contain a finite number of tuples.

Fig. 13.1 shows an instance of a relational database that contains the two
relations Beer and Pub. This database contains tourist information about beers
and the pubs where they are served. It also contains the location of the pubs,
given in \((x, y)\)-coordinates on some tourist map. Each relation contains a finite
number of tuples. A relational database is usually modeled following a database
schema. A schema contains information on the relation names and on the names
of the attributes appearing in relation. In this example, the attributes of Beer
are Name, Pub, City and Postal code. The complete schema of the relational
database of Fig. 13.1 could be written as Beer(Name, Pub, City, Postal code),
Pub(Pub, x, y).
The \( x \) and \( y \) attributes of the relation \textit{Pub} have a geometric or geographic interpretation. But values of these attributes can simply be stored as numbers, as is usually done in business databases. A tourist could consult this database to find out the locations of pubs where his/her preferred beers are served. First-order logic based languages (and their commercial versions, such as SQL) are used in the relational database model, to formulate queries like this. The vocabulary of these logics typically contains the relation names appearing in the schema of the input database. For instance, the first-order formula

\[
\varphi(x, y) = \exists p \exists c \exists p' (\text{Beer}(\text{Westvleteren}, p, c, p') \land \text{Pub}(p, x, y))
\]

when interpreted over the database of Fig. 13.1, defines the \((x, y)\)-coordinates of the location of the pubs where they serve my favorite beer.

But a tourist is usually also given more explicit geographic information, e.g., in the form of maps such as the one depicted in Fig. 13.2 and he/she typically wants to ask questions that combine spatial and alpha-numeric information, such as “Where in Flanders, not too far from the river Scheldt, can I drink a Duvel?”

In the relational database model, it is difficult to support queries like this one. Unlike the locations of pubs, the locations of rivers or regions would require the storage of infinitely many \( x \)- and \( y \)-coordinates of points. Storing infinitely many tuples is not possible and in computer science it is customary to find \textit{finite} representations of even infinite sets or objects.

In the 1980s, extensions of the relational model were proposed with special-purpose data types and operators. Data types like “polyline” and “polygon” were introduced to support, e.g., the storage of rivers and regions. Ad-hoc operations like intersection of polygons were added to popular query languages such as SQL. Since then, spatial database theory and technology has developed