Visualising Lattices with Tabular Diagrams ⋆

Uta Priss

Ostfalia University, Wolfenbüttel, Germany
www.upriss.org.uk

Abstract. Euler and Hasse diagrams are well-known visualisations of sets. This paper introduces a novel type of visualisation, Tabular diagrams, which is essentially a type of Euler diagram where lines have been omitted or a 2-dimensional Linear diagram. Tabular diagrams are utilised to visualise lattices in comparison to Euler and Hasse diagrams. For that purpose, lattice terminology is applied to all three types of diagrams.

1 Introduction

Formal Concept Analysis (FCA) is a mathematical method for knowledge representation with many applications that uses lattice theory [5]. A challenge for FCA is that users need to be trained in order to be able to read Hasse diagrams of lattices. Euler diagrams tend to be perceived as more "intuitive" to read than Hasse diagrams but also have certain disadvantages compared to Hasse diagrams [8]. An experiment of Chapman et al. [3] demonstrates that Linear diagrams are more effective for certain retrieval tasks than Euler diagrams. But Linear diagrams often require repetition of attributes in order to represent the data of an application. Tabular diagrams as suggested in this paper are essentially 2-dimensional Linear diagrams which can represent more attributes without repetitions than Linear diagrams. The usability of Tabular diagrams should be similar to Linear diagrams because reading tables is a skill that most users are accomplished in. But the focus of this paper is on structural aspects, not on usability which will be left for future research.

Euler diagrams are a form of graphical representation of set theory that is similar to Venn diagrams but leaves off any regions that are known to be empty. Fig. 1 shows three Euler diagrams with corresponding Hasse diagrams (which are explained in the next Section). Euler diagrams consist of closed curves with labels representing sets. The smallest undivided areas in an Euler diagram are called *minimal regions*. Regions are defined as sets (or unions) of minimal regions. Zones are maximal regions that are within a set of curves and outwith the remaining curves. Thus for a set of sets, $A := \{a_1, ..., a_n\}$, zones can be described as $(a_1 \cap ... \cap a_k) \setminus (a_{k+1} \cap ... \cap a_n)$. The reason for distinguishing between minimal regions and zones is that zones are the smallest mathematical meaningful areas of an Euler diagram whereas minimal regions are the smallest visibly undivided areas. In Euler diagrams that are not *well-formed*, it is possible that zones and minimal regions do not coincide and some zones consist of several minimal regions. A number of other criteria for being well-formed can be specified (cf.

⋆ Published in Basu et al. Proc of DIAGRAMS'21, LNAI 12909, Springer-Verlag

[4]), for example, allowing at most two curves to meet in any point and disallowing curve edges to meet in more than one adjacent point. In this paper we are only discussing Euler diagrams that fulfil a "zones=minimal regions" condition. Shading of a zone is sometimes used in order to indicate that a zone must be empty. In some cases, it is not possible to generate a well-formed Euler diagram without using shading.

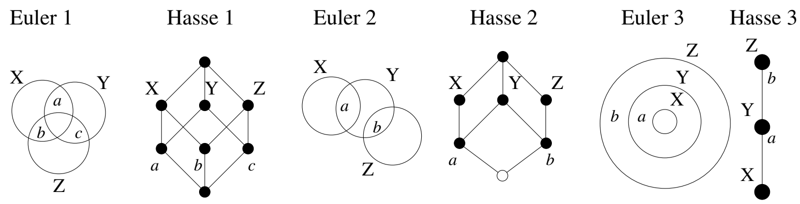

Fig. 1. Euler and Hasse diagrams

2 Euler Diagrams and FCA

Because the focus of this paper is on FCA applications, the normal Euler diagram terminology is amended in this paper using notions from FCA. In this paper, the labels of curves are called *attributes*. Elements of sets denoted by attributes are called *objects*. These two notions allow a distinction between the sets and elements of an application, consisting of objects and attributes, from other sets and elements encountered in the discussion about Euler diagrams. The notions have purely a structural meaning. Thus objects and attributes can correspond to any kind of data in an application. Regions that can be described as intersections of attributes without using union or set difference as operations are called *i-regions*. The following lemma summarises well-known structural properties of zones and i-regions:

Lemma 1. *There is a 1-to-1 correspondence between zones and i-regions in an Euler diagram: for each zone* $(a_1 \cap ... \cap a_k) \setminus (a_{k+1} \cap ... \cap a_n)$ *its corresponding i-region is* $a_1 \cap ... \cap a_k$ *or denoted as an element of a powerset:* $\{a_1, ..., a_k\}$*. A set of i-regions together with* $\subseteq$ *forms a partially ordered set. This ordering can be isomorphically transferred to an ordering amongst zones if a zone* a *is called* below another zone b *(written as* $a \leq b$*) if* $a_1 \subseteq b_1$ *holds for the i-regions* a_1 *and* b_1 *corresponding to* a *and* b*.*

It is know from lattice theory that a subset of a powerset forms a lattice if it is closed with respect to intersections. Therefore a set of i-regions forms a lattice if it is closed with respect to intersections of sets of attributes. Thus in Fig. 1 Euler diagram 2, $\{\{\}, \{X\}, \{Y\}, \{Z\}, \{X, Y\}, \{Y, Z\}\}$, the intersection of the sets $\{X, Y\}$ and $\{Y, Z\}$ is required ($\{Y\}$) but not intersections of attributes such as $X \cap Z$. The i-regions in Euler diagrams 1 and 3 in Fig. 1 form lattices, the i-regions in Euler diagram 2 do not because the intersection over the empty set, i.e. $\{X, Y, Z\}$, is missing. If a set of i-regions does not form a lattice, it can be embedded into a lattice by adding the missing intersections.

The remainder of this section translates Euler diagram terminology into lattice terminology as provided by FCA. It is assumed in this paper that all sets of i-regions are embedded into lattices. An i-region $\{a_1, ..., a_k\}$ then determines a *concept* as a pair of sets of objects and attributes. The set $\{a_1, ..., a_k\}$ of attributes of a concept is called the *intension* of the concept. The set of objects of a concept is called an *extension* and consists of all objects that are elements of $a_1 \cap ... \cap a_k$. The ordering amongst i-regions discussed in Lemma 1 is called *conceptual ordering*, the lattice a *concept lattice*. Concepts that are added during an embedding into a lattice are called *supplemental concepts* in this paper. They correspond to missing or shaded zones of an Euler diagram. Supplemental concepts and thus concepts corresponding to missing or shaded zones of an Euler diagram do not have any *immediate objects* in their extension that is objects that belong to them but not to any lower concepts.

Hasse diagrams (as in Fig. 1) are a well-known diagrammatic representation of partially ordered sets and thus of lattices. The ordering is visualised as a transitive reduction because all edges that are implied by the transitivity of the ordering are omitted. For concept lattices, the ordering represents the conceptual ordering. Hasse diagrams are directed graphs where all edges are read in the direction from the visually lower to the higher end. Nodes in a Hasse diagram correspond to concepts. In order to read the extension of a concept in a Hasse diagram, all objects at the concept or at any concepts below it (according to the ordering) need to be collected. In order to read the intension of a concept, all attributes at a concept or at any concepts above it need to be collected. The nodes of supplemental concepts are drawn as unfilled circles in the Hasse diagrams (as in Hasse diagram 2 in Fig. 1). One advantage of using FCA is that it provides a variety of existing software[1] for generating lattices and their Hasse diagrams from data.

For the purposes of this paper, it should not be necessary to explain more details about FCA. Priss [8] provides a slightly more detailed introduction to FCA and its relationship to Venn, Euler and Hasse diagrams. Priss [8] concludes that lattice-theoretical properties can provide some further clues about when Euler diagrams are well-formed and discusses some advantages and disadvantages of Hasse diagrams compared to Euler diagrams.

3 Tabular Diagrams

Apart from Euler and Hasse diagrams, a further visualisation of concept lattices called *Tabular diagrams* is introduced in this paper. Tabular diagrams are essentially a 2-dimensional version of the "Linear diagrams" invented by Leibniz and discussed by Chapman et al. [3]. Tabular diagrams are best characterised as matrix-based diagrams (according to [2]) and appear under many different notions (mosaic plots/displays, contingency tables, Karnaugh maps) often with additional purposes, for example displaying frequencies within each zone. So far we have not been able to find a more general notion for or discussion of Tabular diagrams in the literature.

Fig. 2 shows an example of a lattice where the attributes are the numbers 2, 3, 5, and 7 and the objects are numbers that are products of prime numbers. Objects have an

[1] cf. https://upriss.github.io/fca/fcasoftware.html

attribute if they are divisible by that number. For example, 30 has the attributes 2, 3 and 5, but not 7. Such lattices as on the left side of Fig. 2. are well-known as examples of Boolean algebras. For the Tabular diagram in the middle, the set of attributes is partitioned into two sets. Intersections amongst attributes that belong to the same partition are indicated by overlapping brackets and a separate column. Intersections amongst attributes that belong to different partitions correspond to regions of the table. Objects are written into the zones. The Tabular diagram in Fig. 2 contains as many zones as there are concepts, but it is possible to have fewer zones than concepts if there are supplemental concepts. It is not possible to add another attribute to the Tabular diagram in Fig. 2 which intersects with all previous attributes. The strategy for Linear diagrams (as in Fig. 2, on the right) is to repeat attributes if a diagram is impossible to construct otherwise. Attributes can also be repeated for Tabular diagrams. But not all Tabular diagrams with more than four attributes require repetitions. Shading can be used for zones that do not belong to concepts at all if the Tabular diagram cannot be constructed otherwise. Supplemental concepts correspond to empty or missing zones but not to shaded zones. The bottom concept can be omitted if it has an empty extension.

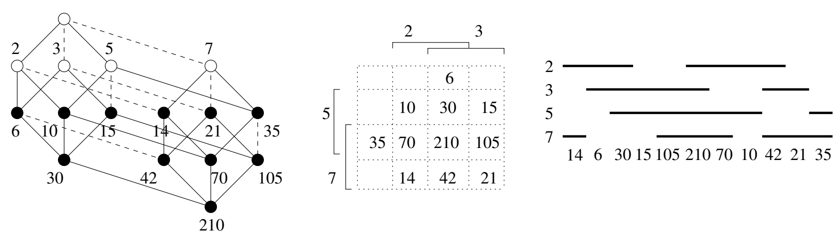

Fig. 2. Hasse, Tabular and Linear diagrams of a Boolean lattice with 4 attributes

All types of diagrams (Hasse, Euler, Linear and Tabular) are difficult to visually parse for large sets of data. Therefore, the fact that Tabular diagrams have some limitations for larger data sets does not necessarily disadvantage them compared to the other types. The dashed lines in the Hasse diagram in Fig. 2 indicate pairs of zones which are not neighbours in the Tabular diagram even though they could be neighbours if the rows and columns were permuted differently. In the Tabular diagrams in this paper, the top is usually the left upper corner and the bottom is close to the centre or omitted. Thus some relationships are more easily visible in a Hasse diagram than in a Tabular diagram. Tracing lines can become difficult in a large Hasse diagram. Determining which attributes belong to an object seems to be easier in a Tabular diagram because it only involves reading the row and column headings.

Fig. 3 contains another example modelled as isomorphic Hasse, Euler and Tabular diagrams. Brackets are not needed in Tabular diagrams for attributes that only span one row or column. The bottom element is omitted in all four diagrams. Fig. 3 demonstrates that Tabular diagrams correspond to Euler diagrams where curves are rectangular and arranged in rows and columns. Euler diagrams with parallel curves are usually considered not well-formed. We would argue that in the case of Tabular diagrams the parallel

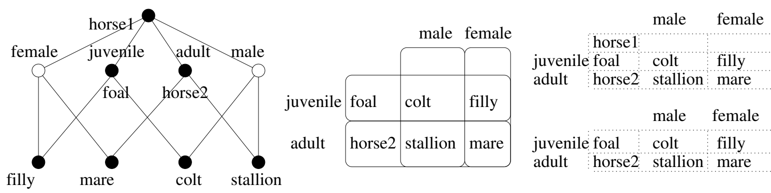

Fig. 3. Hasse, Euler and 2 Tabular diagrams of a linguistic example

curves are not a limitation because rows and columns of a table are easy to read. The Euler diagram in Fig. 3 cannot be drawn in a well-formed manner. The word "horse" has two meanings in this example: "horse1" refers to the species, "horse2" to the adult animal. The outer zone may be less obvious to see in the Euler diagram. The object "horse1" which would have to be placed into the outer zone has therefore been omitted in the Euler diagram. If the object "horse1" is deleted from the example, then the Tabular diagram can be reduced to the version shown in the lower corner. The lower Tabular diagram still contains 9 concepts: 4 labelled rows and columns, 4 intersections of rows and columns and the table as a whole, but three supplemental concepts are omitted.

From a structural viewpoint the question arises as to which sets of data can be represented by Tabular diagrams without shading or repetition. With respect to repetitions Petersen's [7] analysis provides some clues. Translated into the terminology of this paper, Petersen provides a characterisation with FCA of when a Linear diagram can be represented without repetitions. Her characterisation essentially checks whether a planar lattice exists for the data in which a line can be drawn from each object to the bottom concept without crossing the edges of the Hasse diagram. If a Hasse diagram is tree-like after omitting its bottom node, then it fulfils Petersen's condition. Furthermore if a Hasse diagram contains a cycle, the cycle must contain the bottom node or be at the side of the diagram, but not in the middle in order to fulfil the condition.

For Tabular diagrams the question is then whether the set of attributes can be partitioned into two sets which are representable as Linear diagrams without repetitions. In that case, it is known from FCA that the resulting lattice can be embedded into the direct product of the two lattices corresponding to the Linear diagrams. Most likely providing a characteristic or algorithm for producing Tabular diagrams in this manner is non-trivial. Because there are $2^n/2$ possible ways to split a set of n attributes into two partitions, calculating all possibilities is not feasible for large sets. But as mentioned before, diagrams are mainly of interest for fairly small sets of data and some heuristics can be applied to reduce the number of possibilities, such as:

- check whether omitting the bottom concept generates a more tree-like structure,
- look for partitions that split the set of attributes approximately in half,
- check whether the attribute set can be simplified (for example if an attribute and its negation exist in the data, only one of them may be required),
- determine if certain lattice properties exist in the data using FCA software and use them to partition the data (for example, the attributes of a chain or antichain should be kept in one partition).

- According to FCA: if a lattice is a direct product (possibly minus the bottom concept), then the number of rows and columns of the Tabular diagram should be a divisor of the number of concepts (possibly minus 1). Fig. 2 and Fig. 3 show examples for this case (with 4 × 4 and 3 × 3 concepts).

4 More Examples of Tabular Diagrams

This section discusses some slightly more complex examples that pertain to data from applications. The Hasse diagram in Fig. 4 is a well-known FCA example of a lexical field of bodies of water based on linguistic "componential analysis" which determines semantic components of words [6]. Each semantic component relates to a positive and a negative attribute depending on whether it exists or not in the word. This results in each object in the lattice (with the exception of "channel") having either the positive or the negative counterpart of each of the four attributes (such as "inland" or "maritime"). For "channel" it is not specified within the provided data whether it is natural or artificial. The supplemental concepts correspond to more general concepts (such as "stagnant natural body of water") which are not lexicalised in the data.

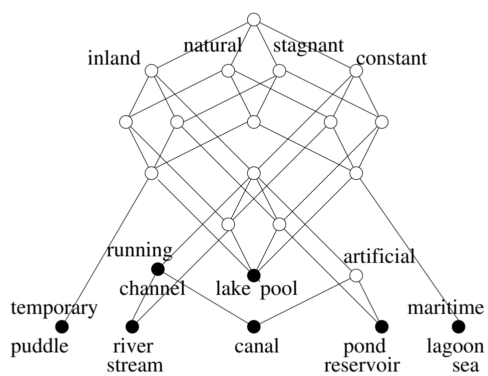

Fig. 4. Hasse diagram of a lexical field

The left side of Fig. 5 shows a Tabular diagram for the lattice in Fig. 4. It contains 23 zones, corresponding to 23 concepts except the bottom concept. The 16 empty zones represent supplemental concepts. The shaded zones belong to combinations which do not exist in the data at all, for example, there is no concept for maritime and artificial because no object has both those attributes. If all the non-shaded zones of one attribute are adjacent, it is possible to represent the non-shaded zones by drawing a curve around them as shown in the right side of Fig. 5. In a sense this corresponds to adding a third dimension to the Tabular diagram. The diagram on the right side of Fig. 5 also contains 23 zones, but the zone "constant, artificial, inland" might be overlooked because it is empty, quite small and not rectangular in shape. The top row and left column can be omitted if it is not desired to show all supplemental concepts. One might be tempted

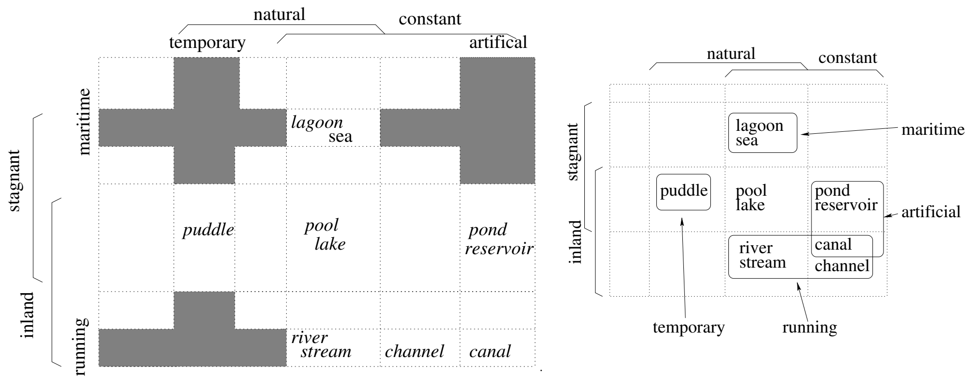

Fig. 5. Tabular diagrams for the lexical field in Fig. 4

to simply add the negative attributes as row and column headings in the right Tabular diagram in Fig. 5. But that would change the data. For example, the information that temporary bodies of water are always inland and stagnant would be lost.

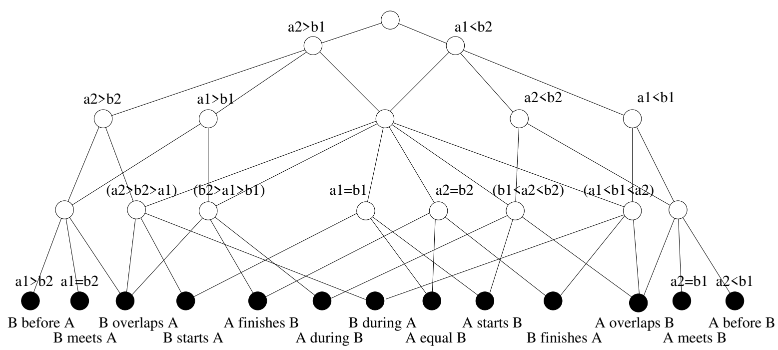

Fig. 6. Hasse diagram of Allen's [1] temporal relations

Fig. 6 and Fig. 7 display a further example containing Allen's 13 temporal relations [1]. These relations are used in formal ontologies in order to express all possibilities of how two temporal intervals (A and B) can be related to each other, such as one occurring before the other one or both overlapping based on the relationships of their start points (a1 and b1) and end points (a2 and b2). The concept lattice in Fig. 6 contains 29 concepts (without the bottom concept). Allen himself uses a table with 144 fields to display the transitivity relationships amongst the 13 temporal relations. A lattice representation summarises the relationships because each field in Allen's table is the intension of a concept. The lattice contains all possible logical combinations. In this example, all concepts other than the neighbours of the bottom concept are supplemental yielding a Tabular diagram in Fig. 7 without empty zones. Because of the missing supplemental

		a2>b1	a2>b1	a2>b1 / a1<b1	a1<b1	a1<b1
		a1>b1	a1=b1	(a1<b1<a2)	a2=b1	a2<b1
a1<b2	a2<b2	A during B	A starts B	A overlaps B	A meets B	A before B
a1<b2	a2=b2	A finishes B	A equal B	B finishes A		
a1<b2 / a2>b2	(a1<b2<a2)	B overlaps A	B starts A	B during A		
a2>b2	a1=b2	B meets A				
a2>b2	a1>b2	B before A				

Fig. 7. Tabular diagram of Allen's [1] temporal relations

concepts it is more difficult in the Tabular diagram to count the total number of concepts in the lattice because all possible intersections of attributes need to be considered.

5 Conclusion

Tabular diagrams provide a concise representation of the information contained in a concept lattice. It is straightforward to determine the attributes for each object by retrieving the row and column headings and to determine the objects for each attribute by determining the zones belonging to the row or column headings. Implications can also be read from Tabular diagrams (such as "temporary" implies "natural", "stagnant" and "inland" in Fig. 5). If supplemental concepts are omitted, then counting all concepts is more challenging but it may not be necessary to read such information from the diagrams because it can be algorithmically determined.

Many questions are left for future research, such as which data can be represented as non-repetitive Tabular diagrams, what are suitable algorithms to create and optimise Tabular diagrams and how does the readability and visual scalability of Tabular diagrams compare to other types of diagrams.

References

1. Allen, J. (1983). *Maintaining Knowledge about Temporal Intervals.* Communications of the ACM 26, 11, p. 832-843.
2. Alsallakh, B.; Micallef, L.; Aigner, W.; Hauser, H.; Miksch, S.; Rodgers, P. (2016). *The State-of-the-Art of Set Visualization.* Computer Graphics Forum, 35, 1. p. 234-260.
3. Chapman, P.; Stapleton, G.; Rodgers, P.; Micallef, L.; Blake, A. (2014). *Visualizing sets: An empirical comparison of diagram types.* In: International Conference on Theory and Application of Diagrams, Springer, p. 146-160.
4. Flower, J.; Fish, A.; & Howse, J. (2008). *Euler diagram generation.* Journal of Visual Languages & Computing, 19, 6, p. 675-694.
5. Ganter, B.; Wille, R. (1999). *Formal Concept Analysis. Mathematical Foundations.* Berlin-Heidelberg-New York: Springer.
6. Kipke, U.; Wille, R. (1987). *Formale Begriffsanalyse erläutert an einem Wortfeld.* LDV-Forum 5, p. 31-36.
7. Petersen, W. (2010). *Linear Coding of Non-Linear Hierarchies: Revitalization of an Ancient Classification Method.* In A. Fink et al., (eds.), Advances in Data Analysis, Data Handling and Business Intelligence, Springer, p. 307-316.
8. Priss, U. (2020). *Set Visualisations with Euler and Hasse Diagrams.* In: Graph Structures for Knowledge Representation and Reasoning, Springer. p. 72-83.