A data representation for abstract reasoning

J.J. Sarbo
Computing Science Institute
University of Nijmegen
The Netherlands

J.I. Farkas
J.A.T.E., Department of
Pedagogy and Psychology
Szeged, Hungary

Abstract

Straightforward representation of relational specifications by trees, graphs etc. are usually inadequate when a relation on specifications has to be determined. A remedy for this problem is provided by the Galois lattice representation of relations. By extending Galois lattices with semantical information we get a supporting representation for abstract reasoning. We exemplify this lattice representation by a specification of card games.

1 Introduction

Whether a specification (or part of it) is an instance of a more general one can be useful information when one aims at improving his(her) specification, e.g. by introducing more suitable abstractions. This problem arises, amongst others, in the top-down development of specifications. In such a case, one gives first a general description of the problem and then refines it, makes it more specific, such that it finally covers a large part of the problem domain. The one uncovered is then used to complete the specification in an ad hoc way, usually.

Due to the rigidity of most formalisms and representations, it is in most cases very difficult to systematically improve, e.g. transform, such an abstract specification so that it covers the whole problem domain. We show that the Galois lattice representation may help to cope with this problem in a systematic way. This is due to, amongst others, the elegant algebraic properties, like closure and decomposition, of these lattices.

Decomposition of Galois lattices is described in [Wil83]. Batch and incremental algorithms for building Galois lattices are found in [CR93] and [GMA95]. A related approach using automatic discovery of implication rules from concept lattices is introduced in [GM94].

2 Galois lattices

Galois (also called concept) lattices were first defined by R. Wille ([Wil82]). Basically a *concept lattice* is a representation of a (e.g. binary) relation R between a set of objects G (Gegenstände) and a set of attributes M (Merkmale). The triple (G, M, R) is called a *context*. For $g \in G$, $m \in M$, $(g, m) \in R$, or equivalently, gRm, iff object g has the attribute m. A generalization of R to an n-ary relation is found in [Wil82]. The concept lattice arises from the context by applying a Galois connection between the power sets of G and M. This Galois connection is a particular one, called the *polar* ([Bir67]).

We define polars stepwise, starting from a set of more simple functions.

Definition 2.1 For relation $R \subseteq G \times M$ we define a function to $\mathcal{P}.M$ from G by defining for every $g \in G$: $g.R = \{m : m \in M : gRm\}$.

Definition 2.2 For relation $R \subseteq G \times M$ we define a function to $\mathcal{P}.G$ from M by defining for every $m \in M$: $R.m = \{g : g \in G : gRm\}$.

Definition 2.3 For every $\mathsf{G} \in \mathcal{P}.G$ and $\mathsf{M} \in \mathcal{P}.M$, we define the *right polar* and *left polar* as $R^r.\mathsf{G} = \cap.(g : g \in \mathsf{G} : g.R)$ and $R^l.\mathsf{M} = \cap.(m : m \in \mathsf{M} : R.m)$, resp.

Definition 2.4 For $\mathsf{G} \in G$, $\mathsf{M} \in M$, the pair (G, M) is called a *concept* if $\mathsf{G} = R^l.\mathsf{M}$ and $\mathsf{M} = R^r.\mathsf{G}$.

Definition 2.5 The *concept lattice* (Begriffsverband) of a context (G, M, R), denoted as $\underline{\mathcal{B}}(G, M, R)$, is the set of concepts with the ordering: $(\mathsf{G}_1, \mathsf{M}_1) \leq (\mathsf{G}_2, \mathsf{M}_2) \equiv \mathsf{G}_1 \subseteq \mathsf{G}_2$ $(\equiv \mathsf{M}_1 \supseteq \mathsf{M}_2)$.

Theorem 2.6 (1) Let (G, M, R) be a context. Then $\underline{\mathcal{B}}(G, M, R)$ is a complete lattice in which infimum and supremum can be described as follows:

a $\underset{j \in J}{\sqcup}(A_j, B_j) = ((R^l.R^r.\underset{j \in J}{\cup} A_j), \underset{j \in J}{\cap} B_j)$,
b $\underset{j \in J}{\sqcap}(A_j, B_j) = (\underset{j \in J}{\cap} A_j,\ R^r.R^l.(\underset{j \in J}{\cup} B_j))$.

(2) A complete lattice L is isomorphic to $\underline{\mathcal{B}}(G, M, R)$ iff there are mappings $\gamma \in L \leftarrow G$ and $\mu \in L \leftarrow M$ such that $\gamma.G$ is join-dense in L (i.e. $L = \{\sqcup \mathsf{X} \mid \mathsf{X} \subseteq \gamma.G\}$), $\mu.M$ is meet-dense in L (i.e. $L = \{\sqcap \mathsf{X} \mid \mathsf{X} \subseteq \mu.M\}$), and $gRm \equiv \gamma.g \leq \mu.m$ for all $g \in G$ and $m \in M$.

2.7 An example

Our main example is about card games.[1] This is a rich domain of knowledge and we will be able to specify only a small part of it.

We begin with the specification of the simple notion: trichotomy. The reader may wonder, why we need this notion. We can reassure him (her): alike to a Tsjechov drama, a gun hanging on the wall in the first act will certainly be fired sometime in the play.

What is trichotomy? As Websters defines it: "*a division into three parts, elements, or classes*". We formalize this notion as follows. Let the parts be given by $G = \{1, 2, 3\}$, and their properties by $M = \{$differs from 1 and 2, differs from 1 and 3, differs from 2 and 3$\}$. In fact, G and M can be chosen equivalent. Following the definition above, we define R as $\{$(1, differs from 2 and 3), (2, differs from 1 and 3), (3, differs from 1 and 2)$\}$.

[1] Information about card games is found at `http://www.netlink.co.uk/users/pagat/`.

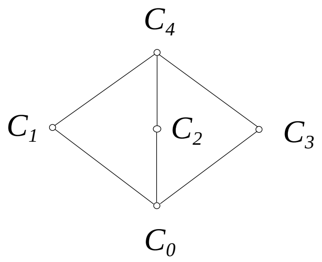

Figure 1: Concept lattice of trichotomy

The concepts of (G, M, R) are: $C_0 = (\{\ \}, \{$differs from 1 and 2, differs from 1 and 3, differs from 2 and 3$\})$, $C_1 = (\{1\}, \{$differs from 2 and 3$\})$, $C_2 = (\{2\}, \{$differs from 1 and 3$\})$, $C_3 = (\{3\}, \{$differs from 1 and 2$\})$, $C_4 = (\{1, 2, 3\}, \{\ \})$. The concept lattice is depicted in Fig. 1.

After this short detour let us go further with the specification of card games. From a general point of view any card game can be characterized by cards, players and laws. Cards have usually some code, e.g. $\{2, 3, \ldots, J, Q, K, A\}$ and $\{\clubsuit, \diamondsuit, \heartsuit, \spadesuit\}$. Laws are basically of two sorts: playing rules, e.g. follow suit, follow picture, and coding rules, e.g. $2 < 3 < \ldots < K < A$ and $\clubsuit < \diamondsuit < \heartsuit < \spadesuit$. There may be rules for the players as well, e.g. playing in pairs.

In sum, we can specify cards and players relationally as follows (we represent relations by matrices):

	codes
laws	+
cards	+

	rules
laws	+
players	+

The composition of the these relations yields our general specification of card games:

	codes	*rules*
laws	+	+
players		+
cards	+	

The concepts are:

$$
\begin{aligned}
M_0 = &\quad (\{\text{laws}\}, \{\text{codes}, \text{rules}\})\ , \\
M_1 = &\quad (\{\text{laws}, \text{players}\}, \{\text{rules}\})\ , \\
M_2 = &\quad (\{\text{laws}, \text{cards}\}, \{\text{codes}\})\ , \\
M_3 = &\quad (\{\text{cards}, \text{players}, \text{laws}\}, \{\ \})\ .
\end{aligned}
$$

The concept lattice is depicted in Fig. 2.

Let us emphasize that the above specification is just one of the many possible ones. We only use our example and its development for illustrating the particular features of Galois lattices.

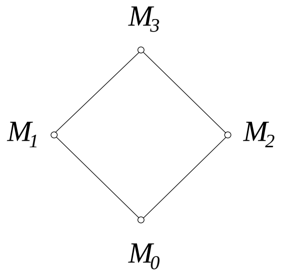

Figure 2: Concept lattice of card games (general specification)

3 From general to specific

The abstract specification of card games is further developed by refinement. There may be various ways for achieving that, as there are numerous different games, specific instances of the abstract specification, available.

Again, we make some particular choice. Assume we have knowledge about bridge, canasta and similar games. We set out our knowledge in the relation below. In this, we make use of more specific notions, like players' rules (the rules to be followed by the players), dealing order (how many cards are dealt), ranking order (the ordering of cards) and party scores (the rules of counting tricks).

These more specific notions refine those of our general specification by the following assignments: laws:=players' rules, codes:=ranking order or dealing order, and rules:=party scores (we omit identity assignments). By this, our general specification can be refined to the following relation.

(players' rules, ranking order)
: In most games a card of a higher rank overtrumps one of a lower rank (of the same suit).

(players' rules, party scores)
: An extra bonus is given, in some games, for the last trick and for a (re)double.

(players, party scores)
: In games played in pairs the score of a pair is the total of that of the individual players.

(cards, ranking order)
: Four of "black"-3 cards can be thrown as a single card (canasta).

(cards, dealing order)
: Cards are dealt according to the dealing order.

Besides bridge and canasta, we have also knowledge also about other card games, resembling to, but not perfectly fitting in the refined specification above. This corresponds to the uncovered part of the problem domain (cf. Sect. 1).

For example, in some games there are card rules (rules about how a card can make a trick) and party rules (rules for calculating the winnings from the scores and tricks). We add these notions to the set of objects. Besides, there are games with particular bidding orders (how the bidding is arranged) and card scores (how the scores of cards are determined). These notions are added to the set of attributes.

Below we set out and exemplify the relation between these notions. (Let us mention that, due to space, the set of these notions and their relation are necessarily incomplete.)

(players, dealing order)
: There is a particular player who deals. There is also an agreement about who deals next, e.g. in clockwise order. In some card games the leader is dealt one card extra (e.g. canasta, rummy).

(card rules, dealing order)
: There are rules about the number and amount of cards to be dealt. For example, 3-4 at a time (e.g. canasta, ulti (a Hungarian card game) and rummy), or one by one (e.g. bridge).

(players' rules, dealing order)
: Whether all cards are dealt (e.g. bridge) or some are undealt (e.g. canasta); if there is a talon (e.g. ulti).

(cards, bidding order)
: If there is a forehand then he/she may take up the talon (stock) first (e.g. ulti) and may bid first (e.g. bridge).

(card rules, bidding order)
: In some games, e.g. in zsírozás (a Hungarian card game), the owner of a bidden card is the partner of the leader. In another game the player owning ♣2 is the leader.

(party rules, bidding order)
: The scores of the leader and his/her partner are summed up (e.g. zsírozás, canasta).

(players, ranking order)
: In some games, e.g. in pesten (a Dutch card game), playing a card-8 gets the next player be skipped over for one round. After playing a card-2 or a joker in canasta, a hand is allowed to take up the pile only if it has two or more cards of the same rank as the one on top of the pile.

(card rules, ranking order)
: There can be different ranking of cards, that is, which card tricks which other, even within the same card game. In ulti, for example, there are two sorts of great slems, A–K and A–10, referring to the rankings $A > K > Q > J > 10$ and $A > 10 > K > Q > J$, resp.

(card rules, card scores)
: In some games the card scores are $10 = 10$, $J = 1$, $Q = 2$, $K = 3$, whereas in Black Jack, $A = 1$ or $A = 10$.

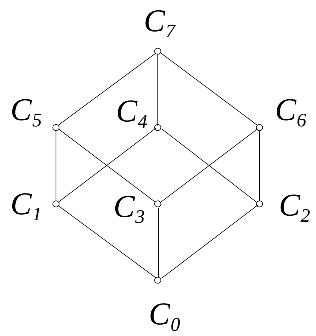

Figure 3: Concept lattice of card games (refined specification)

(party rules, card scores)
: The score of a card may depend on the sort of play, e.g. in canasta, a "red"-3 and a joker yield 100 and 50 points, resp.

(cards, card scores)
: Cards and their score are related in all score calculating games.

(party rules, party scores)
: The party score may depend on the bidden and made tricks (e.g. bridge, ulti).

In sum, we have the following context: $G = \{$players (p), cards (c), card rules (r_c), players' rules (r_p), party rules $(r_P)\}$; $M = \{$dealing order (o_d), bidding order (o_b), ranking order (o_r), card scores (s_c), party scores $(s_P)\}$. The relation between G and M is given by the following matrix:

	o_d	o_b	o_r	s_c	s_P
p	+		+		+
c	+	+	+	+	
r_c	+	+	+	+	
r_p	+		+		+
r_P		+		+	+

The concepts of (G, M, R) are:

$$
\begin{aligned}
C_0 &= (\{\ \}, \{o_d, o_b, o_r, s_c, s_P\}) \ , \\
C_1 &= (\{p, r_p\}, \{o_d, o_r, s_P\}) \ , \\
C_2 &= (\{c, r_c\}, \{o_d, o_b, o_r, s_c\}) \ , \\
C_3 &= (\{r_P\}, \{o_b, s_c, s_P\}) \ , \\
C_4 &= (\{p, r_p, c, r_c\}, \{o_d, o_r\}) \ , \\
C_5 &= (\{p, r_p, r_P\}, \{s_P\}) \ , \\
C_6 &= (\{c, r_c, r_P\}, \{o_b, s_c\}) \ , \\
C_7 &= (\{p, r_p, c, r_c, r_P\}, \{\ \}) \ ,
\end{aligned}
$$

The concept lattice is depicted in Fig. 3.

Relations represented as sets of pairs can be composed by set union. Having this in mind, we can decompose our relation above to three relations:

	o_d	o_b	o_r	s_c
c	+	+	+	+
r_c	+	+	+	+

	o_d	o_r	s_P
p	+	+	+
r_p	+	+	+

	o_b	s_c	s_P
r_P	+	+	+

These relations may be called the specific relations for cards, players and party.

We may wonder whether there are other decompositions possible? We are motivated for finding them as we are not completely satisfied with our specification so far. It must be clear that our general specification from Sect. 2.7 had only a limited impact on the refined specification above.

The question is: how can we find those decompositions? Whereas the composition of relations, as we have shown, can be easily implemented in the context, decompositions of a relation, in general, can be more easily calculated in the concept lattice representation of the context. In the following section we introduce some definitions we will need for finding decompositions.

4 Concept sublattices

Definition 4.1 A lattice L_1 is *homomorphic* to a lattice L_2 if there exists an injective, order-preserving map h from L_2 to L_1. We say then that L_1 is homomorphic to L_2 by h. If h is bijective, it is called an isomorphism.

Definition 4.2 A map $\varphi \in (G_1, M_1, R_1) \leftarrow (G_2, M_2, R_2)$, with $\varphi = (\varphi_G, \varphi_M)$, a pair of injective functions, is called a *context embedding* if $\varphi_G.G_2 \subseteq G_1$, $\varphi_M.M_2 \subseteq M_1$, and $(\varphi_G, \varphi_M).R_2 \subseteq R_1$.

Definition 4.3 Let $\underline{\mathfrak{B}}_1 = \underline{\mathfrak{B}}(G_1, M_1, R_1)$ and $\underline{\mathfrak{B}}_2 = \underline{\mathfrak{B}}(G_2, M_2, R_2)$ be concept lattices. Then $\underline{\mathfrak{B}}_2$ is a *concept sublattice* of $\underline{\mathfrak{B}}_1$ if $\underline{\mathfrak{B}}_1$ is homomorphic to $\underline{\mathfrak{B}}_2$ by some map h and there exist a context embedding φ such that the following diagram commutes (the function to concept lattice from context, described in theorem 2.6, is denoted by the symbol β):

$$\begin{tikzcd}
\underline{\mathfrak{B}}_2 \arrow[d, "h"'] & (G_2, M_2, R_2) \arrow[l, "\beta"'] \arrow[d, "\varphi"] \\
\underline{\mathfrak{B}}_1 & (G_1, M_1, R_1) \arrow[l, "\beta"]
\end{tikzcd}$$

An algorithm for concept sublattices is given in [Sar96]. This algorithm determines for two lattices, $\underline{\mathfrak{B}}_1$ and $\underline{\mathfrak{B}}_2$, whether $\underline{\mathfrak{B}}_1$ contains a sublattice isomorphic to $\underline{\mathfrak{B}}_2$. The lattices are represented as graphs, and the nodes of $\underline{\mathfrak{B}}_2$ are partitioned according to their depth (the depth of a node is the length of the shortest path from the root to that node).

The algorithm traverses the graph of $\underline{\mathcal{B}}_1$ top-down and the partitions of $\underline{\mathcal{B}}_2$ in depth order. In each step, nodes of a partition of $\underline{\mathcal{B}}_2$ are mapped to those nodes of $\underline{\mathcal{B}}_1$ that are "below" the nodes already involved in a map of some earlier partition. Having found a map, the algorithm checks whether it is order preserving.

The complexity of lattice embedding is exponential ([Sar96]). The algorithm can be optimized by only considering irreducible elements of $\underline{\mathcal{B}}_2$. This optimization, however, does not change the worst case complexity, as the number of irreducible elements is in the order of the size of the lattice .

Definition 4.4 Let $\underline{\mathcal{B}} = (B, \leq)$ be a concept lattice and $\{\underline{\mathcal{B}}_j \mid j \in J\}$ a collection of concept lattices. A *decomposition of a concept lattice* $\underline{\mathcal{B}}$ is a decomposition of B, $\{B_i \mid i \in I\}$, such that, for all $i \in I$: $\underline{\mathcal{B}}_i = (B_i, \leq)$ is a concept sublattice of $\underline{\mathcal{B}}$ and $\underline{\mathcal{B}}_i \cong \underline{\mathcal{B}}_j$ for some $j \in J$.

5 From specific to general

So far, we have developed a general and a specific relation for our problem. Let us investigate how these relations (specifications) are related.

We will show that by using definition 4.3 three occurrences of the general specification as concept sublattices of the specific one can be found.

These sublattices are defined by the following pairs of maps (given as sets of assignments):

(1) φ_G = {laws:=r_c, players:=p, card:=c}, φ_M = {codes:=s_c, rules:=o_d} (this corresponds to card games where cards are central, e.g. bridge). The other two maps are: (2) φ_G = {laws:=r_P, players:=p, cards:=c}, φ_M = {codes:=s_c, rules:=s_P}, and (3) φ_G = {laws:=r_p, players:=p, cards:=c, φ_M = {codes:=o_r, rules:=s_P}. Typical card games are Black Jack, for (2), and patience, for (3).

The concept sublattices are:

$A_1 = C_2$-C_4-C_6-C_7: card–oriented card games;

$A_2 = C_3$-C_5-C_6-C_7: winnings–oriented card games.

$A_3 = C_1$-C_4-C_5-C_7: player–oriented card games;

(We call a concept sublattice an *x-oriented* card game if object x occurs in its bottom concept. Clearly, x occurs then in each concept, and so, characterizes the sublattice.)

Note that there are three more sublattices, those having C_0 as bottom element, but these are not concept sublattices as an injective φ cannot be defined.

6 Abstract as concrete

The existence of the three occurrences of the general specification, A_1–A_3, indicate that a better structure for the specification of our problem could be found. The key to such a structure lies in recognizing that the set $\{A_1, A_2, A_3\}$ forms a decomposition of our specification (except for the meaningless bottom concept) and that these instances of the general specification are semantically different.

We emphasize that by reasoning about sublattices, we make an important abstraction: we consider those abstract entities as elementary ones. But then, on a more abstract level, our whole procedure could be repeated. Indeed, we let $G = \{A_1, A_2, A_3\}$, $M = \{$differs from A_2 and A_3, differs from A_1 and A_3, differs from A_1 and $A_2\}$, and R be defined as $\{(A_1,$ differs from A_2 and $A_3)$, $(A_2,$ differs from A_1 and $A_3)$, $(A_3,$ differs from A_1 and $A_2)\}$. Clearly, the concept lattice for this new context is isomorphic to the one of Fig. 1 (an appropriate φ can be found easily). We conclude that our specification of card games is, in essence, a trichotomy!

Curtain

Epilogue

In this paper we show that Galois (or concept) lattices are powerful and flexible 'tools' for representing relational specifications. By allowing context embedding, a semantical extension to Galois lattices, the representation can be used for practical problems. In current research we apply concept lattices in the formal concept analysis of natural language.

References

[Bir67] G. Birkhoff. *Lattice theory.* American Mathematical Society, 1967.

[CR93] C. Carpineto and G. Romano. GALOIS: An order-theoretic approach to conceptual clustering. In *Proc. of the 10th Int. Conf. on Machine Learning*, pages 33–40, Amherst (MA), 1993.

[GM94] R.G. Godin and R. Missaoui. An incremental concept formation approach for learning from databases. *Theoretical Computer Science*, 133:387–419, 1994.

[GMA95] R. Godin, R. Missaoui, and H. Alaoui. Incremental concept formation algorithms based on Galois (concept) lattices. *Computational Intelligence*, 11(2):246–267, 1995.

[Sar96] J. Sarbo. Lattice embedding. In P.W. Eklund, Gerard Ellis, and Graham Mann, editors, *Conceptual Structures: Knowledge Representation as Interlingua (ICCS'96)*, volume 1115 of *Springer-Verlag*, pages 293–307, 1996.

[Wil82] R. Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. In I. Rival, editor, *Ordered sets*, pages 445–470. D. Reidel, Dordrecht-Boston, 1982.

[Wil83] R. Wille. Subdirect decomposition of concept lattices. *Algebra Universalis*, 17:275–287, 1983.