

A Sequent Based Logic for Coincidence Grids

Dave Barker-Plummer
Stanford University
Stanford, CA, 94305-4101, USA
dbp@csl.i.stanford.edu

Nik Swoboda
Universidad Politécnica de Madrid
Boadilla del Monte, Madrid, 28660, Spain
nswoboda@clip.dia.fi.upm.es

Abstract

Information is often represented in tabular format in everyday documents such as balance sheets, sales figures, and so on. Tables represent an interesting point in the spectrum of representation systems between pictures and sentences, since some aspect of tables are sentential or conventional in nature, while others are graphical. In this paper we describe the logic of a particular formalized tabular representation system, that of coincidence grids. Although less common than everyday tables, this system is recommended for use in the search for solution of so-called “Logic Puzzles”. Such puzzles provide a specific reasoning task in service of which the tabular representation is used.

1 Introduction

Representations appear to range along a spectrum from the highly conventional sentential representations to pictorial representations which are strongly isomorphic to the objects which they represent. The diagrams used in the **Hyperproof** program are “pictorial” in this sense, being pictures of a checkerboard on which blocks of various sizes and shapes are placed [3]. The diagrams introduced by Euler and Venn [4, 8] and studied by Shin in [6] and Hammer in [5] have some features of pictorial representations, but lack others. For example, in Euler diagrams, closed curves are used to represent sets, and the points within such curves represent the members of the represented sets. This is not a direct pictorial representation of a set in the way that **Hyperproof**’s blocks are, or could be, pictures of real objects.

The main body of this paper is concerned with a discussion of the logic of a particular tabular representation, one that we call coincidence grids. This representation is recommended for use in the solution of certain “logic puzzles” found some puzzle books. We have chosen to study this somewhat uncommon representation because the representation is used in service of particular reasoning problems, and these problems have clear cut structures. This representation uses graphical constraints in only a very simple way compared to representations like the Venn and Euler systems. The only graphical constraint which is exploited by the use of tabular representations is that

each cell of the table can contain exactly one value, and therefore if a value is already present in a cell then no other value can fill that role.

1.1 Outline

The methodology which we adopt is exactly that used by logicians investigating proof and model theories of the more familiar sentential logics, for example first order logic. In Section 3 we specify the syntax of the representation, and then, in Section 4 we present a model theory for the representation, that is a mapping from the representation to the world which allows us to assert that certain instances of the representation are accurate descriptions of the (real or imaginary) world. Next, in Section 5, we describe the inference rules which may be applied to instances of the representation to produce new instances. Finally, in Sections 6 and 7, we give proofs of soundness and completeness for the logic described.

2 Coincidence Grids

Coincidence grids are recommended in “logic puzzle” books as an aid to solving certain kinds of logic puzzles. Here is an example problem:

Sylvia and four other workers in Midsville were unemployed for a short time last year when they decided to change their occupations (one was a telephone operator) and undergo retraining for new jobs. The five are now happily re-employed (one is a mechanic). From the premises below, determine each worker’s first name, last name (one’s is Swanson), former job, and present job.

1. The five workers are Tom, the former welder, the present arcade manager, Ms. Cortez, and the present fitness instructor (who is not Mr. Bertram).
2. Ralph used to be a foreman.
3. Marie who is neither Cortez nor Monroe, used to be a secretary.
4. Mr. Hampton is now a mail carrier.
5. The programmer, who is not Erica, used to repair TVs.

Figure 1 shows an example coincidence grid used in the solution of this puzzle. Puzzles of this type involve the attributes possessed by individuals. Each individual, typically a person, is known to have a number of attributes: first name, current occupation, etc. The set of available values for these attributes is stated in the puzzle and these values are shared among the individuals exhaustively and exclusively. A solution to the puzzle is a statement of exactly which single and unique value for all of the attributes each individual has.

The main interest in this paper is in formalizing and characterizing the reasoning that may be performed using the coincidence grid representation system. In contrast, the main interest in logic puzzles is in the extraction of the correct information for display in the initial diagram. This is an essentially heterogeneous reasoning problem, of the form discussed in [1, 2, 7] but we do not discuss this feature of logic puzzles in this paper. If one were to imagine the sentential assumptions expressed in a formal logic, perhaps as formulae whose atomic subformulae are constrained to be equalities, then the puzzles would be quite trivial to solve, indicating that the real trick in these puzzles is an appropriate understanding of the subtleties of the semantics of natural language.

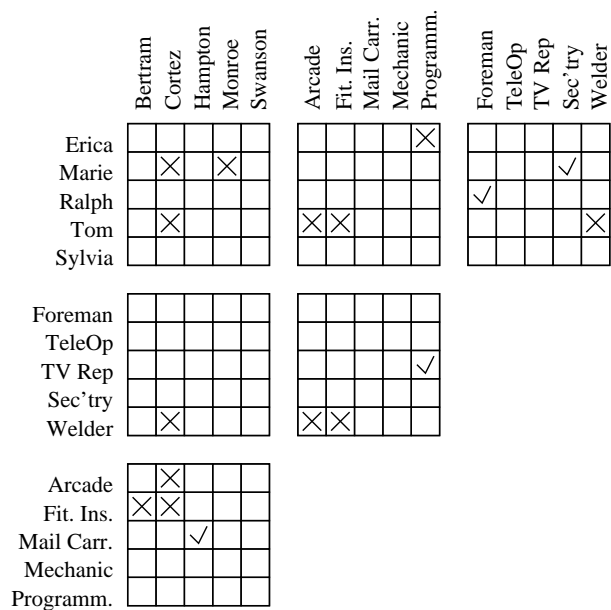


Figure 1: A coincidence grid diagram

There are two variable quantities in the coincidence grid, the number of attributes with which the problem is concerned, and the number of values each attribute may take on. Information concerning these problems comes in the form of assertions about the coincidence or non-coincidence of pairs of values for attributes. For example, the sentence “Ralph used to be the foreman” asserts that the object with first name “Ralph” also has the former occupation “foreman”. The diagram of Figure 1 consists of a number of individual grids, each of which have as their axes two of the attributes mentioned in the problem. For example the rightmost grid on the top row of the diagram has as its axes the “first name” and “former occupation” attributes. The diagram has a unique square grid for each pair of distinct attributes.

If we adopt the convention that a cell marked with a \checkmark which is in a particular row and column indicates that the object with the property labeling the row is the same as the object with the property labeling the column, and the same cell marked with a \times means that those same properties are known not to hold of the same object, then we can use the representation to indicate concisely certain assertions about the problem. For example, the \checkmark in the leftmost column of the grid representing the product of the “first name” and “former occupation” attributes, represents the assertion that “Ralph used to be a foreman” (hypothesis 2 of the example problem). While the \times in that same grid represents the assertion that “Tom is not the former welder”.

3 Syntax

We begin by defining the basic syntactic building blocks which will be used to construct coincidence grids:

- Grids - for all natural numbers n we will have a countably infinite number of $n \times n$ grids each consisting of n^2 cells. Each grid is taken to represent informa-

tion relating two attributes, each row or column represents information about an individual value, and each cell is taken to represent whether there is some object which has as values both those represented by that cell's row and column.

- Labels - a countably infinite number of labels (collected into the set \mathcal{L}) used to give names to the rows and columns of each grid.
- Marks - Cells of grids can be marked with either \checkmark or \times . \checkmark will be used in a cell when a single object is taken to have the pair of values of the cell's row and column labels, and \times will be used when the object does not have that value pair.

When referring to the rows and columns of a grid we will rely on the common understanding of these notions. For convenience we will make reference to cells of a grid using pairs of rows and columns, for this we will use a square bracket notation, e.g., $[r, c]$. When using this notation the order of r and c is irrelevant, i.e., $[r, c] = [c, r]$.

Definition 3.1 (Labeled Grid)

A **labeled grid** l of size n is $\langle g_l, rows_l, cols_l \rangle$ where g_l is a $n \times n$ grid and $rows_l, cols_l$ are one-to-one functions from the rows (resp. columns) of g_l to disjoint subsets of \mathcal{L} . We will refer to the ranges of $rows_l$ and $cols_l$ as **label sets**.

For convenience we define $labels_l = rows_l \cup cols_l$ ¹. Using the function $labels_l$ we can derive the partial function $cellFor_l$ from pairs of cell labels to cells (with cells represented by the intersection of a column and a row), $cellFor_l(i, j) = [a, b]$, when $labels_l(a) = i$ and $labels_l(b) = j$.² Finally we will sometimes refer to the cells of a labeled grid using the pairs of their column and row labels l, m , e.g., (l, m) where as before the order is irrelevant, i.e., $(l, m) = (m, l)$.

Definition 3.2 (Grid Layout)

A **grid layout** is a compatible collection of labeled grids. A collection of labeled grids, l_1, \dots, l_k all of size m , is said to be compatible when:

- No two labeled grids in the layout have the same set of labels (for each $i \neq j$, $labels_{l_i} \neq labels_{l_j}$). Note that this condition also excludes the inclusion of two labeled grids where the columns and rows are swapped.
- Row and column labels travel in packs, i.e., it isn't possible for the same label to appear in more than one distinct label set (for each i, j , $rows_{l_i}$ and $cols_{l_i}$ are each either equal to or disjoint from each of $rows_{l_j}$ and $cols_{l_j}$).
- Every pair of label sets from some labeled grid is represented by some labeled grid in the layout (for each distinct pair $rows_{l_i}, cols_{l_j}$ there is a grid l with $labels_l = rows_{l_i} \cup cols_{l_j}$).

We call a grid layout with n distinct row label collections, each of m labels, an n, m -grid layout.

When drawing a grid layout, we observe the convention that all grids sharing the same row label collections are drawn in series from right to left and with their row

¹Here and where convenient we will view functions as a sets of ordered pairs, and subject them to set operations.

²Here and throughout the remainder of the paper subscripts will be omitted when they can be unambiguously inferred from the context.

labels in the same order, and that all grids sharing the same set of column labels are drawn top to bottom with their columns in the same order. Thus the labels for the rows and columns of the grid layout can be placed along the top and left edges of the grid layout labeling rows and columns which span multiple grids.

The grid layout defines the tabular structure in which information may be represented. Information is represented by placing values into this structure. This is modeled using a marking function.

Definition 3.3 (Marking function)

A **marking function**, M , for the grid layout G is a (possibly partial) function from pairs of labels of the cells of G to the set $\{\checkmark, \times\}$. Given a cell in a row labeled r and column labeled s , $M(r, s) = \checkmark$ and $M(r, s) = \times$ are taken to mean that the referenced cell is marked with the corresponding symbol. $M(r, s)$ is undefined when a cell is unmarked. We will say that a marking function is **total** when it assigns either \checkmark or \times to all pairs of labels of cells in the grid layout.

Definition 3.4 (Coincidence grid)

A **coincidence grid**, (G, M) , is a grid layout G along with a marking function M for that grid layout.

4 Semantics

Coincidence grids are used to reason about information regarding the values for attributes of a set of objects. Coincidence Structures are mathematical objects which model this kind of information and thereby are used to give meaning to coincidence grids.

Definition 4.1 (Coincidence Structure)

An n, m -**coincidence structure** is a tuple $\langle \text{Partition}, \text{denotedBy} \rangle$ where *Partition* is a collection of n disjoint finite sets each with m elements and *denotedBy* is a one-to-one function from members of sets in *Partition* to labels in \mathcal{L} . We call the union of the sets in *Partition* the values of the structure.

If A is an coincidence structure, the relation coincides_A is defined so that $\text{coincides}_A(a, b)$ is true just when a and b are labels in \mathcal{L} and there are members j, k of the same set in Partition_A such that $\text{denotedBy}_A(j) = a$ and $\text{denotedBy}_A(k) = b$. It is trivial to show that coincides_A is an equivalence relation.

Definition 4.2 (\models)

Given a n, m -coincidence structure A and the coincidence grid (G, M) with G a n, m -grid layout, we write $A \models (G, M)$, and say that A is a model of (G, M) iff

- for each labeled grid g in G there are not two row labels l, l' such that $\text{coincides}(l, l')$ nor two column labels m, m' such that $\text{coincides}(m, m')$.
- for each cell in g in a row labeled l and a column labeled m marked with \checkmark , $\text{coincides}(l, m)$ is true.
- for each cell in g in a row labeled i and a column labeled j marked with \times , $\text{coincides}(i, j)$ is false.

Definition 4.3 ($\mathcal{C} \models \mathcal{C}'$)

A coincidence grid \mathcal{C}' is a **consequence** of a coincidence grid \mathcal{C} , $\mathcal{C} \models \mathcal{C}'$, if every model of \mathcal{C} is a model of \mathcal{C}' .

5 Proof System

The inference rules of the proof theory are given in sequent style below. When we reason with coincidence grids, the grid layout is fixed: inference proceeds by modifying the marking function, by adding or removing values at individual cells.

5.1 Erasure

The rule of ERASURE allows us to remove zero or more marks from a diagram.

$$\frac{}{(G, M) \rightsquigarrow (G, N)}$$

Proviso: $N \subseteq M$

Figure 2: Rule: ERASURE

The ERASURE rule does not require N to be a proper subset of M . When N and M are identical, we draw attention to this fact by referring to the rule as REITERATION.

5.2 Cases

Definition 5.1 (Extension of M at a cell)

Let M be a marking function which is undefined at (l, m) .

- $M_{(l,m)}^{\checkmark} = M \cup \{((l, m), \checkmark)\}$, i.e. $M_{(l,m)}^{\checkmark}$ is just like M except that $M_{(l,m)}^{\checkmark}$ assigns \checkmark to (l, m) .
- $M_{(l,m)}^{\times} = M \cup \{((l, m), \times)\}$, i.e. $M_{(l,m)}^{\times}$ is just like M except that $M_{(l,m)}^{\times}$ assigns \times to (l, m) .

The CASES rule allows us to examine the exclusive cases generated by extending a marking function at a single cell in both possible ways.

$$\frac{(G, M_{(l,m)}^{\checkmark}) \rightsquigarrow (G, N) \quad (G, M_{(l,m)}^{\times}) \rightsquigarrow (G, N)}{(G, M) \rightsquigarrow (G, N)}$$

Figure 3: Rule: CASES

5.3 Contradiction

We now need a crucial definition which defines the pathways that allow information to flow between grids. Intuitively, a cell in grid relates two values (a, b) to one another (perhaps by containing a check). There is a cell in another grid relating the one of these values to a third value (a, c) say. Yet a third cell relates (b, c) , and the values on these

three cells are dependent on one another (in particular, it isn't consistent for exactly two of them to contain a check and the other a cross.) We identify these cell collections as *triads*.

Definition 5.2 (Triad)

Given a coincidence grid C , a **triad** is three cells of C related by a chain of attribute pairs. Three cells $[x_1, y_1], [x_2, y_2], [x_3, y_3]$ are related by a chain of attribute pairs when there are labels t_1, t_2, t_3 in three labeled grids a, b, c in C , such that $cellFor_a(t_1, t_2) = [x_1, y_1]$, $cellFor_b(t_2, t_3) = [x_2, y_2]$, and $cellFor_c(t_1, t_3) = [x_3, y_3]$.

In Figure 1, an example of a triad would be the cells (TV Rep, Programm.), (Programm., Erica), (TV Rep, Erica). This example also shows the importance of triads, i.e., since we know that the former TV repair-person is currently employed as a programmer, and we know that Erica isn't currently a programmer we can conclude that Erica wasn't formerly employed as a TV repair-person. With the notion of triad in hand, we can now define a contradictory marking function for a diagram, which we will later show means that the diagram cannot represent a solution to the problem.

Definition 5.3 (Contradictory Marking Function)

A marking function M is **contradictory** if any of the following conditions hold:

1. M assigns two checks in the same row or column of a grid ($M(l_0, l_1) = \checkmark = M(l_0, l_2)$ for any l_0 from one label set and with l_1 and l_2 distinct values from a second label set).
2. M assigns one row or column of a grid to be completely filled by crosses ($M(l_0, l) = \times$ for any l_0 and for all l drawn from a second label set.)
3. M assigns a triad two checks and a cross ($M(l_0, l_1) = \checkmark = M(l_0, l_2)$ and $M(l_1, l_2) = \times$).

$$\frac{}{(G, M) \rightsquigarrow (G, N)}$$

Proviso: M is contradictory

Figure 4: Rule: CONTRADICTION

Definition 5.4 (\vdash)

For any coincidence grids (G, M) and (G, M') , we say that (G, M') can be proved from (G, M) , written $(G, M) \vdash (G, M')$, iff there is a tree of applications of the above rules of inference whose root contains $(G, M) \rightsquigarrow (G, M')$, and whose leaves are all closed (derived from no premises using ERASURE or CONTRADICTION).

A simple example proof is shown in Figure 5. Before leaving this discussion of the proof theory, we prove the following important result, which we will use later:

Proposition 5.1 For any sequent $(G, M) \rightsquigarrow (G, N)$, we can build a proof tree whose leaves contain all of the sequents $(G, M_i) \rightsquigarrow (G, N)$, where M_i is a total extension of M .

Proof The proof is by induction on the number of places that M is undefined.

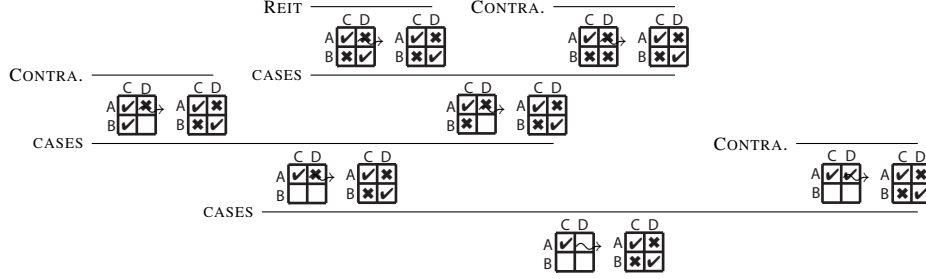


Figure 5: An Example Proof

Basis: If M is total, then we can build a proof with $(G, M) \rightsquigarrow (G, N)$ as the only leaf of a tree consisting of one application of the REITERATION rule.

Step: Assume that the result follows for all marking functions undefined at $i < k$ cells, we will show that the result follows for any marking function undefined at k cells.

A single application of CASES to the given sequent yields the following tree:

$$\text{CASES} \frac{(G, M_{(l,m)}^{\checkmark}) \rightsquigarrow (G, N) \quad (G, M_{(l,m)}^{\times}) \rightsquigarrow (G, N)}{(G, M) \rightsquigarrow (G, N)}$$

By the induction hypothesis, we can build a proof tree with the required property above each of the new sequents:

$$\text{CASES} \frac{\frac{\Pi}{(G, M_{(l,m)}^{\checkmark}) \rightsquigarrow (G, N)} \quad \frac{\Sigma}{(G, M_{(l,m)}^{\times}) \rightsquigarrow (G, N)}}{(G, M) \rightsquigarrow (G, N)}$$

The proof Π has at its leaves all of the sequents which are total extensions of $M_{(l,m)}^{\checkmark}$, and Σ 's leaves are the total extensions of $M_{(l,m)}^{\times}$, so together the leaves have all total extensions of (G, M) on their left hand sides (and (G, N) on the right).

6 Soundness

Theorem 6.1 (Soundness) For any coincidence grids (G, M) and (G, M') , if $(G, M) \vdash (G, M')$ then $(G, M) \models (G, M')$

Proof The proof is by induction on the height of the proof tree.

Basis If the height of the tree is 1, then the proof must involve one application of CONTRADICTION or ERASURE.

- **ERASURE:** Suppose for the sake of contradiction that $(G, M) \not\models (G, M')$. There exists some model, A such that $A \models (G, M)$ and $A \not\models (G, M')$. Since both coincidence grids contain the same labeled grid G we know that $A \not\models (G, M')$ can not be due an incompatibility between label sets in G and the coincides relation. Thus there is some cell (l, m) such that it is marked with a check in M' and $\text{coincides}(l, m)$ is false, or it is marked with a cross in M' and $\text{coincides}(l, m)$ is true. In either case, this mark was present in M , and A therefore fails to be a model of (G, M) . Contradiction.

- **Contradiction:** *It suffices to show that any coincidence grid to which CONTRADICTION can be applied has no models.*

Suppose for sake of contradiction that (G, M) is an coincidence grid of size n to which CONTRADICTION applies, and that $A \models (G, M)$. There are three cases depending on which of the clauses of CONTRADICTION applied.

1. *Some grid of (G, M) contains a row labeled l with two check marks, in the columns labeled m and m' say. This means that both $\text{coincides}(l, m)$ and $\text{coincides}(l, m')$ are true, and hence that $\text{coincides}(m, m')$ is also true, which means that $A \not\models (G, M)$, contradiction. The case of two checks in the same column is analogous.*
2. *Some grid of (G, M) contains a row $[l, m_1], \dots, [l, m_n]$ with all crosses. This means that no row label m_i is mapped to the same member of A as l is. But this is impossible since there are n members of A , n of the m_i , and each m_i is mapped to a different member of A . Contradiction. The case of a grid with a column with all crosses is analogous.*
3. *Some triad in (G, M) contains two checks and a cross, i.e., $M(l_0, l_1) = \checkmark = M(l_0, l_2)$ and $M(l_1, l_2) = \times$. Since $M(l_0, l_1) = \checkmark = M(l_0, l_2)$, $\text{coincide}(l_0, l_1)$ and $\text{coincide}(l_0, l_2)$ are derived from A . Since coincide is transitive, it follows that $\text{coincide}(l_1, l_2)$. But since $A \models (G, M)$ and $M(l_1, l_2) = \times$, this cannot be true. Contradiction.*

Step *Suppose that the height of the proof tree is $k > 1$ and that the result holds for all proofs of length less than k .*

The inference at the root must be an application of CASES on some cell with labels l and m . We must show that $(G, M) \models (G, M')$. Suppose not, i.e. that there is some model A of (G, M) which is not a model of (G, M') . In every model $\text{coincides}(l, m)$ is either true or false, and so either A is a model of $(G, M_{(l,m)}^\checkmark)$ or of $(G, M_{(l,m)}^\times)$. But we know that $(G, M_{(l,m)}^\checkmark) \models (G, M')$ and that $(G, M_{(l,m)}^\times) \not\models (G, M')$ from the induction hypothesis, which is a contradiction.

7 Completeness

In this section we will demonstrate the completeness of the proof system, i.e., that it is sufficiently powerful to allow that any logical consequence of a coincidence grid can be proved to be so. To show this result, we will present a generic strategy (Algorithm 7.1) for building a proof with any coincidence grid \mathcal{C} as the premise and any coincidence grid \mathcal{C}' as the conclusion. Furthermore we will show that if this strategy fails then that \mathcal{C}' can not be a logical consequence of \mathcal{C} .

The strategy proceeds as follows. First, we build a proof tree with $\mathcal{C} \rightsquigarrow \mathcal{C}'$ as the root and a leaf $\mathcal{C}_i \rightsquigarrow \mathcal{C}'$ for each total extension of \mathcal{C} . If there is any leaf which can not be closed, established using either the ERASURE or CONTRADICTION rules with no premise then the strategy fails.

Algorithm 7.1 (Canonical Proof Strategy)

We begin with any two coincidence grids (G, M) (the premise) and (G, M') (the conclusion).

Step #1: *Start the proof with the sequent $(G, M) \rightsquigarrow (G, M')$ as the root.*

	Bertram	Cortez	Hampton	Monroe	Swanson
Erica					
Marie					
Ralph			×	×	×
Tom			×	×	×
Sylvia			×	×	×

Figure 6: A coincidence grid with no models to which CONTRADICTION can not be applied

Step #2: While there exist unmarked cells in the left hand side of the sequent of some leaf of the proof: select at random any unmarked cell (l, m) in (G, M'') of each leaf $(G, M'') \rightsquigarrow (G, M')$ and apply the CASES rule to add the branches $(G, M'_{(l,m)}) \rightsquigarrow (G, M')$ and $(G, M''_{(l,m)}) \rightsquigarrow (G, M')$ to the proof above that leaf.

Step #3: Close each leaf with an application of the ERASURE or the CONTRADICTION rule with no premise. If this is not possible for any leaf then fail, otherwise the proof is complete.

To prove that this strategy is correct (that any logical consequence of a diagram can be proved in this manner) we first consider the case that \mathcal{C} has no models, and show that if a diagram has no models then the contradiction rule can be applied to every total extension of \mathcal{C} (Proposition 7.2). Then we need to consider the case where the proof has a leaf $\mathcal{C}_i \rightsquigarrow \mathcal{C}'$ which can not be established using either the ERASURE or CONTRADICTION rules. First we show that any total non-contradictory coincidence grid is true in some model (Proposition 7.1). Then we know that \mathcal{C}_i must have a model, and that \mathcal{C}_i and \mathcal{C}' differ on some cell. Using this diagram we build a model A such that $A \models \mathcal{C}$ but that $A \not\models \mathcal{C}'$ (Proposition 7.4), which means that it can't be the case that $\mathcal{C} \models \mathcal{C}'$.

We begin by observing that there are some coincidence grids which have no models, but to which CONTRADICTION can not be applied. An example of one such coincidence grid can be found in Figure 6. We need to show that every total extension of such a marking function does permit the application of CONTRADICTION.

Proposition 7.1 All coincidence grid (G, M) with G a total marking function which is not contradictory are true in a model.

Proof We construct a model of A of (G, M) . For each label l in G , determine $\{m \mid M(l, m) = \checkmark\} \cup \{l\}$. There is exactly one m along each dimension that has this property, since CONTRADICTION does not apply to (G, M) . Let Properties be the collection of these sets. (In other words, we use the labels themselves as the values of $\langle \text{Properties}, \text{denotedBy} \rangle$.) As the function denotedBy we use the identity function on the labels of G . $\langle \text{Properties}, \text{denotedBy} \rangle$ is a model of (G, M) .

Proposition 7.2 If (G, M) has no models, then CONTRADICTION can be applied to every total extension of (G, M) .

Proof We prove the contrapositive. Let T be a total extension of M to which CON-

TRADITION does not apply. Using Proposition 7.1, we know that (G, T) has a model, and since T is an extension of M we know that this model is also a model of (G, M) .

Proposition 7.3 *If (G, M) has no models, then $(G, M) \vdash (G, M')$ for any marking function M' .*

Proof By Proposition 5.1 there is a proof tree Π whose root contains $(G, M) \rightsquigarrow (G, M')$ and whose leaves contain the sequents $(G, M_i) \rightsquigarrow (G, M')$ for all total extensions, M_i of M . By Proposition 7.2, the contradiction rule can be applied to each leaf of this tree, and so $(G, M) \vdash (G, M')$.

Proposition 7.4 *Given coincidence grids (G, M) which has a model and (G, M') , such that $M(l, m) = \checkmark$ and $M'(l, m) = \times$ for some l, m , then no model A such that $A \models (G, M)$, can be a model of (G, M') .*

Proof Since we know that in (G, M) , $M(l, m) = \checkmark$ we also have that $\text{coincides}_A(l, m)$ can be derived from all models A in which (G, M) is true. However in any model A' which makes (G, M') true, $\text{coincides}_{A'}(l, m)$ can not hold (since $M'(l, m) = \times$), so there can be no model A such that $A \models (G, M)$ and $A \models (G, M')$.

Lemma 7.1 *For all coincidence grid (G, M) and (G, M') , Algorithm 7.1 finds a proof $(G, M) \vdash (G, M')$ iff $(G, M) \models (G, M')$.*

Proof First we consider the possibility that (G, M) has no models. In this case all coincidence grid are logical consequences of (G, M) so the algorithm should generate a valid proof for any coincidence grid (G, M') (see Proposition 7.3). Using Proposition 7.2 we know that the CONTRADICTION rule can be applied to each leaf generated by the algorithm and thus that a valid proof is generated.

Now we consider the case that (G, M) has a model. If the algorithm generates a proof then from soundness we know that $(G, M) \models (G, M')$. If the algorithm fails then we need to show that $(G, M) \not\models (G, M')$. Assuming that the algorithm fails we know that there was some leaf $(G, M_i) \rightsquigarrow (G, M')$ generated by the strategy which could not be closed, established by the ERASURE or CONTRADICTION rules without a premise. From the fact that the CONTRADICTION rule can not be applied to that sequent and Proposition 7.1 we know that (G, M_i) has some model which we will call A . Furthermore from the fact that the ERASURE rule can not be applied and that M_i is total, we know that M_i and M' disagree on the content of some cell. Thus from Proposition 7.4 we know that no model of (G, M_i) can be a model of (G, M') , i.e., $A \not\models (G, M')$. Since (G, M_i) is an extension of (G, M) we know that $A \models (G, M)$ and finally that $(G, M) \not\models (G, M')$.

Theorem 7.1 (Completeness)

For any two coincidence grids (G, M) and (G, M') , if $(G, M) \models (G, M')$ then $(G, M) \vdash (G, M')$.

Proof The proof of this theorem is a direct result of Lemma 7.1.

Corollary 7.1 *For any two coincidence grid $\mathcal{C}, \mathcal{C}'$ the question of whether \mathcal{C}' is a logical consequence of \mathcal{C} is decidable.*

Proof Since there are a finite number of unmarked cells in any coincidence grid we know that Algorithm 7.1 will terminate (though possibly in exponential time), the rest is a direct result of Lemma 7.1.

8 Conclusion

In this paper we have formalized a logic of coincidence grids, by defining the syntax proof theory and semantics for the representation. We have shown that the proof theory is both sound and complete for the semantics.

The logic that we have defined permits a search for proofs based on a “generate and test” method. The CASES rule allows us to mark a previously unmarked cell and then to determine the consequences of each possible mark. While having the twin virtues of soundness and completeness, this is not a natural logic for proof search using these representations. There are more intuitive rules which allow inference between diagrams, for example a rule which allows the addition of a \times to any unmarked cell in a row or column that already contains a \checkmark . This rule, and others like it, are easily definable using the inference rules presented here however we have not yet found a way to define a collection of rules which allow us to dispense with CASES entirely. This is due to the fact that we have yet to find a natural contradiction rule that can be applied to recognize all diagrams that have no models. Finding that stronger version of contradiction and a set of natural inference rules which would allow us to dispense with CASES is the subject of future work.

References

- [1] Dave Barker-Plummer and John Etchemendy. Visual decision making: A computational architecture for heterogeneous reasoning. In Boris Kovalerchuk and James Schwing, editors, *Visual and Spatial Analysis: Advances in Data Mining, Reasoning and Problem Solving*, pages 79–109. Springer, 2004.
- [2] Dave Barker-Plummer and John Etchemendy. A computational architecture for heterogeneous reasoning. *Journal of Theoretical and Experimental Artificial Intelligence*, to appear.
- [3] Jon Barwise and John Etchemendy. *Hyperproof*. CSLI Press, 1994. (ISBN: 1-881526-11-9).
- [4] Leonhard Euler. *Lettres à une Princesse d’Allemagne sur Divers Sujets de Physique et de Philosophie*. Imprimerie de l’Academie Impériale des Sciences, 1768–1772. Letters 102–108.
- [5] Eric Hammer. *Logic and Visual Information*. Number 3 in Studies in Logic, Language and Information. CSLI Publications, 1995.
- [6] Sun-Joo Shin. A Situation-Theoretic Account of Valid Reasoning with Venn Diagrams. In J. Barwise, J.M. Gawron, G. Plotkin, and S. Tutiya, editors, *Situation Theory and its Applications*, number 26 in CSLI Lecture Notes, pages 581–605. CSLI Press, 1991.
- [7] Nik Swoboda and Gerard Allwein. Modeling heterogeneous systems. In Mary Hegarty, Bernd Meyer, and N. Hari Narayanan, editors, *Diagrammatic Representation and Inference*, number 2317 in Lecture Notes in Artificial Intelligence, pages 131–145. Springer-Verlag, Berlin, 2002.
- [8] John Venn. *Symbolic Logic*. Burt Franklin, New York, 1971.