9 Inclusion Dependency

Vittorio: *Fd's and jd's give some structure to relations.*
Alice: *But there are no connections between them.*
Sergio: *Making connections is the next step ...*
Riccardo: *... with some unexpected consequences.*

The story of inclusion dependencies starts in a manner similar to that for functional dependencies: Implication is decidable (although here it is PSPACE-complete), and there is a simple set of inference rules that is sound and complete. But the story becomes much more intriguing when functional and inclusion dependencies are taken together. First, the notion of logical implication will have to be refined because the behavior of these dependencies taken together is different depending on whether infinite instances are permitted. Second, both notions of logical implication are nonrecursive. And third, it can be proven in a formal sense that no "finite" axiomatization exists for either notion of logical implication of the dependencies taken together. At the end of this chapter, two restricted classes of inclusion dependencies are discussed. These are significant because they arise in modeling certain natural relationships such as those encountered in semantic data models. Positive results have been obtained for inclusion dependencies from these restricted classes considered with fd's and other dependencies.

Unlike fd's or jd's, a single inclusion dependency may refer to more than one relation. Also unlike fd's and jd's, inclusion dependencies are "untyped" in the sense that they may call for the comparison of values from columns (of the same or different relations) that are labeled by different attributes. A final important difference from fd's and jd's is that inclusion dependencies are "embedded." Speaking intuitively, to satisfy an inclusion dependency the presence of one tuple in an instance may call for the presence of another tuple, of which only some coordinate values are determined by the dependency and the first tuple. These and other differences will be discussed further in Chapter 10.

9.1 Inclusion Dependency in Isolation

To accommodate the fact that inclusion dependencies permit the comparison of values from different columns of one or more relations, we introduce the following notation. Let R be a relation schema and $X = A_1, \ldots, A_n$ a *sequence* of attributes (possibly with repeats) from R. For an instance I of R, the *projection* of I onto the sequence X, denoted $I[X]$, is the n-ary relation $\{\langle t(A_1), \ldots, t(A_n)\rangle \mid t \in I\}$.

The syntax and semantics of inclusion dependencies is now given by the following:

Definition 9.1.1 Let **R** be a relational schema. An *inclusion dependency* (ind) over **R** is an expression of the form $\sigma = R[A_1, \ldots, A_m] \subseteq S[B_1, \ldots, B_m]$, where

(a) R, S are (possibly identical) relation names in **R**,

(b) $A_1, \ldots, A_m$ is a sequence of distinct attributes of $sort(R)$, and

(c) $B_1, \ldots, B_m$ is a sequence of distinct attributes of $sort(S)$.

An instance **I** of **R** *satisfies* σ, denoted $\mathbf{I} \models \sigma$, if

$$\mathbf{I}(R)[A_1, \ldots, A_m] \subseteq \mathbf{I}(S)[B_1, \ldots, B_m].$$

Satisfaction of a set of ind's is defined in the natural manner.

To illustrate this definition, we recall an example from the previous chapter.

Example 9.1.2 There are two relations: *Movies* with attributes *Title, Director, Actor* and *Showings* with *Theater, Screen, Title, Snack*; and we have an ind

$$Showings[Title] \subseteq Movies[Title].$$

The generalization of ind's to permit repeated attributes on the left-or right-hand side is considered in Exercise 9.4.

The notion of *logical implication* between sets of ind's is defined in analogy with that for fd's. (This will be refined later when fd's and ind's are considered together.)

Rules for Inferring ind Implication

The following set of inference rules will be shown sound and complete for inferring logical implication between sets of ind's. The variables X, Y, and Z range over sequences of distinct attributes; and R, S, and T range over relation names.

IND1: (reflexivity) $R[X] \subseteq R[X]$.

IND2: (projection and permutation) If $R[A_1, \ldots, A_m] \subseteq S[B_1, \ldots, B_m]$, then $R[A_{i_1}, \ldots, A_{i_k}] \subseteq S[B_{i_1}, \ldots, B_{i_k}]$ for each sequence $i_1, \ldots, i_k$ of distinct integers in $\{1, \ldots, m\}$.

IND3: (transitivity) If $R[X] \subseteq S[Y]$ and $S[Y] \subseteq T[Z]$, then $R[X] \subseteq T[Z]$.

The notions of proof and of provability (denoted $\vdash$) using these rules are defined in analogy with that for fd's.

Theorem 9.1.3 The set {IND1, IND2, IND3} is sound and complete for logical implication of ind's.

Proof Soundness of the rules is easily verified. For completeness, let Σ be a set of ind's over database schema $\mathbf{R} = \{R_1, \ldots, R_n\}$, and let $\sigma = R_a[A_1, \ldots, A_m] \subseteq R_b[B_1, \ldots, B_m]$

be an ind over $\mathbf{R}$ such that $\Sigma \models \sigma$. We construct an instance $\mathbf{I}$ of $\mathbf{R}$ and use it to demonstrate that $\Sigma \vdash \sigma$.

To begin, let s' be the tuple over R_a such that $s'(A_i) = i$ for $i \in [1, m]$ and $s'(B) = 0$ otherwise. Set $\mathbf{I}(R_a) = \{s'\}$ and $\mathbf{I}(R_j) = \emptyset$ for $j \neq a$. We now apply the following rule to $\mathbf{I}$ until it can no longer be applied.

$$(*)\qquad \begin{array}{l}\text{If } R_i[C_1, \ldots, C_k] \subseteq R_j[D_1, \ldots, D_k] \in \Sigma \text{ and } t \in \mathbf{I}(R_i), \text{ then add} \\ u \text{ to } \mathbf{I}(R_j), \text{ where } u(D_l) = t(C_l) \text{ for } l \in [1, k] \text{ and } u(D) = 0 \text{ for } D \\ \qquad \notin \{D_1, \ldots, D_k\}.\end{array}$$

Application of this rule will surely terminate, because all tuples are constructed from a set of at most $m + 1$ values. Clearly the result of applying this rule until termination is unique, so let $\mathbf{J}$ be this result.

Remark 9.1.4 This construction is reminiscent of the chase for join dependencies. It differs because the ind's may be embedded. Intuitively, an ind may not specify all the entries of the tuples we are adding. In the preceding rule $(*)$, the same value (0) is always used for tuple entries that are otherwise unspecified. ■

It is easily seen that $\mathbf{J} \models \Sigma$. Because $\Sigma \models \sigma$, we have $\mathbf{J} \models \sigma$. To conclude the proof, we show the following:

$$(**)\qquad \begin{array}{l}\text{If for some } R_j \text{ in } \mathbf{R}, u \in \mathbf{J}(R_j), \text{ integer } q, \text{ and distinct attributes} \\ C_1, \ldots, C_q \text{ in } \mathit{sort}(R_j), u(C_p) > 0 \text{ for } p \in [1, q], \text{ then} \\ \qquad \Sigma \vdash R_a[A_{u(C_1)}, \ldots, A_{u(C_q)}] \subseteq R_j[C_1, \ldots, C_q].\end{array}$$

Suppose that $(**)$ holds. Let s'' be a tuple of $\mathbf{J}(R_b)$ such that $s''[B_1, \ldots, B_m] = s'[A_1, \ldots, A_m]$. (Such a tuple exists because $\mathbf{J} \models \sigma$.) Use $(**)$ with $R_j = R_b$, $q = m$, $C_1, \ldots, C_q = B_1, \ldots, B_m$.

To demonstrate $(**)$, we show inductively that it holds for all tuples of $\mathbf{J}$ by considering them in the order in which they were inserted. The claim holds for s in $\mathbf{J}(R_a)$ by IND1. Suppose now that

- $\mathbf{I}'$ is the instance obtained after k applications of the rule for some $k \geq 0$;
- the claim holds for all tuples in $\mathbf{I}'$; and
- u is added to R_j by the next application of rule $(*)$, due to the ind $R_i[C_1, \ldots, C_k] \subseteq R_j[D_1, \ldots, D_k] \in \Sigma$ and tuple $t \in \mathbf{I}'(R_i)$.

Now let $\{E_1, \ldots, E_q\}$ be a set of distinct attributes in $\mathit{sort}(R_j)$ with $u(E_p) > 0$ for $p \in [1, q]$. By the construction of u in (*), $\{E_1, \ldots, E_q\} \subseteq \{D_1, \ldots, D_k\}$. Choose the mapping ρ such that $D_{\rho(p)} = E_p$ for $p \in [1, q]$. Because $R_i[C_1, \ldots, C_k] \subseteq R_j[D_1, \ldots, D_k] \in \Sigma$, IND2 yields

$$\Sigma \vdash R_i[C_{\rho(1)}, \ldots, C_{\rho(q)}] \subseteq R_j[E_1, \ldots, E_q].$$

By the inductive assumption,

$$\Sigma \vdash R_a[A_{t(C_{\rho(1)})}, \ldots, A_{t(C_{\rho(q)})}] \subseteq R_i[C_{\rho(1)}, \ldots, C_{\rho(q)}].$$

Thus, by IND3,

$$\Sigma \vdash R_a[A_{t(C_{\rho(1)})}, \ldots, A_{t(C_{\rho(q)})}] \subseteq R_j[E_1, \ldots, E_q].$$

Finally, observe that for each p, $t(C_{\rho(p)}) = u(D_{\rho(p)}) = u(E_p)$, so

$$\Sigma \vdash R_a[A_{u(E_1)}, \ldots, A_{u(E_q)}] \subseteq R_j[E_1, \ldots, E_q].$$

Deciding Logical Implication for ind's

The proof of Theorem 9.1.3 yields a decision procedure for determining logical implication between ind's. To see this, we use the following result:

Proposition 9.1.5 Let Σ be a set of ind's over **R** and $R_a[A_1, \ldots, A_m] \subseteq R_b[B_1, \ldots, B_m]$. Then $\Sigma \models R_a[A_1, \ldots, A_m] \subseteq R_b[B_1, \ldots, B_m]$ iff there is a sequence $R_{i_1}[\vec{C}_1], \ldots, R_{i_k}[\vec{C}_k]$ such that

(a) $R_{i_j} \in \mathbf{R}$ for $j \in [1, k]$;
(b) $\vec{C}_j$ is a sequence of m distinct attributes in $sort(R_{i_j})$ for $j \in [1, k]$;
(c) $R_{i_1}[\vec{C}_1] = R_a[A_1, \ldots, A_m]$;
(d) $R_{i_k}[\vec{C}_k] = R_b[B_1, \ldots, B_m]$;
(e) $R_{i_j}[\vec{C}_j] \subseteq R_{i_{j+1}}[\vec{C}_{j+1}]$ can be obtained from an ind in Σ by one application of rule IND2, for $j \in [1, (k-1)]$.

Crux Use the instance **J** constructed in the proof of Theorem 9.1.3. Working backward from the tuple s'' in $\mathbf{J}(R_b)$, a chain of relation-tuple pairs (R_{i_j}, s_j) can be constructed so that each of $1, \ldots, m$ occurs exactly once in s_j, and s_{j+1} is inserted into **I** as a result of s_j and IND2. ■

Based on this, it is straightforward to verify that the following algorithm determines logical implication between ind's. Note that only ind's of arity m are considered in the algorithm.

Algorithm 9.1.6

Input: A set Σ of ind's over **R** and ind $R_a[A_1, \ldots, A_m] \subseteq R_b[B_1, \ldots, B_m]$.

Output: Determine whether $\Sigma \models R_a[A_1, \ldots, A_m] \subseteq R_b[B_1, \ldots, B_m]$.

Procedure: Build a set $\mathcal{E}$ of expressions of the form $R_i[C_1, \ldots, C_m]$ as follows:

1. $\mathcal{E} := \{R_a(A_1, \ldots, A_m)\}$.

2. Repeat until $R_b[B_1, \ldots, B_m] \in \mathcal{E}$ or no change possible:
   If $R_i[C_1, \ldots, C_m] \in \mathcal{E}$ and
   $$R_i[C_1, \ldots, C_m] \subseteq R_j[D_1, \ldots, D_m]$$
   can be derived from an ind of Σ by one application of IND2, then insert $R_j[D_1, \ldots, D_m]$ into $\mathcal{E}$.
3. If $R_b[B_1, \ldots, B_m] \in \mathcal{E}$ then return *yes*; else return *no*. ■

As presented, the preceding algorithm is nondeterministic and might therefore take more than polynomial time. The following result shows that this is indeed likely for any algorithm for deciding implication between ind's.

Theorem 9.1.7 Deciding logical implication for ind's is PSPACE-complete.

Crux Algorithm 9.1.6 can be used to develop a nondeterministic polynomial space procedure for deciding logical implication between ind's. By Savitch's theorem (which states that PSPACE = NPSPACE), this can be transformed into a deterministic algorithm that runs in polynomial space. To show that the problem is PSPACE-hard, we describe a reduction from the problem of linear space acceptance.

A (Turing) machine is *linear bounded* if on each input of size n, the machine does not use more that n tape cells. The problem is the following:

Linear Space Acceptance (LSA) problem

Input: The description of a linear bounded machine M and an input word x;

Output: yes iff M accepts x.

The heart of the proof is, given an instance (M, x) of the LSA problem, to construct a set Σ of ind's and an ind σ such that $\Sigma \models \sigma$ iff x is accepted by M.

Let $M = (K, \Gamma, \Delta, s, h)$ be a Turing machine with states K, alphabet Γ, transition relation Δ, start state s, and accepting state h; and let $x = x_1 \ldots x_n \in \Gamma^*$ have length n.

Configurations of M are viewed as elements of $\Gamma^* K \Gamma^+$ with length $n + 1$, where the placement of the state indicates the head position (the state is listed immediately left of the scanned letter). Observe that transitions can be described by expressions of the form $\alpha_1, \alpha_2, \alpha_3 \to \beta_1, \beta_2, \beta_3$ with $\alpha_1, \ldots, \beta_3$ in $(K \cup \Gamma)$. For instance, the transition

"if reading b in state p, then overwrite with c and move left"

corresponds to $a, p, b \to p, a, c$ for each a in Γ. Let χ be the set of all such expressions corresponding to transitions of M.

The initial configuration is sx. The final configuration is $h\,\not{b}^n$ for some particular letter $\not{b}$, iff M accepts x.

The ind's of Σ are defined over a single relation R. The attributes of R are $\{A_{i,j} \mid i \in (K \cup \Gamma),\ j \in \{1, 2, \ldots, n+1\}\}$. The intuition here is that the attribute $A_{p,j}$ corresponds to the statement that the j^{th} symbol in a given configuration is p. To simplify the presentation, attribute $A_{a,k}$ is simply denoted by the pair (a, k).

The ind σ is

$$R[(s,1),(x_1,2),\ldots,(x_n,n+1)] \subseteq R[(h,1),(\not{b},2),\ldots,(\not{b},n+1)].$$

The ind's in Σ correspond to valid moves of M. In particular, for each $j \in [1, n-1]$, Σ includes all ind's of the form

$$R[(\alpha_1,j),(\alpha_2,j+1),(\alpha_3,j+2),\vec{A}\,] \subseteq R[(\beta_1,j),(\beta_2,j+1),(\beta_3,j+2),\vec{A}\,],$$

where $\alpha_1,\alpha_2,\alpha_3 \to \beta_1,\beta_2,\beta_3$ is in χ, and $\vec{A}$ is an arbitrary fixed sequence that lists all of the attributes in $\Gamma \times \{1,\ldots,j-1,j+3,\ldots,n+1\}$. Thus each ind in Σ has arity $3+(n-2)|\Gamma|$, and $|\Sigma| \leq n|\Delta|$.

Although the choice of $\vec{A}$ permits the introduction of many ind's, observe that the construction is still polynomial in the size of the linear space automaton problem (M, x). Using Proposition 9.1.5, it is now straightforward to verify that $\Sigma \models \sigma$ iff M has an accepting computation of x. ■

Although the general problem of deciding implication for ind's is PSPACE-complete, naturally arising special cases of the problem have polynomial time solutions. This includes the family of ind's that are at most k-ary (ones in which the sequences of attributes have length at most some fixed k) and ind's that have the form $R[\vec{A}\,] \subseteq S[\vec{A}\,]$ (see Exercise 9.10). The latter case arises in examples such as $Grad-Stud[Name, Major] \subseteq Student[Name, Major]$. This theme is also examined at the end of this chapter.

9.2 Finite versus Infinite Implication

We now turn to the interaction between ind's and fd's, which leads to three interesting phenomena. The first of these requires a closer look at the notion of logical implication.

Consider the notion of logical implication used until now: Σ logically implies σ if for all relation (or database) instances $\mathbf{I}$, $\mathbf{I} \models \Sigma$ implies $\mathbf{I} \models \sigma$. Although this notion is close to the corresponding notion of mathematical logic, it is different in a crucial way: In the context of databases considered until now, only *finite* instances are considered. From the point of view of logic, the study of logical implication conducted so far lies within *finite model theory*.

It is also interesting to consider logical implication in the traditional mathematical logic framework in which infinite database instances are permitted. As will be seen shortly, when fd's or ind's are considered separately, permitting infinite instances has no impact on logical implication. However, when fd's and ind's are taken together, the two flavors of logical implication do not coincide.

The notion of infinite relation and database instances is defined in the natural manner. An *unrestricted* relation (database) instance is a relation (database) instance that is either finite or infinite. Based on this, we now redefine "unrestricted implication" to permit infinite instances, and we define "finite logical implication" for the case in which only finite instances are considered.

R	A	B
	1	0
	2	1
	3	2
	4	3
	$\vdots$	$\vdots$

(a)

R	A	B
	1	1
	2	1
	3	2
	4	3
	$\vdots$	$\vdots$

(b)

Figure 9.1: Instances used for distinguishing $\models_{\text{fin}}$ and $\models_{\text{unr}}$

Definition 9.2.1 A set Σ of dependencies over $\mathbf{R}$ *implies without restriction* a dependency σ, denoted $\Sigma \models_{\text{unr}} \sigma$, if for each unrestricted instance $\mathbf{I}$ of $\mathbf{R}$, $\mathbf{I} \models \Sigma$ implies $\mathbf{I} \models \sigma$. A set Σ of dependencies over $\mathbf{R}$ *finitely implies* a dependency σ, denoted $\Sigma \models_{\text{fin}} \sigma$, if for each (finite) instance $\mathbf{I}$ of $\mathbf{R}$, $\mathbf{I} \models \Sigma$ implies $\mathbf{I} \models \sigma$.

If finite and unrestricted implication coincide, or if the kind of implication is understood from the context, then we may use $\models$ rather than $\models_{\text{fin}}$ or $\models_{\text{unr}}$. This is what we implicitly did so far by using $\models$ in place of $\models_{\text{fin}}$.

Of course, if $\Sigma \models_{\text{unr}} \sigma$, then $\Sigma \models_{\text{fin}} \sigma$. The following shows that the converse need not hold:

Theorem 9.2.2

(a) There is a set Σ of fd's and ind's and an ind σ such that $\Sigma \models_{\text{fin}} \sigma$ but $\Sigma \not\models_{\text{unr}} \sigma$.

(b) There is a set Σ of fd's and ind's and an fd σ such that $\Sigma \models_{\text{fin}} \sigma$ but $\Sigma \not\models_{\text{unr}} \sigma$.

Proof For part (a), let R be binary with attributes A, B; let $\Sigma = \{A \to B, R[A] \subseteq R[B]\}$; and let σ be $R[B] \subseteq R[A]$. To see that $\Sigma \models_{\text{fin}} \sigma$, let I be a finite instance of R that satisfies Σ. Because $I \models A \to B$, $|\pi_A(I)| \geq |\pi_B(I)|$ and because $I \models R[A] \subseteq R[B]$, $|\pi_B(I)| \geq |\pi_A(I)|$. It follows that $|\pi_A(I)| = |\pi_B(I)|$. Because I is finite and $\pi_A(I) \subseteq \pi_B(I)$, it follows that $\pi_B(I) \subseteq \pi_A(I)$ and $I \models R[B] \subseteq R[A]$.

On the other hand, the instance shown in Fig. 9.1(a) demonstrates that $\Sigma \not\models_{\text{unr}} \sigma$.

For part (b), let Σ be as before, and let σ be the fd $B \to A$. As before, if $I \models \Sigma$, then $|\pi_A(I)| = |\pi_B(I)|$. Because $I \models A \to B$, each tuple in I has a distinct A-value. Thus the number of B-values occurring in I equals the number of tuples in I. Because I is finite, this implies that $I \models B \to A$. Thus $\Sigma \models_{\text{fin}} \sigma$. On the other hand, the instance shown in Fig. 9.1(b) demonstrates that $\Sigma \not\models_{\text{unr}} \sigma$. ■

It is now natural to reconsider implication for fd's, jd's, and inds taken separately and in combinations. Are unrestricted and finite implication different in these cases? The answer is given by the following:

THEOREM 9.2.3 Unrestricted and finite implication coincide for fd's and jd's considered separately or together and for ind's considered alone.

Proof Unrestricted implication implies finite implication by definition. For fd's and jd's taken separately or together, Theorem 8.4.12 on the relationship between chasing and logical implication can be used to obtain the opposite implication. For ind's, Theorem 9.1.3 shows that finite implication and provability by the ind inference rules are equivalent. It is easily verified that these rules are also sound for unrestricted implication. Thus finite implication implies unrestricted implication for ind's as well. ■

The notion of finite versus unrestricted implication will be revisited in Chapter 10, where dependencies are recast into a logic-based formalism.

Implication Is Undecidable for fd's + ind's

As will be detailed in Chapter 10, fd's and ind's (and most other relational dependencies) can be represented as sentences in first-order logic. By Gödel's Completeness Theorem implication is recursively enumerable for first-order logic. It follows that unrestricted implication is r.e. for fd's and ind's considered together. On the other hand, finite implication for fd's and ind's taken together is co-r.e. This follows from the fact that there is an effective enumeration of all finite instances over a fixed schema; if $\Sigma \not\models_{\text{fin}} \sigma$, then a witness of this fact will eventually be found. When unrestricted and finite implication coincide, this pair of observations is sufficient to imply decidability of implication; this is not the case for fd's and ind's.

The Word Problem for (Finite) Monoids

The proof that (finite) implication for fd's and ind's taken together is undecidable uses a reduction from the word problem for monoids, which we discuss next.

A monoid is a set with an associative binary operation $\circ$ defined on it and an identity element ε. Let Γ be a finite alphabet and Γ^* the *free monoid* generated by Γ (i.e., the set of finite words with letters in Γ with the concatenation operation). Let $E = \{\alpha_i = \beta_i \mid i \in [1..n]\}$ be a finite set of equalities, and let e be an additional equality $\alpha = \beta$, where $\alpha_i, \beta_i, \alpha, \beta \in \Gamma^*$. Then E *(finitely) implies* e, denoted $E \models_{\text{unr}} e$ ($E \models_{\text{fin}} e$), if for each (finite) monoid M and homomorphism $h : \Gamma^* \to M$, if $h(\alpha_i) = h(\beta_i)$ for each $i \in [1..n]$, then $h(\alpha) = h(\beta)$. The *word problem* for (finite) monoids is to decide, given E and e, whether $E \models_{\text{unr}} e$ ($E \models_{\text{fin}} e$). Both the word problem for monoids and the word problem for finite monoids are undecidable.

Using this, we have the following:

THEOREM 9.2.4 Unrestricted and finite implication for fd's and ind's considered together are undecidable. In particular, let Σ range over sets of fd's and ind's. The following sets are not recursive:

(a) $\{(\Sigma, \sigma) \mid \sigma \text{ an ind and } \Sigma \models_{\text{unr}} \sigma\}$; $\{(\Sigma, \sigma) \mid \sigma \text{ an ind and } \Sigma \models_{\text{fin}} \sigma\}$;

(b) $\{(\Sigma, \sigma) \mid \sigma$ an fd and $\Sigma \models_{\text{unr}} \sigma\}$; and $\{(\Sigma, \sigma) \mid \sigma$ an fd and $\Sigma \models_{\text{fin}} \sigma\}$.

Crux We prove (a) using a reduction from the word problem for (finite) monoids to the (finite) implication problem for fd's and ind's. The proof of part (b) is similar and is left for Exercise 9.5. We first consider the unrestricted case.

Let Γ be a fixed alphabet. Let $E = \{\alpha_i = \beta_i \mid i \in [1, n]\}$ be a set of equalities over Γ^*, and let e be another equality $\alpha = \beta$. A *prefix* is defined to be any prefix of $\alpha_i, \beta_i, \alpha$, or β (including the empty string ε, and full words α_1, β_1, etc.). A single relation R is used, which has attributes

(i) A_γ, for each prefix γ;
(ii) A_x, A_y, A_{xy};
(iii) A_{ya}, for each $a \in \Gamma$; and
(iv) A_{xya}, for each $a \in \Gamma$;

where x and y are two fixed symbols.

To understand the correspondence between constrained relations and homomorphisms over monoids, suppose that there is a homomorphism h from Γ^* to some monoid M. Intuitively, a tuple of R will hold information about two elements $h(x), h(y)$ of M (in columns A_x, A_y, respectively) and their product $h(x) \circ h(y) = h(xy)$ (in column A_{xy}). For each a in Γ, tuples will also hold information about $h(ya)$ and $h(xya)$ in columns A_{ya}, A_{xya}. More precisely, the instance $I_{M,h}$ corresponding to the monoid M and the homomorphism $h : \Gamma^* \to M$ is defined by

$$I_{M,h} = \{t_{u,v} \mid u, v \in \Gamma^*\},$$

where for each $u, v \in \Gamma^*$, $t_{u,v}$ is the tuple such that

$$\begin{aligned} t_{u,v}(A_x) &= h(u), & t_{u,v}(A_\gamma) &= h(\gamma), \text{ for each prefix } \gamma, \\ t_{u,v}(A_y) &= h(v), & t_{u,v}(A_{ya}) &= h(va), \text{ for each } a \in \Gamma, \\ t_{u,v}(A_{xy}) &= h(uv), & t_{u,v}(A_{xya}) &= h(uva), \text{ for each } a \in \Gamma. \end{aligned}$$

Formally, to force the correspondence between the relations and homomorphisms over monoids, we use a set Σ of dependencies. In other words, we wish to find a set Σ of dependencies that characterizes precisely the instances over R that correspond to some homomorphism h from Γ^* to some monoid M. The key to the proof is that this can be done using just fd's and ind's. Strictly speaking, the dependencies of (8) in the following list are not ind's because an attribute is repeated in the left-hand side. As discussed in Exercise 9.4(e), the set of dependencies used here can be modified to a set of proper ind's that has the desired properties. In addition, we use fd's with an empty left-hand side, which are sometimes not considered as real fd's. The use of such dependencies is not crucial. A slightly more complicated proof can be found that uses only fd's with a nonempty left-hand side. The set Σ is defined as follows:

1. $\emptyset \to A_\gamma$ for each prefix γ;
2. $A_x A_y \to A_{xy}$;
3. $A_y \to A_{ya}$, for each $a \in \Gamma$;
4. $R[A_\varepsilon] \subseteq R[A_y]$;
5. $R[A_\gamma, A_{\gamma a}] \subseteq R[A_y, A_{ya}]$, for each $a \in \Gamma$ and prefix γ;
6. $R[A_{xy}, A_{xya}] \subseteq R[A_y, A_{ya}]$, for each $a \in \Gamma$;
7. $R[A_x, A_{ya}, A_{xya}] \subseteq R[A_x, A_y, A_{xy}]$, for each $a \in \Gamma$;
8. $R[A_y, A_\varepsilon, A_y] \subseteq R[A_x, A_y, A_{xy}]$; and
9. $R[A_{\alpha_i}] \subseteq R[A_{\beta_i}]$, for each $i \in [1, n]$.

The ind σ is $R[A_\alpha] \subseteq R[A_\beta]$.

Let I be an instance satisfying Σ. Observe that I has to satisfy a number of implied properties. In particular, one can verify that I also satisfies the following property:

$$R[A_{xya}] \subseteq R[A_{ya}] \subseteq R[A_y] = R[A_{xy}] \subseteq R[A_x]$$

and that $adom(I) \subseteq I[A_x]$.

We now show that $\Sigma \models_{\text{unr}} \sigma$ iff $E \models_{\text{unr}} e$. We first show that $E \not\models_{\text{unr}} e$ implies $\Sigma \not\models_{\text{unr}} \sigma$. Suppose that there is a monoid M and homomorphism $h : \Gamma^* \to M$ that satisfies the equations of E but violates the equation e. Consider $I_{M,h}$ defined earlier. It is straightforward to verify that $I \models \Sigma$ but $I \not\models \sigma$.

For the opposite direction, suppose now that $E \models_{\text{unr}} e$, and let I be a (possibly infinite) instance of R that satisfies Σ. To conclude the proof, it must be shown that $I[A_\alpha] \subseteq I[A_\beta]$. (Observe that these two relations both consist of a single tuple because of the fd's with an empty left-hand-side.)

We now define a function $h : \Gamma^* \to adom(I)$. We will prove that h is a homomorphism from Γ^* to a free monoid whose elements are $h(\Gamma^*)$ and that satisfies the equations of E (and hence, e). We will use the fact that the monoid satisfies e to derive that $I[A_\alpha] \subseteq I[A_\beta]$.

We now give an inductive definition of h and show that it has the property that $h(v) \in I[A_y]$ for each $v \in \Gamma^*$.

Basis: Set $h(\varepsilon)$ to be the element in $I[A_\varepsilon]$. Note that $h(\varepsilon)$ is also in $I[A_y]$ because $R[A_\varepsilon] \subseteq R[A_y] \in \Sigma$.

Inductive step: Given $h(v)$ and $a \in \Gamma$, let $t \in I$ be such that $t[A_y] = h(v)$. Define $h(va) = t(A_{ya})$. This is uniquely determined because $A_y \to A_{ya} \in \Sigma$. In addition, $h(va) \in I[A_y]$ because $R[A_x, A_{ya}, A_{xya}] \subseteq R[A_x, A_y, A_{xy}] \in \Sigma$.

We next show by induction on v that

$$(\dagger) \qquad \langle h(u), h(v), h(uv) \rangle \in I[A_x, A_y, A_{xy}] \text{ for each } u, v \in \Gamma^*.$$

For a fixed u, the basis (i.e., $v = \varepsilon$) is provided by the fact that $h(u) \in I[A_y]$ and the ind $R[A_y, A_\varepsilon, A_y] \subseteq R[A_x, A_y, A_{xy}] \in \Sigma$. For the inductive step, let $\langle h(u), h(v), h(uv) \rangle \in I[A_x, A_y, A_{xy}]$ and $a \in \Gamma$. Let $t \in I$ be such that $t[A_x, A_y, A_{xy}] = \langle h(u), h(v), h(uv) \rangle$.

Then by construction of h, $h(va) = t(A_{ya})$, and from the ind $R[A_{xy}, A_{xya}] \subseteq R[A_y, A_{ya}]$, we have $h(uva) = t(A_{xya})$. Finally, the ind $R[A_x, A_{ya}, A_{xya}] \subseteq R[A_x, A_y, A_{xy}]$ implies that $\langle h(u), h(va), h(uva)\rangle \in I[A_x, A_y, A_{xy}]$ as desired.

Define the binary operation $\circ$ on $h(\Gamma^*)$ as follows. For $a, b \in h(\Gamma^*)$, let

$$a \circ b = c \text{ if for some } t \in I,\, t[A_x, A_y, A_{xy}] = \langle a, b, c\rangle.$$

There is such a tuple by (†) and c is uniquely defined because $A_x, A_y \rightarrow A_{xy} \in \Sigma$. Furthermore, by (†), for each u, v, $h(u) \circ h(v) = h(uv)$. Thus for $h(u), h(v), h(w)$ in $h(\Gamma^*)$,

$$(h(u) \circ h(v)) \circ h(w) = h(uvw) = (h(u) \circ h(v)) \circ h(w),$$

and

$$h(u) \circ h(\varepsilon) = h(u)$$

so $(h(\Gamma^*), \circ)$ is a monoid. In addition, h is a homomorphism from the free monoid over Γ^* to the monoid $(h(\Gamma^*), \circ)$.

It is easy to see that $I[A_{\alpha_i}] = \{h(\alpha_i)\}$ and $I[A_{\beta_i}] = \{h(\beta_i)\}$ for $i \in [1, n]$. Let i be fixed. Because $R[A_{\alpha_i}] \subseteq R[A_{\beta_i}]$, $h(\alpha_i) = h(\beta_i)$. Because $E \models_{\text{unr}} e$, $h(\alpha) = h(\beta)$. Thus $I[A_\alpha] = \{h(\alpha)\} = \{h(\beta)\} = I[A_\beta]$. It follows that $I \models_{\text{unr}} R[A_\alpha] \subseteq R[A_\beta]$ as desired.

This completes the proof for the unrestricted case. For the finite case, note that everything has to be finite: The monoid is finite, I is finite, and the monoid $h[\Gamma^*]$ is finite. The rest of the argument is the same. ■

The issue of decidability of finite and unrestricted implication for classes of dependencies is revisited in Chapter 10.

9.3 Nonaxiomatizability of fd's + ind's

The inference rules given previously for fd's, mvd's and ind's can be viewed as "inference rule schemas," in the sense that each of them can be instantiated with specific attribute sets (sequences) to create infinitely many ground inference rules. In these cases the family of inference rule schemas is finite, and we informally refer to them as "finite axiomatizations."

Rather than formalizing the somewhat fuzzy notion of inference rule schema, we focus in this section on families $\mathcal{R}$ of ground inference rules. A (*ground*) *axiomatization* of a family $\mathcal{S}$ of dependencies is a set of ground inference rules that is sound and complete for (finite or unrestricted) implication for $\mathcal{S}$. Two properties of an axiomatization $\mathcal{R}$ will be considered, namely: (1) $\mathcal{R}$ is recursive, and (2) $\mathcal{R}$ is k-ary, in the sense (formally defined later in this section) that each rule in $\mathcal{R}$ has at most k dependencies in its condition.

Speaking intuitively, if $\mathcal{S}$ has a "finite axiomatization," that is, if there is a finite family $\mathcal{R}'$ of inference rule schemas that is sound and complete for $\mathcal{S}$, then $\mathcal{R}'$ specifies a ground axiomatization for $\mathcal{S}$ that is both recursive and k-ary for some k. Two results are demonstrated in this section: (1) There is no recursive axiomatization for finite implication of fd's and ind's, and (2) there is no k-ary axiomatization for finite implication of fd's and ind's. It is also known that there is no k-ary axiomatization for unrestricted implication of fd's and ind's. The intuitive conclusion is that the family of fd's and ind's does not have a "finite axiomatization" for finite implication or for unrestricted implication.

To establish the framework and some notation, we assume temporarily that we are dealing with a family $\mathcal{F}$ of database instances over a fixed database schema $\mathbf{R} = \{R_1, \ldots, R_n\}$. Typically, $\mathcal{F}$ will be the set of all finite instances over $\mathbf{R}$, or the set of all (finite or infinite) instances over $\mathbf{R}$. All the notions that are defined are *with respect to* $\mathcal{F}$. Let $\mathcal{S}$ be a family of dependencies over $\mathbf{R}$. (At present, $\mathcal{S}$ would be the set of fd's and ind's over $\mathbf{R}$.) Logical implication $\models$ among dependencies in $\mathcal{S}$ is defined with respect to $\mathcal{F}$ in the natural manner. In particular, $\models_{\text{unr}}$ and $\models_{\text{fin}}$ are obtained by letting $\mathcal{F}$ be the set of unrestricted or finite instances.

A (*ground*) *inference rule* over $\mathcal{S}$ is an expression of the form

$$\rho = \text{if } S \text{ then } s,$$

where $S \subseteq \mathcal{S}$ and $s \in \mathcal{S}$.

Let $\mathcal{R}$ be a set of rules over $\mathbf{R}$. Then $\mathcal{R}$ is *sound* if each rule in $\mathcal{R}$ is sound. Let $\Sigma \cup \{\sigma\} \subseteq \mathcal{S}$ be a set of dependencies over $\mathbf{R}$. A *proof* of σ from Σ using $\mathcal{R}$ is a finite sequence $\sigma_1, \ldots, \sigma_n = \sigma$ such that for each $i \in [1, n]$, either (1) $\sigma_i \in \Sigma$, or (2) for some rule 'if S then s' in $\mathcal{R}$, $\sigma_i = s$ and $S \subseteq \{\sigma_1, \ldots, \sigma_{i-1}\}$. We write $\Sigma \vdash_{\mathcal{R}} \sigma$ (or $\Sigma \vdash \sigma$ if $\mathcal{R}$ is understood) if there is a proof of σ from Σ using $\mathcal{R}$. Clearly, if each rule in $\mathcal{R}$ is sound, then $\Sigma \vdash \sigma$ implies $\Sigma \models \sigma$. The set $\mathcal{R}$ is *complete* if for each pair (Σ, σ), $\Sigma \models \sigma$ implies $\Sigma \vdash_{\mathcal{R}} \sigma$. A (*sound and complete*) *axiomatization* for logical implication is a set $\mathcal{R}$ of rules that is sound and complete.

The aforementioned notions are now generalized to permit all schemas $\mathbf{R}$. In particular, we consider a set $\mathcal{R}$ of rules that is a union $\cup\{\mathcal{R}_{\mathbf{R}} \mid \mathbf{R} \text{ is a schema}\}$. The notions of sound, proof, etc.can be generalized in the natural fashion.

Note that with the preceding definition, every set $\mathcal{S}$ of dependencies has a sound and complete axiomatization. This is provided by the set $\mathcal{R}$ of all rules of the form

$$\text{if } S \text{ then } s,$$

where $S \models s$. Clearly, such trivial axiomatizations hold no interest. In particular, they are not necessarily effective (i.e., one may not be able to tell if a rule is in $\mathcal{R}$, so one may not be able to construct proofs that can be checked). It is thus natural to restrict $\mathcal{R}$ to be recursive.

We now present the first result of this section, which will imply that there is no recursive axiomatization for finite implication of fd's and ind's. In this result we assume that the dependencies in $\mathcal{S}$ are sentences in first-order logic.

Proposition 9.3.1 Let $\mathcal{S}$ be a class of dependencies. If $\mathcal{S}$ has a recursive axiomatization for finite implications, then finite implication is decidable for $\mathcal{S}$.

Crux Suppose that $\mathcal{S}$ has a recursive axiomatization. Consider the set

$$Implic = \{(S, s) \mid S \subseteq \mathcal{S}, s \in \mathcal{S}, \text{ and } S \models_{\text{fin}} s\}.$$

First note that the set *Implic* is r.e.; indeed, let $\mathcal{R}$ be a recursive axiomatization for $\mathcal{S}$. One can effectively enumerate all proofs of implication that use rules in $\mathcal{R}$. This allows one to enumerate *Implic* effectively. Thus *Implic* is r.e. We argue next that *Implic* is also co-r.e. To conclude that a pair (S, s) is not in *Implic*, it is sufficient to exhibit a finite instance satisfying S and violating s. To enumerate all pairs (S, s) not in *Implic*, one proceeds as follows. The set of *all* pairs (S, s) is clearly r.e., as is the set of all instances over a fixed schema. Repeat for all positive integers n the following. Enumerate the first n pairs (S, s) and the first n instances. For each (S, s) among the n, check whether one of the n instances is a counterexample to the implication $S \models s$, in which case output (S, s). Clearly, this procedure enumerates the complement of *Implic*, so *Implic* is co-r.e. Because it is both r.e. and co-r.e., *Implic* is recursive, so there is an algorithm testing whether (S, s) is in *Implic*. ■

It follows that there is no recursive axiomatization for finite implication of fd's and ind's. [To see this, note that by Theorem 9.2.4, logical implication for fd's and ind's is undecidable. By Proposition 9.3.1, it follows that there can be no finite axiomatization for fd's and ind's.] Because implication for jd's is decidable (Theorem 8.4.12), but there is no axiomatization for them (Theorem 8.3.4), the converse of the preceding proposition does not hold.

Speaking intuitively, the preceding development implies that there is no finite set of inference rule schemas that is sound and complete for finite implication of fd's and ind's. However, the proof is rather indirect. Furthermore, the approach cannot be used in connection with unrestricted implication, nor with classes of dependencies for which finite implication is decidable (see Exercise 9.9). The notion of k-ary axiomatization developed now shall overcome these objections.

A rule 'if S then s' is *k-ary* for some $k \geq 0$ if $|S| = k$. An axiomatization $\mathcal{R}$ is *k-ary* if each rule in $\mathcal{R}$ is l-ary for some $l \leq k$. For example, the instantiations of rules FD1 and IND1 are 0-ary, those of rules FD2 and IND2 are 1-ary, and those of FD3 and IND3 are 2-ary. Theorem 9.3.3 below shows that there is no k-ary axiomatization for finite implication of fd's and ind's.

We now turn to an analog in terms of logical implication of k-ary axiomatizability. Again let $\mathcal{S}$ be a set of dependencies over $\mathbf{R}$, and let $\mathcal{F}$ be a family of instances over $\mathbf{R}$. Let $k \geq 0$. A set $\Gamma \subseteq \mathcal{S}$ is:

closed under implication with respect to $\mathcal{S}$ if $\sigma \in \Gamma$ whenever

$$\text{(a) } \sigma \in \mathcal{S} \text{ and (b) } \Gamma \models \sigma$$

closed under k-ary implication with respect to $\mathcal{S}$ if $\sigma \in \Gamma$ whenever

$$\text{(a) } \sigma \in \mathcal{S}, \text{ and for some } \Sigma \subseteq \Gamma, \text{ (b}_1\text{) } \Sigma \models \sigma \text{ and (b}_2\text{) } |\Sigma| \leq k.$$

Clearly, if Γ is closed under implication, then it is closed under k-ary implication for each

$k \geq 0$, and if Γ is closed under k-ary implication, then it is closed under k'-ary implication for each $k' \leq k$.

Proposition 9.3.2 Let $\mathbf{R}$ be a database schema, $\mathcal{S}$ a set of dependencies over $\mathbf{R}$, and $k \geq 0$. Then there is a k-ary axiomatization for $\mathcal{S}$ iff whenever $\Gamma \subseteq \mathcal{S}$ is closed under k-ary implication, then Γ is closed under implication.

Proof Suppose that there is a k-ary axiomatization for $\mathcal{S}$, and let $\Gamma \subseteq \mathcal{S}$ be closed under k-ary implication. Suppose further that $\Gamma \models \sigma$ for some $\sigma \in \mathcal{S}$. Let $\sigma_1, \ldots, \sigma_n$ be a proof of σ from Γ using $\mathcal{R}$. Using the fact that $\mathcal{R}$ is k-ary and that Γ is closed under k-ary implication, a straightforward induction shows that $\sigma_i \in \Gamma$ for $i \in [1, n]$.

Suppose now that for each $\Gamma \subseteq \mathcal{S}$, if Γ is closed under k-ary implication, then Γ is closed under implication. Set

$$\mathcal{R} = \{\text{`if } S \text{ then } s\text{'} \mid S \subseteq \mathcal{S}, s \in \mathcal{S}, |S| \leq k, \text{ and } S \models s\}.$$

To see that $\mathcal{R}$ is complete, suppose that $\Gamma \models \sigma$. Consider the set $\Gamma^* = \{\gamma \mid \Gamma \vdash_{\mathcal{R}} \gamma\}$. From the construction of $\mathcal{R}$, Γ^* is closed under k-ary implication. By assumption it is closed under implication, and so $\Gamma \vdash_{\mathcal{R}} \sigma$ as desired. ■

In the following, we consider finite implication, so $\mathcal{F}$ is the set of finite instances.

Theorem 9.3.3 For no k does there exist a k-ary sound and complete axiomatization for finite implication of fd's and ind's taken together. More specifically, for each k there is a schema $\mathbf{R}$ for which there is no k-ary sound and complete axiomatization for finite implication of fd's and ind's over $\mathbf{R}$.

Proof Let $k \geq 0$ be fixed. Let $\mathbf{R} = \{R_0, \ldots, R_k\}$ be a database schema where $sort(R_i) = \{A, B\}$ for each $i \in [0, k]$. In the remainder of this proof, addition is always done modulo $k + 1$. The dependencies $\Sigma = \Sigma_a \cup \Sigma_b$ and σ are defined by

(a) $\Sigma_a = \{R_i : A \to B \mid i \in [0, k]\}$;
(b) $\Sigma_b = \{R_i[A] \subseteq R_{i+1}[B] \mid i \in [0, k]\}$; and
(c) $\sigma = R_0[B] \subseteq R_k[A]$.

Let Γ be the union of Σ with all fd's and ind's that are tautologies (i.e., that are satisfied by all finite instances over $\mathbf{R}$).

In the remainder of the proof, it is shown that (1) Γ is not closed under finite implication, but (2) Γ is closed under k-ary finite implication. Proposition 9.3.2 will then imply that the family of fd's and ind's has no k-ary sound and complete axiomatization for $\mathbf{R}$.

First observe that Γ does not contain σ, so to show that Γ is not closed under finite implication, it suffices to demonstrate that $\Sigma \models_{\text{fin}} \sigma$. Let $\mathbf{I}$ be a finite instance of $\mathbf{R}$ that satisfies Σ. By the ind's of Σ, $|\mathbf{I}(R_i)[A]| \leq |\mathbf{I}(R_{i+1})[B]|$ for each $i \in [0, k]$, and by the fd's of Σ, $|\mathbf{I}(R_i)[B]| \leq |\mathbf{I}(R_i)[A]|$ for each $i \in [0, k]$. From this we obtain

$$\begin{aligned}|\mathbf{I}(R_0)[A]| &\le |\mathbf{I}(R_1)[B]| \le |\mathbf{I}(R_1)[A]| \\ &\le \qquad \ldots \\ &\le |\mathbf{I}(R_k)[B]| \le |\mathbf{I}(R_k)[A]| \le |\mathbf{I}(R_0)[B]| \le |\mathbf{I}(R_0)[A]|.\end{aligned}$$

In particular, $|\mathbf{I}(R_k)[A]| = |\mathbf{I}(R_0)[B]|$. Since $\mathbf{I}$ is finite and we have $\mathbf{I}(R_k)[A] \subseteq \mathbf{I}(R_0)[B]$ and $|\mathbf{I}(R_k)[A]| = |\mathbf{I}(R_0)[B]|$, it follows that $\mathbf{I}(R_0)[B] \subseteq \mathbf{I}(R_k)[A]$ as desired.

We now show that Γ is closed under k-ary finite implication. Suppose that $\Delta \subseteq \Gamma$ has no more than k elements ($|\Delta| \le k$). It must be shown that if γ is an fd or ind and $\Delta \models_{\text{fin}} \gamma$, then $\gamma \in \Gamma$. Because Σ contains $k+1$ ind's, any subset Δ of Γ that has no more than k members must omit some ind δ of Σ. We shall exhibit an instance $\mathbf{I}$ such that $\mathbf{I} \models \gamma$ iff $\gamma \in \Gamma - \{\delta\}$. (Thus $\mathbf{I}$ will be an Armstrong instance for $\Gamma - \{\delta\}$.) It will then follow that $\Gamma - \{\delta\}$ is closed under finite implication. Because $\Delta \subseteq \Gamma - \{\delta\}$, this will imply that for each fd or ind γ, if $\Delta \models_{\text{fin}} \gamma$, then $\Gamma - \{\delta\} \models_{\text{fin}} \gamma$, so $\gamma \in \Gamma$.

Because Σ is symmetric with regard to ind's, we can assume without loss of generality that δ is the ind $R_k[A] \subseteq R_0[B]$. Assuming that $\mathbf{N} \times \mathbf{N}$ is contained in the underlying domain, define $\mathbf{I}$ so that

$$\mathbf{I}(R_0) = \{\langle(0,0),(0,k+1)\rangle,\ \langle(1,0),(1,k+1)\rangle,\ \langle(2,0),(1,k+1)\rangle\}$$

and for each $i \in [1,k]$,

$$\begin{aligned}\mathbf{I}(R_i) = \{&\langle(0,i),(0,i-1)\rangle,\ \langle(1,i),(1,i-1)\rangle,\ldots,\\ &\langle(2i+1,i),(2i+1,i-1)\rangle,\ \langle(2i+2,i),(2i+1,i-1)\rangle\}.\end{aligned}$$

Figure 9.2 shows $\mathbf{I}$ for the case $k = 3$.

We now show for each fd and ind γ over $\mathbf{R}$ that $\mathbf{I} \models \gamma$ iff $\gamma \in \Gamma - \delta$. Three cases arise:

1. γ is a tautology. Then this clearly holds.
2. γ is an fd that is not a tautology. Then γ is equivalent to one of the following for some $i \in [0,k]$:

$$\begin{array}{ll} R_i : A \to B, & R_i : B \to A, \\ R_i : \emptyset \to A, & R_i : \emptyset \to B, \\ \text{or} & R_i : \emptyset \to AB. \end{array}$$

   If γ is $R_i : A \to B$, then $\gamma \in \Gamma$ and clearly $\mathbf{I} \models \gamma$. In the other cases, $\gamma \notin \Gamma$ and $\mathbf{I} \not\models \gamma$.
3. γ is an ind that is not a tautology. Considering now which ind's $\mathbf{I}$ satisfies, note that the only pairs of nondisjoint columns of relations in $\mathbf{I}$ are

$$\begin{array}{l} \mathbf{I}(R_0)[A], \mathbf{I}(R_1)[B]; \\ \mathbf{I}(R_1)[A], \mathbf{I}(R_2)[B]; \quad \ldots; \\ \mathbf{I}(R_{k-1})[A], \mathbf{I}(R_k)[B]. \end{array}$$

   Furthermore, $\mathbf{I} \not\models R_{i+1}[B] \subseteq R_i[A]$ for each $i \in [0,k]$; and $\mathbf{I} \models R_i[A] \subseteq R_{i+1}[B]$.

This implies that $\mathbf{I} \models \gamma$ iff $\gamma \in \Gamma - \{\delta\}$, as desired. ■

$\mathbf{I}(R_0)$	A	B
	(0,0)	(0,4)
	(1,0)	(1,4)
	(2,0)	(1,4)

$\mathbf{I}(R_1)$	A	B
	(0,1)	(0,0)
	(1,1)	(1,0)
	(2,1)	(2,0)
	(3,1)	(3,0)
	(4,1)	(3,0)

$\mathbf{I}(R_2)$	A	B
	(0,2)	(0,1)
	(1,2)	(1,1)
	(2,2)	(2,1)
	(3,2)	(3,1)
	(4,2)	(4,1)
	(5,2)	(5,1)
	(6,2)	(5,1)

$\mathbf{I}(R_3)$	A	B
	(0,3)	(0,2)
	(1,3)	(1,2)
	(2,3)	(2,2)
	(3,3)	(3,2)
	(4,3)	(4,2)
	(5,3)	(5,2)
	(6,3)	(6,2)
	(7,3)	(7,2)
	(8,3)	(7,2)

Figure 9.2: An Armstrong relation for $\Gamma - \delta$

In the proof of the preceding theorem all relations used are binary, and all fd's and ind's are *unary*, in the sense that at most one attribute appears on either side of each dependency. In proofs that there is no k-ary axiomatization for unrestricted implication of fd's and ind's, some of the ind's used involve at least two attributes on each side. This cannot be improved to unary ind's, because there is a 2-ary sound and complete axiomatization for unrestricted implication of unary ind's and arbitrary fd's (see Exercise 9.18).

9.4 Restricted Kinds of Inclusion Dependency

This section explores two restrictions on ind's for which several positive results have been obtained. The first one focuses on sets of ind's that are acyclic in a natural sense, and the second restricts the ind's to having only one attribute on either side. The restricted dependencies are important because they are sufficient to model many natural relationships, such as those captured by semantic models (see Chapter 11). These include subtype relationships of the kind "every student is also a person."

This section also presents a generalization of the chase that incorporates ind's. Because ind's are embedded, chasing in this context may lead to infinite chasing sequences. In the context of acyclic sets of ind's, however, the chasing sequences are guaranteed to terminate. The study of infinite chasing sequences will be taken up in earnest in Chapter 10.

Ind's and the Chase

Because ind's may involve more than one relation, the formal notation of the chase must be extended. Suppose now that **R** is a database schema, and let $q = (\mathbf{T}, t)$ be a tableau query over **R**. The fd and jd rules are generalized to this context in the natural fashion.

We first present an example and then describe the rule that is used for ind's.

Example 9.4.1 Consider the database schemas consisting of two relation schemas P, Q with $sort(P) = ABC$, $sort(Q) = DEF$, the dependencies

$$Q[DE] \subseteq P[AB] \qquad \text{and} \qquad P : A \to B,$$

and the tableau **T** shown in Fig. 9.3. Consider $\mathbf{T}_1$ and $\mathbf{T}_2$ in the same figure. The tableau $\mathbf{T}_1$ is obtained by applying to **T** the ind rule given after this example. The intuition is that the tuples $\langle x, y_i \rangle$ should also be in the P-relation because of the ind. Then $\mathbf{T}_2$ is obtained by applying the fd rule. Tableau minimization can be applied to obtain $\mathbf{T}_3$.

The following rule is used for ind's.

ind rule: Let $\sigma = R[X] \subseteq S[Y]$ be an ind, let $u \in \mathbf{T}(R)$, and suppose that there is no free tuple $v \in \mathbf{T}(S)$ such that $v[Y] = u[X]$. In this case, we say that σ is *applicable* to $R(u)$. Let w be a free tuple over S such that $w[Y] = u[X]$ and w has distinct new variables in all coordinates of $sort(S) - Y$ that are greater than all variables occurring in q. Then "the" result of applying σ to $R(u)$ is $(\mathbf{T}', t)$, where

- $\mathbf{T}'(P) = \mathbf{T}(P)$ for each relation name $P \in \mathbf{R} - \{S\}$, and
- $\mathbf{T}'(S) = \mathbf{T}(S) \cup \{w\}$.

For a tableau query q and a set Σ of ind's, it is possible that two terminal chasing sequences end with nonisomorphic tableau queries, that there are no finite terminal chasing sequences, or that there are both finite terminal chasing sequences and infinite chasing sequences (see Exercise 9.12). General approaches to resolving this problem will be considered in Chapter 10. In the present discussion, we focus on acyclic sets of ind's, for which the chase always terminates after a finite number of steps.

Acyclic Inclusion Dependencies

Definition 9.4.2 A family Σ of ind's over **R** is *acyclic* if there is no sequence $R_i[X_i] \subseteq S_i[Y_i]$ ($i \in [1, n]$) of ind's in Σ where for $i \in [1, n]$, $R_{i+1} = S_i$ for $i \in [1, n-1]$, and $R_1 = S_n$. A family Σ of dependencies has *acyclic* ind's if the set of ind's in Σ is acyclic.

The following is easily verified (see Exercise 9.14):

Proposition 9.4.3 Let q be a tableau query and Σ a set of fd's, jd's, and acyclic ind's over **R**. Then each chasing sequence of q by Σ terminates after an exponentially bounded number of steps.

$\mathbf{T}(P)$	A	B	C

$\mathbf{T}(Q)$	D	E	F
	x	y_1	z
	x	y_2	x
t	y_1	x	

$\mathbf{T}_1(P)$	A	B	C
	x	y_1	w_1
	x	y_2	w_2

$\mathbf{T}_1(Q)$	D	E	F
	x	y_1	z
	x	y_2	x
t	y_1	x	

$\mathbf{T}_2(P)$	A	B	C
	x	y_1	w_1
	x	y_1	w_2

$\mathbf{T}_2(Q)$	D	E	F
	x	y_1	z
	x	y_1	x
t	y_1	x	

$\mathbf{T}_3(P)$	A	B	C
	x	y_1	w_1

$\mathbf{T}_3(Q)$	D	E	F
	x	y_1	x
t	y_1	x	

Figure 9.3: Chasing with ind's

For each tableau query q and set Σ of fd's, jd's, and acyclic ind's, let $chase(q, \Sigma)$ denote the result of some arbitrary chasing sequence of q by Σ. (One can easily come up with some syntactic strategy for arbitrarily choosing this sequence.)

Using an analog to Lemma 8.4.3, one obtains the following result on tableau query containment (an analog to Theorem 8.4.8).

Theorem 9.4.4 Let q, q' be tableau queries and Σ a set of fd's, jd's, and acyclic ind's over $\mathbf{R}$. Then $q \subseteq_\Sigma q'$ iff $chase(q, \Sigma) \subseteq chase(q', \Sigma)$.

Next we consider the application of the chase to implication of dependencies. For database schema $\mathbf{R}$ and ind $\sigma = R[X] \subseteq S[Y]$ over $\mathbf{R}$, the *tableau query* of σ is $q_\sigma = (\{R(u_\sigma)\}, \langle u_\sigma \rangle)$, where u_σ is a free tuple all of whose entries are distinct. For example, given $R[ABCD]$, $S[EFG]$, and $\sigma = R[BC] \subseteq S[GE]$, $q_\sigma = (\{R(x_1, x_2, x_3, x_4)\}, \langle x_1, x_2,$

$x_3, x_4\rangle$). In analogy with Theorem 8.4.12, we have the following for fd's, jd's, and acyclic ind's.

Theorem 9.4.5 Let Σ be a set of fd's, jd's, and acyclic ind's over database schema **R** and let **T** be the tableau in $chase(q_\sigma, \Sigma)$. Then $\Sigma \models_{\text{unr}} \sigma$ iff

(a) For fd or jd σ over **R**, **T** satisfies the conditions of Theorem 8.4.12.
(b) For ind $\sigma = R[X] \in S[Y]$, $u_\sigma[X] \in \mathbf{T}(S)[Y]$.

This yields the following:

Corollary 9.4.6 Finite and unrestricted implication for sets of fd's, jd's, and acyclic ind's coincide and are decidable in exponential time.

An improvement of the complexity here seems unlikely, because implication of an ind by an acyclic set of ind's is NP-complete (see Exercise 9.14).

Unary Inclusion Dependencies

A *unary inclusion dependency* (uind) is an ind in which exactly one attribute appears on each side. The uind's arise frequently in relation schemas in which certain columns range over values that correspond to entity types (e.g., if *SS#* is a key for the *Person* relation and is also used to identify people in the *Employee* relation).

As with arbitrary ind's, unrestricted and finite implication do not coincide for fd's and uind's (proof of Theorem 9.2.2). However, both forms of implication are decidable in polynomial time. In this section, the focus is on finite implication. We present a sound and complete axiomatization for finite implication of fd's and uind's (but in agreement with Theorem 9.3.3, it is not k-ary for any k).

For uind's considered in isolation, the inference rules for ind's are specialized to yield the following two rules, which are sound and complete for (unrestricted and finite) implication. Here A, B, and C range over attributes and R, S, and T over relation names:

UIND1: (reflexivity) $R[A] \subseteq R[A]$.

UIND2: (transitivity) If $R[A] \subseteq S[B]$ and $S[B] \subseteq T[C]$, then $R[A] \subseteq T[C]$.

To capture the interaction of fd's and uind's in the finite case, the following family of rules is used:

C*:* (cycle rules) For each positive integer n,

$$\text{if} \begin{cases} R_1 : A_1 \to B_1, \\ R_2[A_2] \subseteq R_1[B_1], \\ \dots, \\ R_n : A_n \to B_n, \text{ and} \\ R_1[A_1] \subseteq R_n[B_n] \end{cases} \qquad \text{then} \begin{cases} R_1 : B_1 \to A_1, \\ R_1[B_1] \subseteq R_2[A_2], \\ \dots, \\ R_n : B_n \to A_n, \text{ and} \\ R_n[B_n] \subseteq R_1[A_1]. \end{cases}$$

The soundness of this family of rules follows from a straightforward cardinality argument. More generally, we have the following (see Exercise 9.16):

Theorem 9.4.7 The set {FD1, FD2, FD3, UIND1, UIND2} along with the cycle rules (C) is sound and complete for finite implication of fd's and uind's. Furthermore, finite implication is decidable in polynomial time.

Bibliographic Notes

Inclusion dependency is based on the notion of referential integrity, which was known to the broader database community during the 1970s (see, e.g., [Dat81]). A seminal paper on the theory of ind's is [CFP84], in which inference rules for ind's are presented and the nonaxiomatizability of both finite and unrestricted implication for fd's and ind's is demonstrated. A non-k-ary sound and complete set of inference rules for finite implication of fd's and ind's is presented in [Mit83b]. Another seminal paper is [JK84b], which also observed the distinction between finite and unrestricted implication for fd's and ind's, generalized the chase to incorporate fd's and ind's, and used this to characterize containment between conjunctive queries. Related work is reported in [LMG83].

Undecidability of (finite) implication for fd's and ind's taken together was shown independently by [CV85] and [Mit83a]. The proof of Theorem 9.2.4 is taken from [CV85]. (The undecidability of the word problem for monoids is from [Pos47], and of the word problem for finite monoids is from [Gur66].)

Acyclic ind's were introduced in [Sci86]. Complexity results for acyclic ind's include that implication for acyclic ind's alone is NP-complete [CK86], and implication for fd's and acyclic ind's has an exponential lower bound [CK85].

Given the PSPACE complexity of implication for ind's and the negative results in connection with fd's, unary ind's emerged as a more tractable form of inclusion dependency. The decision problems for finite and unrestricted implication for uind's and fd's taken together, although not coextensive, both lie in polynomial time [CKV90]. This extensive paper also develops axiomatizations of both finite and unrestricted logical implication for unary ind's and fd's considered together, and develops results for uind's with some of the more general dependencies studied in Chapter 10.

Typed ind's are studied in [CK86]. In addition to using traditional techniques from dependency theory, such as chasing, this work develops tools for analyzing ind's using equational theories.

Ind's in connection with other dependencies are also studied in [CV83].

Exercises

Exercise 9.1 Complete the proof of Proposition 9.1.5.

Exercise 9.2 Complete the proof of Theorem 9.1.7.

Exercise 9.3 [CFP84] (In this exercise, by a slight abuse of notation, we allow fd's with sequences rather than sets of attributes.) Demonstrate the following:

(a) If $|\vec{A}| = |\vec{B}|$, then $\{R[\vec{A}\vec{C}] \subseteq S[\vec{B}\vec{D}], S : \vec{B} \to \vec{D}\} \models_{\text{unr}} R : \vec{A} \to \vec{C}$.

(b) If $|\vec{A}| = |\vec{B}|$, then $\{R[\vec{A}\vec{C}] \subseteq S[\vec{B}\vec{D}], R[\vec{A}\vec{E}] \subseteq S[\vec{B}\vec{F}], S : \vec{B} \rightarrow \vec{D}\} \models_{\text{unr}} R[\vec{A}\vec{C}\vec{E}] \subseteq S[\vec{B}\vec{D}\vec{F}]$.

(c) Suppose that $|\vec{A}| = |\vec{B}|$; $\Sigma = \{R[\vec{A}\vec{C}] \subseteq S[\vec{B}\vec{D}], R[\vec{A}\vec{E}] \subseteq S[\vec{B}\vec{D}], S : \vec{B} \rightarrow \vec{D}\}$; and $\mathbf{I} \models \Sigma$. Then $u[\vec{C}] = u[\vec{E}]$ for each $u \in \mathbf{I}(R)$.

Exercise 9.4 As defined in the text, we require in ind $R[A_1, \ldots, A_m] \subseteq S[B_1, \ldots, B_m]$ that the A_i's and B_i's are distinct. A *repeats-permitted inclusion dependency* (rind) is defined as was inclusion dependency, except that repeats are permitted in the attribute sequences on both the left- and right-hand sides.

(a) Show that if Σ is a set of ind's, σ a rind, and $\Sigma \models_{\text{unr}} \sigma$, then σ is equivalent to an ind.

(b) Exhibit a set Σ of ind's and fd's such that $\Sigma \models_{\text{unr}} R[AB] \subseteq S[CC]$. Do the same for $R[AA] \subseteq R[BC]$.

♠ (c) [Mit83a] Consider the rules

IND4: If $R[A_1A_2] \subseteq S[BB]$ and $R[\vec{C}] \subseteq T[\vec{D}]$, then $R[\vec{C}'] \subseteq T[\vec{D}]$, where $\vec{C}'$ is obtained from $\vec{C}$ by replacing one or more occurrences of A_2 by A_1.

IND5: If $R[A_1A_2] \subseteq S[BB]$ and $T[\vec{C}] \subseteq R[\vec{D}]$, then $T[\vec{C}] \subseteq R[\vec{D}']$, where $\vec{D}'$ is obtained from $\vec{D}$ by replacing one or more occurrences of A_2 by A_1.

Prove that the inference rules {IND1, IND2, IND3, IND4, IND5} are sound and complete for finite implication of sets of rind's.

(d) Prove that unrestricted and finite implication coincide for rind's.

(e) A *left-repeats-permitted inclusion dependency* (l-rind) is a rind for which there are no repeats on the right-hand side. Given a set $\Sigma \cup \{\sigma\}$ of l-rind's over $\mathbf{R}$, describe how to construct a schema $\mathbf{R}'$ and ind's $\Sigma' \cup \{\sigma'\}$ over $\mathbf{R}'$ such that $\Sigma \models \sigma$ iff $\Sigma' \models \sigma'$ and $\Sigma \models_{\text{fin}} \sigma$ iff $\Sigma' \models_{\text{fin}} \sigma'$.

(f) Do the same as in part (e), except for arbitrary rind's.

Exercise 9.5 [CV85] Prove part (b) of Theorem 9.2.4. *Hint:* In the proof of part (a), extend the schema of R to include new attributes $A_{\alpha'}$, $A_{\beta'}$, and $A_{y'}$; add dependencies $A_y \rightarrow A_{y'}$, $R[A_\alpha, A_{\alpha'}] \subseteq R[A_y, A_{y'}]$, $R[A_\beta, A_{\beta'}] \subseteq R[A_y, A_{y'}]$; and use $A_{\alpha'} \rightarrow A_{\beta'}$ as σ.

Exercise 9.6

(a) Develop an alternative proof of Theorem 9.3.3 in which δ is an fd rather than an ind.

(b) In the proof of Theorem 9.3.3 for finite implication, the dependency σ used is an ind. Using the same set Σ, find an fd that can be used in place of σ in the proof.

Exercise 9.7 Prove that there is no k for which there is a k-ary sound and complete axiomatization for finite implication of fd's, jd's, and ind's.

★ **Exercise 9.8** [SW82] Prove that there is no k-ary sound and complete set of inference rules for finite implication of emvd's.

Exercise 9.9 Recall the notion of sort-set dependency (ssd) from Exercise 8.32.

(a) Prove that finite and unrestricted implication coincide for fd's and ssd's considered together. Conclude that implication for fd's and ssd's is decidable.

★ (b) [GH86] Prove that there is no k-ary sound and complete set of inference rules for finite implication of fd's (key dependencies) and ssd's taken together.

Exercise 9.10

(a) [CFP84] A set of ind's is *bounded* by k if each ind in the set has at most k attributes on the left-hand side and on the right-hand side. Show that logical implication for bounded sets of ind's is decidable in polynomial time.

(b) [CV83] An ind is *typed* if it has the form $R[\vec{A}\,] \subseteq S[\vec{A}\,]$. Exhibit a polynomial time algorithm for deciding logical implication between typed ind's.

Exercise 9.11 Suppose that some attribute domains may be finite.

(a) Show that {IND1, IND2, IND3} remains sound in the framework.

(b) Show that if one-element domains are permitted, then {IND1, IND2, IND3} is not complete.

(c) Show for each $n > 0$ that if all domains are required to have at least n elements, then {IND1, IND2, IND3} is not complete.

Exercise 9.12 Suppose that no restrictions are put on the order of application of ind rules in chasing sequences.

(a) Exhibit a tableau query q and a set Σ of ind's and two terminal chasing sequences of q by Σ that end with nonisomorphic tableau queries.

(b) Exhibit a tableau query q and a set Σ of ind's, a terminal chasing sequence of q by Σ, and an infinite chasing sequence of q by Σ.

(c) Exhibit a tableau query q and a set Σ of ind's such that q has no finite terminal chasing sequence by Σ.

♠ **Exercise 9.13** [JK84b] Recall that for tableau queries q and q' and a set Σ of fd's and jd's over R, $q \subseteq_\Sigma q'$ if for each instance I that satisfies Σ, $q(I) \subseteq q'(I)$. In the context of ind's, this containment relationship may depend on whether infinite instances are permitted or not. For tableau queries q, q' and a set Σ of dependencies over $\mathbf{R}$, we write $q \subseteq_{\Sigma,\text{fin}} q'$ ($q \subseteq_{\Sigma,\text{unr}} q'$) if $q(\mathbf{I}) \subseteq q'(\mathbf{I})$ for each finite (unrestricted) instance $\mathbf{I}$ that satisfies Σ.

(a) Show that if Σ is a set of fd's and jd's, then $\subseteq_{\Sigma,\text{fin}}$ and $\subseteq_{\Sigma,\text{unr}}$ coincide.

(b) Exhibit a set Σ of fd's and ind's and tableau queries q, q' such that $q \subseteq_{\Sigma,\text{fin}} q'$ but $q \not\subseteq_{\Sigma,\text{unr}} q'$.

Exercise 9.14

(a) Prove Proposition 9.4.3.

(b) Prove Theorem 9.4.4.

(c) Let q be a tableau query and Σ a set of fd's, jd's, and ind's over $\mathbf{R}$, where the set of ind's in Σ is acyclic; and suppose that q', q'' are the final tableaux of two terminal chasing sequences of q by Σ (where the order of rule application is not restricted). Prove that $q \equiv q'$.

(d) Prove Theorem 9.4.5.

(e) Prove Corollary 9.4.6.

Exercise 9.15

(a) Exhibit an acyclic set Σ of ind's and a tableau query q such that $chase(q, \Sigma)$ is exponential in the size of Σ and q.

(b) [CK86] Prove that implication of an ind by an acyclic set of ind's is NP-complete. *Hint:* Use a reduction from the problem of Permutation Generation [GJ79].

(c) [CK86] Recall from Exercise 9.10(b) that an ind is *typed* if it has the form $R[\vec{A}] \subseteq S[\vec{A}]$. Prove that implication of an ind by a set of fd's and an acyclic set of typed ind's is NP-hard. *Hint:* Use a reduction from 3-SAT.

♠ **Exercise 9.16** [CKV90] In this exercise you will prove Theorem 9.4.7. The exercise begins by focusing on the unirelational case; for notational convenience we omit the relation name from uind's in this context.

Given a set Σ of fd's and uind's over R, define $G(\Sigma)$ to be a multigraph with node set R and two colors of edges: a red edge from A to B if $A \to B \in \Sigma$, and a black edge from A to B is $B \subseteq A \in \Sigma$. If A and B have red (black) edges in both directions, replace them with an undirected red (black) edge.

(a) Suppose that Σ is closed under the inference rules. Prove that $G(\Sigma)$ has the following properties:
   1. Nodes have red (black) self-loops, and the red (black) subgraph of $G(\Sigma)$ is transitively closed.
   2. The subgraphs induced by the strongly connected components of $G(\Sigma)$ contain only undirected edges.
   3. In each strongly connected component, the red (black) subset of edges forms a collection of node disjoint cliques (the red and black partitions of nodes could be different).
   4. If $A_1 \ldots A_m \to B$ is an fd in Σ and $A_1, \ldots, A_m$ have common ancestor A in the red subgraph of $G(\Sigma)$, then $G(\Sigma)$ contains a red edge from A to B.

(b) Given a set Σ of fd's and uind's closed under the inference rules, use $G(\Sigma)$ to build counterexample instances that demonstrate that $\Sigma \not\vdash \sigma$ implies $\Sigma \not\models_{\text{fin}} \sigma$ for fd or uind σ.

(c) Use the rules to develop a polynomial time algorithm for inferring finite implication for a set of fd's and uind's.

(d) Generalize the preceding development to arbitrary database schemas.

Exercise 9.17

(a) Let $k > 1$ be an integer. Prove that there is a database schema $\mathbf{R}$ with at least one unary relation $R \in \mathbf{R}$, and a set Σ of fd's and ind's such that
   (i) for each $\mathbf{I} \models \Sigma$, $|\mathbf{I}(R)| = 0$ or $|\mathbf{I}(R)| = 1$ or $|\mathbf{I}(R)| \geq k$.
   (ii) for each $l \geq k$ there is an instance $\mathbf{I}_l \models \Sigma$ with $|\mathbf{I}(R)| = l$.

(b) Prove that this result cannot be strengthened so that condition (i) reads
   (i) (i′) for each $\mathbf{I} \models \Sigma$, $|\mathbf{I}(R)| = 0$ or $|\mathbf{I}(R)| = 1$ or $|\mathbf{I}(R)| = k$.

♠ **Exercise 9.18** [CKV90]

(a) Show that the set of inference rules containing {FD1, FD2, FD3, UIND1, UIND2} and

FD-UIND1: If $\emptyset \to A$ and $R[B] \subseteq R[A]$, then $\emptyset \to B$.
FD-UIND1: If $\emptyset \to A$ and $R[B] \subseteq R[A]$, then $R[A] \subseteq R[B]$.

is sound and complete for unrestricted logical implication of fd's and uind's over a single relation schema R.

(b) Generalize this result to arbitrary database schemas, under the assumption that in all instances, each relation is nonempty.