The Chase

Jef Wijsen

June 4, 2019

1 Full Dependencies

A *term* is either a constant or a variable. A *full dependency* can take two forms:

$$B \rightarrow H, \tag{1}$$
$$B \rightarrow s = t, \tag{2}$$

where B is a set of (not necessarily ground) atoms, H is an atom, s and t are terms, and every variable occurring at the right-hand of $\rightarrow$, also occurs in B. All predicates are edb predicates.

Full dependencies of the forms (1) and (2) are called *tuple-generating* (ftgd) and *equality-generating* (fegd), respectively. Functional dependencies are full equality-generating dependencies. Multivalued and join dependencies are full tuple-generating dependencies.[1]

Example 1 Persons are uniquely identified by their social security number. No person can have two distinct names, years of birth, or citizenships (column Nat). Every citizenship should occur in the list N.

P	$SS\#$	$Name$	$Birth$	Nat
	123	Smith	1964	USA
	456	Jones	1970	GB

N	Nat
	USA
	GB
	NL

$$\begin{aligned}
P(x, y_1, z_1, w_1), P(x, y_2, z_2, w_2) &\rightarrow y_1 = y_2\\
P(x, y_1, z_1, w_1), P(x, y_2, z_2, w_2) &\rightarrow z_1 = z_2\\
P(x, y_1, z_1, w_1), P(x, y_2, z_2, w_2) &\rightarrow w_1 = w_2\\
P(x, y, z, w) &\rightarrow N(w)
\end{aligned}$$

Notice that the three egds are equivalent to the functional dependency $P : \{SS\#\} \rightarrow \{Name, Birth, Nat\}$. $\triangleleft$

A database I *satisfies* (1) if for every valuation ν, if $\nu(B) \subseteq I$, then $\nu(H) \in I$. A database I *satisfies* (2) if for every valuation ν, if $\nu(B) \subseteq I$, then $\nu(s) = \nu(t)$. We write $I \models \sigma$ to denote that the database I satisfies the full dependency σ.

Let σ be a full dependency, and Σ a set of full dependencies. We write $I \models \Sigma$ to denote that I satisfies each full dependency of Σ. We say that Σ *logically implies* σ, denoted $\Sigma \models \sigma$, if for every database I, $I \models \Sigma$ implies $I \models \sigma$. We say that σ is *trivial* if it is satisfied by every database.

Example 2 Let

$$\begin{aligned}
\sigma_1 &: R(x, y_1, z_1), R(x, y_2, z_2) \rightarrow y_1 = y_2\\
\sigma_2 &: R(x, y, z'), R(x, y', z) \rightarrow R(x, y, z)
\end{aligned}$$

Then, $\{\sigma_1\} \models \sigma_2$, a result that is known as Heath's Theorem. $\triangleleft$

[1] See the course *Bases de Données I* for a definition of functional, multivalued, and join dependencies.

In the technical treatment, we will often need to identify two variables or to instantiate a variable by a constant. When identifying two variables v and w, one can choose between replacing all occurrences of v with w, or replacing all occurrences of w with v. To make this choice deterministic, we will assume a linear order $\prec$ on the set of all variables, and if we have to identify two variables v, w with $v \prec w$, we will replace all occurrences of w with v.

So we assume a linear order $\prec$ on the set of variables; we extend $\prec$ such that $c \prec v$ for every variable v and constant c. Therefore, a variable v can be replaced with a constant c, but not the other way around.

Let s, t be terms. If s, t are both constants, then $\mathsf{id}_{s=t}$ is the identity substitution if $s = t$, and $\mathsf{id}_{s=t}$ is undefined if $s \neq t$ (because we cannot identify distinct constants). If s, t are not both constants, then $\mathsf{id}_{s=t}$ is the substitution such that for every variable v,

$$\mathsf{id}_{s=t}(v) = \begin{cases} v & \text{if } v \notin \{s, t\}, \\ s & \text{if } v \in \{s, t\} \text{ and } s \preceq t, \\ t & \text{if } v \in \{s, t\} \text{ and } t \preceq s. \end{cases}$$

With this definition, $\mathsf{id}_{s=t}$ and $\mathsf{id}_{t=s}$ are the same substitution for all terms s, t that are not distinct constants. For example, if $t \prec y$, we have $\mathsf{id}_{t=y}(t) = \mathsf{id}_{t=y}(y) = t$.

2 The Chase

In this section, we develop an algorithm for the following problem:

PROBLEM:	Logical implication for full dependencies
INPUT:	a set Σ of full dependencies, a full dependency $L \rightarrow r$
QUESTION:	Does $\Sigma \models L \rightarrow r$ hold?

Note that in $L \rightarrow r$, the symbols L and r are used for the Left-hand and right-hand of the full dependency, respectively. The right-hand r is either an atom or an equality. If $L \rightarrow r$ is a trivial full dependency, then $\Sigma \models L \rightarrow r$ is obviously true.[2] In the following discussion, we can therefore assume that $L \rightarrow r$ is not trivial.

The *Chase* is an algorithm for the above logical implication problem that uses a simple strategy:

> Find a counterexample for $\Sigma \models L \rightarrow r$ (that is, a database I such that $I \models \Sigma$ and $I \not\models L \rightarrow r$), or decide that no such counterexample exists.

At first sight, this strategy may look ineffective since there are infinitely many distinct databases. However, it turns out that if there exists a counterexample, then such a counterexample can be easily computed from the canonical database of L.[3] Here, the *canonical database* is the database obtained from L by treating distinct variables as new distinct constants not occurring elsewhere.

It is correct to write $L \not\models L \rightarrow r$, where the symbol L preceding $\not\models$ is the canonical database. Indeed, the symbol r denotes either an atom not in L (because ftgds $L \rightarrow H$ are trivial if $H \in L$, and we assumed that $L \rightarrow r$ is not trivial) or an equality between two syntactically distinct terms (because fegds of the form $L \rightarrow t = t$ are trivial).

If $L \models \Sigma$, then the canonical database L is a counterexample for $\Sigma \models L \rightarrow r$, and the implication problem is solved. But what if $L \not\models \Sigma$? That is, what if Σ contains an ftgd $B \rightarrow H$ or an fegd $B \rightarrow s = t$ that is not satisfied by the canonical database L? In this situation, the Chase will arbitrarily pick a full dependency $\sigma \in \Sigma$ that is not satisfied by L, and minimally change L in order to make it satisfy σ. Such minimal change will be denoted by $\vdash_\Sigma$.

[2]It is an easy exercise to develop an algorithm that decides whether a full dependency $L \rightarrow r$ is trivial.

[3]Recall that canonical databases were also used for deciding containment of conjunctive queries.

Formally, we write

$$L \to r \vdash_\Sigma L' \to r'$$

if $L' \to r'$ can be derived from $L \to r$ by a single application of one of the following *chase rules*:

Chase rule for ftgd: for some ftgd $B \to H$ of Σ, for some substitution θ such that $\theta(B) \subseteq L$, we have $L' = L \cup \{\theta(H)\}$ and $r' = r$.

> *Intuition:* Think of L as a database that must be changed to satisfy $B \to H$. Then, if L contains all facts of $\theta(B)$, it must also contain $\theta(H)$. Therefore, we replace L with $L \cup \{\theta(H)\}$ to make it satisfy $B \to H$.

Chase rule for fegd: for some fegd $B \to s = t$ of Σ, for some substitution θ such that $\theta(B) \subseteq L$ and $\theta(s), \theta(t)$ are not distinct constants, we have $L' = \mathsf{id}_{\theta(s)=\theta(t)}(L)$ and $r' = \mathsf{id}_{\theta(s)=\theta(t)}(r)$.[4]

> *Intuition:* Think of L as a database that must be changed to satisfy $B \to s = t$. Then, if L contains all facts of $\theta(B)$, we must equalize $\theta(s)$ and $\theta(t)$. A problem obviously occurs if $\theta(s)$ and $\theta(t)$ are distinct constants, in which case $\mathsf{id}_{\theta(s)=\theta(t)}$ is undefined and the chase rule cannot be applied.

Example 3 We illustrate the fegd chase rule. Let σ_1 be the ftgd $R(x,y), R(y,z), S(x,y,z) \to R(x,z)$. Assume that Σ contains an fegd $R(u,v), R(v,w) \to w = a$ (call it σ_2), where a is a constant. Then, it is correct to write

$$R(x,y), R(y,z), S(x,y,z) \to R(x,z) \vdash_\Sigma R(x,y), R(y,a), S(x,y,a) \to R(x,a).$$

Indeed, the valuation $\theta = \{u \mapsto x, v \mapsto y, w \mapsto z\}$ maps the body of σ_2 into the body of σ_1, while $\theta(w) = z$ and $\theta(a) = a$ are not two distinct constants. Therefore, we apply $\mathsf{id}_{z=a}$ to σ_1, i.e., we replace all occurrences of z in σ_1 with a.

In the preceding paragraph, the fegd σ_2 was applied to the ftgd σ_1. Applying σ_2 to an fegd is not really different, for example,

$$R(x,y), R(y,z), S(x,y,z) \to z = x \vdash_\Sigma R(x,y), R(y,a), S(x,y,a) \to a = x,$$

where again it was assumed that $\sigma_2 \in \Sigma$. $\triangleleft$

Example 4 We now illustrate the ftgd chase rule. Let σ_1 be the ftgd $R(x,y), R(y,z), S(x,y,z) \to R(x,z)$. Assume that Σ contains an ftgd $R(u,v), R(v,w) \to R(w,u)$ (call it σ_3). Then, it is correct to write

$$R(x,y), R(y,z), S(x,y,z) \to R(x,z) \vdash_\Sigma R(x,y), R(y,z), R(z,x), S(x,y,z) \to R(x,z).$$

Indeed, the valuation $\theta = \{u \mapsto x, v \mapsto y, w \mapsto z\}$ maps the body of σ_3 into the body of σ_1. Therefore, we add $R(\theta(w), \theta(u)) = R(z,x)$ to the body of σ_1. Applying σ_3 to an fegd is not really different, for example,

$$R(x,y), R(y,z), S(x,y,z) \to z = x \vdash_\Sigma R(x,y), R(y,z), R(z,x), S(x,y,z) \to z = x,$$

where again it was assumed that $\sigma_3 \in \Sigma$. $\triangleleft$

The Chase algorithm repeatedly applies the above chase rules until no more changes can be made.

A *chase* of $L \to r$ by Σ is a maximal (in the sense that it cannot be extended) sequence $\sigma_0, \sigma_1, \ldots, \sigma_n$ such that $\sigma_0 = L \to r$ and for each $i \in \{1, 2, \ldots, n\}$, $\sigma_{i-1} \vdash_\Sigma \sigma_i$ and $\sigma_{i-1} \neq \sigma_i$.

It is an easy exercise to show that no chase can go on forever.[5]

One can show the following.

[4] Note that if r is an equality, say r is $t_1 = t_2$, then $\mathsf{id}_{\theta(s)=\theta(t)}(r)$ is the equality $\mathsf{id}_{\theta(s)=\theta(t)}(t_1) = \mathsf{id}_{\theta(s)=\theta(t)}(t_2)$.

[5] *Hint:* each term that occurs in some σ_i must already occur in σ_0.

Theorem 1 *Let Σ be a set of full dependencies, and let $L \to r$ be a full dependency. Let $L_n \to r_n$ be the last full dependency in a chase of $L \to r$ by Σ. If the canonical database L_n is not a counterexample for $\Sigma \models L \to r$, then $\Sigma \models L \to r$.*

Proof. Let $\sigma_0, \sigma_1, \ldots, \sigma_n$ be a chase of $L \to r$ by Σ. For every $i \in \{0, 1, \ldots, n\}$, let $\sigma_i = L_i \to r_i$, where r_i is either H_i (ftgd) or $s_i = t_i$ (fegd). In particular, $L_0 = L$ and $r_0 = r$.

The proof is by contraposition. Assume $\Sigma \not\models L \to r$. If suffices to show that the canonical database L_n satisfies Σ and falsifies $L \to r$.

Let I be a database that is a counterexample for $\Sigma \models L_0 \to r_0$, that is, $I \models \Sigma$ and $I \not\models L_0 \to r_0$ (there is at least one such I). Then, we can assume a valuation ν such that

$\nu(L_0) \subseteq I$	
Case $L_0 \to r_0$ is an ftgd: $\nu(H_0) \notin I$.	*Case $L_0 \to r_0$ is an fegd:* $\nu(s_0) \neq \nu(t_0)$.

(3)

We show that for every index $i \in \{0, 1, \ldots, n\}$,

$\nu(L_i) \subseteq I$	
Case $L_0 \to r_0$ is an ftgd: $\nu(H_i) = \nu(H_0)$.	*Case $L_0 \to r_0$ is an fegd:* $\nu(s_i) = \nu(s_0)$ and $\nu(t_i) = \nu(t_0)$.

(4)

Remark 1 (4) has a practical implication. If $L_i \to H_i$ is a trivial ftgd (i.e., $H_i \in L_i$), then it follows $\nu(H_0) = \nu(H_i) \in \nu(L_i) \subseteq I$, contradicting $\nu(H_0) \notin I$. Likewise, if $L_i \to s_i = t_i$ is a trivial fegd (i.e., s_i and t_i are the same term), then it follows $\nu(s_0) = \nu(s_i) = \nu(t_i) = \nu(t_0)$, contradicting $\nu(s_0) \neq \nu(t_0)$. Therefore, if $\Sigma \not\models L \to r$ (the assumption we made), the chase will never derive a trivial full dependency. Consequently, if a chase of $L \to r$ by Σ derives a trivial full dependency, then $\Sigma \models L \to r$. $\triangleleft$

The proof is by induction on increasing i. The basis of the induction, $i = 0$, is trivial. For the induction step, $i \to i + 1$, we distinguish two cases, depending on which chase rule was applied to derive σ_{i+1} from σ_i.

Case σ_{i+1} was derived by the ftgd chase rule. We can assume a ftgd $B \to H$ in Σ and a substitution θ such that $\theta(B) \subseteq L_i$ and $L_{i+1} = L_i \cup \{\theta(H)\}$ and $r_{i+1} = r_i$. Therefore,

$$\nu(\theta(B)) \subseteq \nu(L_i). \tag{5}$$

Since $\nu(L_i) \subseteq I$ by the induction hypothesis, (5) implies $\nu \circ \theta(B) \subseteq I$. From $I \models B \to H$ (because I satisfies all full dependencies in Σ), it follows $\nu \circ \theta(H) \in I$. It follows that the desired result holds for index $i + 1$:

$\nu(L_{i+1}) \subseteq I$	
Case $L_0 \to r_0$ is an ftgd: $H_{i+1} = H_i$ with $\nu(H_{i+1}) = \nu(H_0)$.	*Case $L_0 \to r_0$ is an fegd:* $s_{i+1} = s_i$ with $\nu(s_{i+1}) = \nu(s_0)$ and $t_{i+1} = t_i$ with $\nu(t_{i+1}) = \nu(t_0)$.

Case σ_{i+1} was derived by the fegd chase rule. We can assume a fegd $B \to s = t$ in Σ and a substitution θ such that $\theta(B) \subseteq L_i$ with $\theta(s), \theta(t)$ not distinct constants, $L_{i+1} = \mathsf{id}_{\theta(s)=\theta(t)}(L_i)$, and $r_{i+1} = \mathsf{id}_{\theta(s)=\theta(t)}(r_i)$. Therefore,

$$\nu(\theta(B)) \subseteq \nu(L_i). \tag{6}$$

Since $\nu(L_i) \subseteq I$ by the induction hypothesis, (6) implies $\nu \circ \theta(B) \subseteq I$. From $I \models B \rightarrow s = t$ (because I satisfies all full dependencies in Σ), it follows $\nu \circ \theta(s) = \nu \circ \theta(t)$. It can now be easily seen that $\nu(L_{i+1}) = \nu(L_i)$: intuitively, since ν maps the terms $\theta(s)$ and $\theta(t)$ to the same constant, it does not matter whether these terms have been identified (as in L_{i+1}) or not (as in L_i). Since $\nu(r_{i+1}) = \nu(r_i)$ holds by the same reasoning, it is correct to conclude that the desired result holds for index $i + 1$:

$\nu(L_{i+1}) \subseteq I$	
Case $L_0 \rightarrow r_0$ is an ftgd: $\nu(H_{i+1}) = \nu(H_i) = \nu(H_0)$.	*Case $L_0 \rightarrow r_0$ is an fegd:* $\nu(s_{i+1}) = \nu(s_i) = \nu(s_0)$ and $\nu(t_{i+1}) = \nu(t_i) = \nu(t_0)$.

So we have shown that (4) holds for all $i \in \{1, \ldots, n\}$. In particular, for $i = n$, we obtain:

$\nu(L_n) \subseteq I$	
Case $L_0 \rightarrow r_0$ is an ftgd: $\nu(H_n) = \nu(H_0)$.	*Case $L_0 \rightarrow r_0$ is an fegd:* $\nu(s_n) = \nu(s_0)$ and $\nu(t_n) = \nu(t_0)$.

(7)

Remark 2 Notice that I is an arbitrary counterexample for $\Sigma \models L \rightarrow r$. Therefore, (7) means that L_n is homomorphic to *every* database I that is a counterexample for $\Sigma \models L \rightarrow r$.[6] $\triangleleft$

We now show that the canonical database L_n falsifies $L_0 \rightarrow r_0$. Let μ be the composition of all substitutions $\mathrm{id}_{\theta(s)=\theta(t)}$ applied in all applications of the fegd rule. That is, μ applies the substitutions applied in any application of an fegd, in the same order as these substitutions were applied in the chase. It is an easy exercise to show the following.

$\mu(L_0) \subseteq L_n$	
Case $L_0 \rightarrow r_0$ is an ftgd: $H_n = \mu(H_0)$.	*Case $L_0 \rightarrow r_0$ is an fegd:* $s_n = \mu(s_0)$ and $t_n = \mu(t_0)$.

(8)

Assume towards a contradiction that $L_n \models L_0 \rightarrow r_0$. Since $\mu(L_0) \subseteq L_n$, it must be that

Case $L_0 \rightarrow r_0$ is an ftgd: $\mu(H_0) = H_n \in L_n$, hence $\nu(H_n) \in \nu(L_n)$. From (7), it follows $\nu(H_0) \in I$, contradicting (3).	*Case $L_0 \rightarrow r_0$ is an fegd:* $s_n = \mu(s_0) = \mu(t_0) = t_n$, hence $\nu(s_n) = \nu(t_n)$. From (7), it follows $\nu(s_0) = \nu(t_0)$, contradicting (3).

So it is correct to conclude, by contradiction, that $L_n \not\models L_0 \rightarrow r_0$.

We finally show that the canonical database L_n satisfies Σ. For every ftgd $B \rightarrow H$ in Σ, if $\theta(B) \subseteq L_n$ for some θ, then $\theta(H) \in L_n$, or else the chase would have continued. It follows that L_n satisfies every ftgd in Σ.

[6] A set B_1 of atoms is said to be homomorphic to a set B_2 of atoms if there exists a substitution h, called homomorphism, such that $h(B_1) \subseteq B_2$.

Assume next that Σ contains some fegd $B \rightarrow s = t$ such that $\theta(B) \subseteq L_n$ for some θ. Two cases can occur.

Case $\theta(s), \theta(t)$ are not distinct constants. Then $\theta(s) = \theta(t)$, or else the chase would have continued. It follows that $L_n \models B \rightarrow s = t$.

Case $\theta(s), \theta(t)$ are distinct constants. Let $\theta(s) = a$ and $\theta(t) = b$ with $a \neq b$. Then for I and ν introduced before, we have $\nu \circ \theta(B) \subseteq \nu(L_n) \subseteq I$. Since $I \models B \rightarrow s = t$, we have $\nu \circ \theta(s) = \nu \circ \theta(t)$, hence $a = \nu(a) = \nu(b) = b$, a contradiction. We conclude by contradiction that this case cannot occur.

From what precedes, it is correct to conclude that L_n satisfies Σ. This concludes the proof. □

The inverse of Theorem 1 holds obviously true: if the canonical database L_n is a counterexample for $\Sigma \models L \rightarrow r$, then $\Sigma \not\models L \rightarrow r$.

Example 5 Is it true that $\{\bowtie [AC, ABD], B \rightarrow C\} \models A \rightarrow C$? Let

$$\begin{aligned}
\sigma_1 &: R(x, y', z, w'), R(x, y, z', w) \rightarrow R(x, y, z, w) \\
\sigma_2 &: R(x_1, y, z_1, w_1), R(x_2, y, z_2, w_2) \rightarrow z_1 = z_2 \\
\sigma_3 &: R(x, y_1, z_1, w_1), R(x, y_2, z_2, w_2) \rightarrow z_1 = z_2
\end{aligned}$$

Obviously, $\sigma_1 \equiv \bowtie [AC, ABD]$, $\sigma_2 \equiv B \rightarrow C$, and $\sigma_3 \equiv A \rightarrow C$.

Here is a chase of σ_3 by $\Sigma = \{\sigma_1, \sigma_2\}$:

$$\begin{aligned}
\sigma_3 &: R(x, y_1, z_1, w_1), R(x, y_2, z_2, w_2) \rightarrow z_1 = z_2 \\
\text{Apply } \sigma_1 &: R(x, y_1, z_1, w_1), R(x, y_2, z_2, w_2), R(x, y_2, z_1, w_2) \rightarrow z_1 = z_2 \\
\text{Apply } \sigma_2 &: R(x, y_1, z_1, w_1), R(x, y_2, z_1, w_2) \rightarrow z_1 = z_1
\end{aligned}$$

Since the canonical database $\{R(x, y_1, z_1, w_1), R(x, y_2, z_1, w_2)\}$ satisfies σ_3, it is not a counterexample for $\{\sigma_1, \sigma_2\} \models \sigma_3$. By Theorem 1, it is correct to conclude $\{\sigma_1, \sigma_2\} \models \sigma_3$.

Alternatively, one can conclude $\{\sigma_1, \sigma_2\} \models \sigma_3$ from Remark 1 and the observation that the fegd with head $z_1 = z_1$ is trivial.

Notice that another chase is possible, in which σ_1 is applied twice:

$$\begin{aligned}
\sigma_3 &: R(x, y_1, z_1, w_1), R(x, y_2, z_2, w_2) \rightarrow z_1 = z_2 \\
\text{Apply } \sigma_1 &: R(x, y_1, z_1, w_1), R(x, y_2, z_2, w_2), R(x, y_2, z_1, w_2) \rightarrow z_1 = z_2 \\
\text{Apply again } \sigma_1 &: R(x, y_1, z_1, w_1), R(x, y_2, z_2, w_2), R(x, y_2, z_1, w_2), R(x, y_1, z_2, w_1) \rightarrow z_1 = z_2 \\
\text{Apply } \sigma_2 &: R(x, y_1, z_1, w_1), R(x, y_2, z_1, w_2) \rightarrow z_1 = z_1
\end{aligned}$$

Both chases end with the same trivial fegd. ◁

Example 6 Is it true that $\{\bowtie [ABC, AD, BC, CE]\} \models \bowtie [ABC, AD, BCE]$?

$$\begin{aligned}
\sigma_1 &: R(x, y, z, u_1, w_1), R(x, y_2, z_2, u, w_2), R(x_3, y, z, u_3, w_3), R(x_4, y_4, z, u_4, w) \rightarrow R(x, y, z, u, w) \\
\sigma_2 &: R(x, y, z, u_1, w_1), R(x, y_2, z_2, u, w_2), R(x_3, y, z, u_3, w) \rightarrow R(x, y, z, u, w)
\end{aligned}$$

We chase σ_2 by $\Sigma = \{\sigma_1\}$. Let θ be the substitution $\theta = \{w_3/w, x_4/x_3, y_4/y, u_4/u_3\}$, extended to be the identity on all other terms. Then, θ maps the body of σ_1 into the body of σ_2. Consequently, we derive a new full dependency by adding $\theta(R(x, y, z, u, w)) = R(x, y, z, u, w)$ to the body of σ_2:

$$R(x, y, z, u_1, w_1), R(x, y_2, z_2, u, w_2), R(x_3, y, z, u_3, w), R(x, y, z, u, w) \rightarrow R(x, y, z, u, w)$$

Since the latter full dependency is obviously trivial, we conclude $\{\sigma_1\} \models \sigma_2$. ◁

Example 7 Let $\Sigma = \{R(x) \rightarrow x = a, R(x) \rightarrow x = b\}$. Then,

$$R(v) \rightarrow S(v) \vdash_\Sigma R(a) \rightarrow S(a)$$

is a chase of $R(v) \rightarrow S(v)$ by Σ. Note that this chase cannot be extended:

- applying $R(x) \rightarrow x = a$ on $R(a) \rightarrow S(a)$ results again in $R(a) \rightarrow S(a)$, while it is required that $\sigma_{i-1} \neq \sigma_i$ in a chase;
- $R(x) \rightarrow x = b$ cannot be applied on $R(a) \rightarrow S(a)$, because a and b are distinct constants.

Since $\{R(a)\}$ is not a counterexample for $\Sigma \models R(v) \rightarrow S(v)$ (because $\{R(a)\} \not\models R(x) \rightarrow x = b$), it is correct to conclude $\Sigma \models R(v) \rightarrow S(v)$ by Theorem 1. Of course, $\Sigma \models R(v) \rightarrow S(v)$ can be easily shown without using Theorem 1: a database that satisfies Σ can contain no R-fact, and hence must necessarily satisfy $R(v) \rightarrow S(v)$ (because an implication is true if its premise is false).

Notice incidentally that there exists another chase of $R(v) \rightarrow S(v)$ by Σ:

$$R(v) \rightarrow S(v) \vdash_\Sigma R(b) \rightarrow S(b).$$

◁

Example 8 Let a, b be constants.

$$\begin{aligned} \sigma_1 &: R(x), S(y) \rightarrow x = y \\ \sigma_2 &: R(a), S(b) \rightarrow T(a, b) \end{aligned}$$

Do we have $\{\sigma_1\} \models \sigma_2$? A chase of σ_2 by $\Sigma = \{\sigma_1\}$ immediately ends with σ_2. Since $\{R(a), S(b)\}$ is not a counterexample for $\{\sigma_1\} \models \sigma_2$, we conclude that $\{\sigma_1\} \models \sigma_2$ by Theorem 1. Of course, $\{\sigma_1\} \models \sigma_2$ can be easily shown without using Theorem 1: a database that satisfies σ_1 cannot contain both $R(a)$ and $R(b)$, and hence must necessarily satisfy σ_2 (because an implication is true if its premise is false). ◁

Example 9 Let

$$\begin{aligned} \sigma_1 &: R(x, y) \rightarrow R(y, x), \\ \sigma_2 &: R(x, y), S(y, z), R(z, u), S(u, x) \rightarrow y = u, \\ \sigma_3 &: R(x, y), S(y, z), R(z, u), S(u, x) \rightarrow S(x, u). \end{aligned}$$

Let $\Sigma = \{\sigma_1, \sigma_2\}$. Do we have $\Sigma \models \sigma_3$? Here is a chase of σ_3 by $\{\sigma_1, \sigma_2\}$:

$$\begin{array}{lll} & R(x, y), S(y, z), R(z, u), S(u, x) \rightarrow S(x, u) & \\ \vdash_\Sigma & R(x, u), S(u, z), R(z, u), S(u, x) \rightarrow S(x, u) & \text{(Application of } \sigma_2\text{)} \\ \vdash_\Sigma & R(x, u), R(u, x), S(u, z), R(z, u), S(u, x) \rightarrow S(x, u) & \text{(Application of } \sigma_1\text{)} \\ \vdash_\Sigma & R(x, u), R(u, x), S(u, z), R(z, u), R(u, z), S(u, x) \rightarrow S(x, u) & \text{(Application of } \sigma_1\text{)} \end{array}$$

The canonical database $\{R(x, u), R(u, x), S(u, z), R(z, u), R(u, z), S(u, x)\}$ satisfies Σ and falsifies σ_3, hence $\Sigma \not\models \sigma_3$. ◁

3 Application: Optimization of Conjunctive Queries

Let $H \leftarrow B$ be a conjunctive query. If we treat the relation name in H as an edb predicate, then $B \rightarrow H$ is an ftgd that can be chased.

The chase of a conjunctive query $H \leftarrow L$ by a set Σ of full dependencies is completely analogous to the chase of the ftgd $L \rightarrow H$ by Σ. The only difference is the syntax: $H \leftarrow L$ for a query, $L \rightarrow H$ for an ftgd. Further, we often write $q : H \leftarrow L$ to make clear that q denotes the conjunctive query $H \leftarrow L$.

Theorem 2 *Let $q : H \leftarrow L$ be a conjunctive query. Let Σ be a set of full dependencies. Let $q_n : H_n \leftarrow L_n$ be the last element in a chase of q by Σ. Then, for every database I such that $I \models \Sigma$, we have $q(I) = q_n(I)$.*

Proof. Obviously, $\Sigma \not\models L \rightarrow H$, because the predicate in H is an idb predicate and thus does not occur in Σ. Let I be a database such that $I \models \Sigma$.

$\boxed{q(I) \subseteq q_n(I)}$ Let A be an arbitrary fact in $q(I)$. We can assume the existence of a valuation ν such that $\nu(L) \subseteq I$ and $\nu(H) = A$. We are in the same situation as in the proof of Theorem 1. Then, (7) implies that $A \in q_n(I)$.

$\boxed{q_n \sqsubseteq q}$ By (8), the substitution μ is a homomorphism from q to q_n. By the *Homomorphism Theorem*, we have $q_n \sqsubseteq q$. So $q_n(J) \subseteq q(J)$ holds for all databases J. □

The following example illustrates that the chase in combination with minimization of conjunctive queries provides a useful optimization technique.

Example 10 [1, Example 8.4.11]

$$\begin{aligned}
\Sigma &= \{B \rightarrow D, D \rightarrow C, \bowtie [AB, ACD]\} \\
q &: \mathsf{Answer}(w, x, z) \leftarrow R(w_1, x, y_1, z_1), R(w, x_2, y_1, z_1), R(w, x_3, y_3, z)
\end{aligned}$$

It can be easily seen that q is minimal. Here is a chase of q by Σ:

$$q_0 : \mathsf{Answer}(w, x, z) \leftarrow R(w_1, x, y_1, z_1), R(w, x_2, y_1, z_1), R(w, x_3, y_3, z)$$

$$\vdash_\Sigma q_1 : \mathsf{Answer}(w, x, z) \leftarrow R(w_1, x, y_1, z_1), R(w, x_2, y_1, z_1), R(w, x_3, y_3, z), R(w, x_2, y_3, z) \quad \text{(Application of } \bowtie [AB, ACD])$$

$$\vdash_\Sigma q_2 : \mathsf{Answer}(w, x, z) \leftarrow R(w_1, x, y_1, z_1), R(w, x_2, y_1, z_1), R(w, x_3, y_3, z), R(w, x_2, y_3, z), R(w, x_3, y_1, z_1) \quad \text{(Application of } \bowtie [AB, ACD])$$

$$\vdash_\Sigma q_3 : \mathsf{Answer}(w, x, z) \leftarrow R(w_1, x, y_1, z), R(w, x_2, y_1, z), R(w, x_3, y_3, z), R(w, x_2, y_3, z), R(w, x_3, y_1, z) \quad \text{(Application of } B \rightarrow D)$$

$$\vdash_\Sigma q_4 : \mathsf{Answer}(w, x, z) \leftarrow R(w_1, x, y_1, z), R(w, x_2, y_1, z), R(w, x_3, y_1, z) \quad \text{(Application of } D \rightarrow C)$$

Minimizing the conjunctive query q_4 results in (use the substitution that maps x_3 to x_2):

$$q' : \mathsf{Answer}(w, x, z) \leftarrow R(w_1, x, y_1, z), R(w, x_2, y_1, z)$$

From Theorem 2, it follows that $q(I) = q'(I)$ for all databases I that satisfy Σ (even though $q \not\equiv q'$). ◁

4 Exercises

1. Let

$$\begin{aligned}
\Sigma &= \left\{ \begin{array}{l} R(x, y), R(y, z) \rightarrow R(x, z), \\ R(x, y), R(x, z) \rightarrow y = z \end{array} \right\} \\
\sigma_1 &= R(u, v), R(v, w), R(w, z) \rightarrow R(u, u) \\
\sigma_2 &= R(u, v), R(v, w), R(w, z) \rightarrow R(z, z)
\end{aligned}$$

   Use Theorem 1 to show that $\Sigma \not\models \sigma_1$ and $\Sigma \models \sigma_2$.

Answer. Here is a chase of σ_1 by Σ.

$$\begin{array}{ll} & R(u,v), R(v,w), R(w,z) \rightarrow R(u,u) \\ \vdash_\Sigma & R(u,v), R(v,w), R(w,z), R(u,w) \rightarrow R(u,u) \\ \vdash_\Sigma & R(u,v), R(v,v), R(v,z) \rightarrow R(u,u) \\ \vdash_\Sigma & R(u,v), R(v,v) \rightarrow R(u,u) \end{array}$$

The canonical database $\{R(u,v), R(v,v)\}$ satisfies Σ but falsifies σ_1, hence $\Sigma \not\models \sigma_1$. To see that $\{R(u,v), R(v,v)\}$ falsifies σ_1, notice that $\{u \mapsto u, v \mapsto v, w \mapsto v, z \mapsto v\}$ maps the body of σ_1 into $\{R(u,v), R(v,v)\}$, but maps the head of σ_1 to $R(u,u) \notin \{R(u,v), R(v,v)\}$.

Here is a chase of σ_2 by Σ.

$$\begin{array}{ll} & R(u,v), R(v,w), R(w,z) \rightarrow R(z,z) \\ \vdash_\Sigma & R(u,v), R(v,w), R(w,z), R(u,w) \rightarrow R(z,z) \\ \vdash_\Sigma & R(u,v), R(v,v), R(v,z) \rightarrow R(z,z) \\ \vdash_\Sigma & R(u,v), R(v,v) \rightarrow R(v,v) \end{array}$$

The canonical database $\{R(u,v), R(v,v)\}$ satisfies $\Sigma \cup \{\sigma_2\}$, and hence is not a counterexample for $\Sigma \models \sigma_2$. From Theorem 1, it follows $\Sigma \models \sigma_2$. Alternatively, $\Sigma \models \sigma_2$ follows from Remark 1 and the observation that $R(u,v), R(v,v) \rightarrow R(v,v)$ is trivial.

2. Show that if $\sigma_0, \sigma_1, \ldots, \sigma_n$ is a chase of σ by Σ, then for all $i, j \in \{1, 2, \ldots, n\}$, $i \neq j$ implies $\sigma_i \neq \sigma_j$. Without this property, the chase could be trapped in an infinite loop.

3. Let Σ be a set of full dependencies. Let $L \rightarrow r$ be a full dependency such that $\Sigma \not\models L \rightarrow r$.

   Assume that $L' \rightarrow r'$ is the last element in a chase of $L \rightarrow r$ by Σ. Assume that $L'' \rightarrow r''$ is the last element in another chase of $L \rightarrow r$ by Σ. Note that L' and L'' could be different, because two chases can apply chase rules in a different order. Use Remark 2 to show that L' and L'' are homomorphic to one another.

4. Show that if $\Sigma \not\models L \rightarrow r$, then the counterexample found by a chase is not necessarily the smallest (with respect to cardinality) counterexample possible.

   Answer. Take $\Sigma = \{R(u,w) \rightarrow R(w,u)\}$. Then,

   $$R(x,y) \rightarrow x = a \vdash_\Sigma R(x,y), R(y,x) \rightarrow x = a$$

   is a chase of $R(x,y) \rightarrow x = a$ by Σ. The canonical database $\{R(x,y), R(y,x)\}$ is a counterexample for $\Sigma \models R(x,y) \rightarrow x = a$.

   Note that $\{R(b,b)\}$ is a smaller counterexample for $\Sigma \models R(x,y) \rightarrow x = a$. Note also that the counterexample $\{R(x,y), R(y,x)\}$ found by the chase is homomorphic to $\{R(b,b)\}$, confirming Remark 2.

5. Show that $\{A \rightarrow C, B \rightarrow C, C \rightarrow D, DE \rightarrow C, CE \rightarrow A\} \models \bowtie [AD, AB, BE, CDE, AE]$, where the set of attributes is $ABCDE$.

Answer (Sketch). To simplify notation, we represent the ftgd $\bowtie [AD, AB, BE, CDE, AE]$ as a table.

A	B	C	D	E
x_0	y_1	z_1	u_0	w_1
x_0	y_0	z_2	u_2	w_2
x_3	y_0	z_3	u_3	w_0
x_4	y_4	z_0	u_0	w_0
x_0	y_5	z_5	u_5	w_0
x_0	y_0	z_0	u_0	w_0

$\vdash_\Sigma$

A	B	C	D	E
x_0	y_1	z_1	u_0	w_1
x_0	y_0	z_1	u_2	w_2
x_3	y_0	z_3	u_3	w_0
x_4	y_4	z_0	u_0	w_0
x_0	y_5	z_1	u_5	w_0
x_0	y_0	z_0	u_0	w_0

$\vdash_\Sigma$

A	B	C	D	E
x_0	y_1	z_1	u_0	w_1
x_0	y_0	z_1	u_2	w_2
x_3	y_0	z_1	u_3	w_0
x_4	y_4	z_0	u_0	w_0
x_0	y_5	z_1	u_5	w_0
x_0	y_0	z_0	u_0	w_0

$\vdash_\Sigma$

A	B	C	D	E
x_0	y_1	z_1	u_0	w_1
x_0	y_0	z_1	u_0	w_2
x_3	y_0	z_1	u_0	w_0
x_4	y_4	z_0	u_0	w_0
x_0	y_5	z_1	u_0	w_0
x_0	y_0	z_0	u_0	w_0

$\vdash_\Sigma$

A	B	C	D	E
x_0	y_1	z_1	u_0	w_1
x_0	y_0	z_1	u_0	w_2
x_3	y_0	z_0	u_0	w_0
x_4	y_4	z_0	u_0	w_0
x_0	y_5	z_0	u_0	w_0
x_0	y_0	z_0	u_0	w_0

$\vdash_\Sigma$

A	B	C	D	E
x_0	y_1	z_1	u_0	w_1
x_0	y_0	z_1	u_0	w_2
x_0	y_0	z_0	u_0	w_0
x_0	y_4	z_0	u_0	w_0
x_0	y_5	z_0	u_0	w_0
x_0	y_0	z_0	u_0	w_0

$\vdash_\Sigma$

A	B	C	D	E
x_0	y_1	z_0	u_0	w_1
x_0	y_0	z_0	u_0	w_2
x_0	y_0	z_0	u_0	w_0
x_0	y_4	z_0	u_0	w_0
x_0	y_5	z_0	u_0	w_0
x_0	y_0	z_0	u_0	w_0

We applied, respectively, $A \to C$, $B \to C$, $C \to D$, $DE \to C$, $CE \to A$, $A \to C$. Notice that the sixth ftgd is already trivial (the body contains the head $\langle x_0, y_0, z_0, u_0, w_0 \rangle$); at that point, we could already have stopped the chase, because Remark 1 tells us that no counterexample will be found.

6. Let

$$\begin{aligned} \Sigma &= \{A \to B, \bowtie [BC, ABD]\}, \\ q &: \mathsf{Answer}(u, x, y, z) \leftarrow R(u, x, y_0, z), R(u, x_0, y, z_0). \end{aligned}$$

Simplify q knowing that it is applied only on databases satisfying Σ.

Answer. Here is a chase of q by Σ.

$$\begin{aligned} & q_0 : \mathsf{Answer}(u, x, y, z) \leftarrow R(u, x, y_0, z), R(u, x_0, y, z_0) \\ \vdash_\Sigma\ & q_1 : \mathsf{Answer}(u, x, y, z) \leftarrow R(u, x, y_0, z), R(u, x, y, z_0) \\ \vdash_\Sigma\ & q_2 : \mathsf{Answer}(u, x, y, z) \leftarrow R(u, x, y_0, z), R(u, x, y, z_0), R(u, x, y_0, z_0) \\ \vdash_\Sigma\ & q_3 : \mathsf{Answer}(u, x, y, z) \leftarrow R(u, x, y_0, z), R(u, x, y, z_0), R(u, x, y_0, z_0), R(u, x, y, z) \end{aligned}$$

If we minimize the final query of the chase, we obtain:

$$q_3' : \mathsf{Answer}(u, x, y, z) \leftarrow R(u, x, y, z).$$

On databases satisfying Σ, the queries q and q_3' return the same answer. Note that q_3' simply returns all tuples of R.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.