The Boyce-Codd-Heath Normal Form for SQL

Flavio Ferrarotti[1], Sven Hartmann[2], Henning Köhler[3], Sebastian Link[1], and Millist Vincent[4]

[1] School of Information Management, Victoria University of Wellington, New Zealand
[2] Institut für Informatik, Technische Universität Clausthal, Germany
[3] School of Information Technology & Electrical Engineering, University of Queensland, Australia
[4] School of Computer and Information Science, University of South Australia, Australia

Abstract. In the relational model of data the Boyce-Codd-Heath normal form condition guarantees the elimination of data redundancy in terms of functional dependencies. For efficient means of data processing the industry standard SQL permits partial data and duplicate rows of data to occur in database systems. Consequently, the combined class of uniqueness constraints and functional dependencies is more expressive than the class of functional dependencies itself. Hence, the Boyce-Codd-Heath normal form condition is not suitable for SQL databases. We characterize the associated implication problem of the combined class in the presence of NOT NULL constraints axiomatically, algorithmically and logically. Based on these results we are able to establish a suitable normal form condition for SQL.

1 Introduction

In the *relational model of data* [7] a relation schema R denotes a finite set of attributes A that have a countably infinite domain $dom(A)$. A relation over R is a finite set of tuples, i.e. elements of the cartesian product over the domains. In addition, constraints restrict relations to those considered meaningful for the application. A functional dependency (FD) over R is an expression $X \to Y$ with $X, Y \subseteq R$. It restricts relations to those where every pair of tuples with the same values on all the attributes in X also has the same values on all the attributes in Y. FDs are essential for database design and data processing: if there is an FD $X \to Y$ over R with $Y \not\subseteq X$, then either all the attributes of $R - XY$ are also functionally dependent on X or there are relations with redundant data value occurrences. Redundancy can lead to inefficiencies with updates. A relation schema R is in *Boyce-Codd-Heath normal form* (BCHNF) [8, 14, 19] with respect to a given set Σ of FDs if for every FD $X \to Y$ in Σ, $Y \subseteq X$ or the FD $X \to R$ is implied by Σ. Hence, if R is in BCHNF, then no *set* of tuples over R contains two elements with the same values on all the attributes in X.

Example 1. Consider the relation schema SCHEDULE with attributes *Location*, *Time*, and *Speaker*, and FD set Σ consisting of *Location, Time* $\to$ *Speaker*, and

Speaker, Time → *Location*. Then Schedule is in BCHNF with respect to Σ. The following relation r on the left is an *Armstrong relation* for Σ. That is, r satisfies all the FDs in Σ and violates all the FDs not implied by Σ.

relation r

Location	*Time*	*Speaker*
Green Room	10am	Hilbert
Blue Room	10am	Gauss
Red Room	11am	Gauss
Red Room	01pm	Grothendieck
Red Room	02pm	Grothendieck

table t

Location	*Time*	*Speaker*
Green Room	10am	`ni`
Blue Room	10am	Gauss
Red Room	11am	Gauss
Red Room	01pm	Grothendieck
Red Room	02pm	Grothendieck
Red Room	02pm	Grothendieck

No data value occurrence in r is *redundant*: if we conceal any single value, then the remaining values and the FDs do not determine the concealed value. □

Commercial database systems deviate from the relational model of data. In the data definition and query standard *SQL* [9] database instances are tables where the column headers of the table correspond to attributes. The rows of the table correspond to tuples, but a table can contain different rows that have the same value in every column. Hence, an SQL table is a *bag* of rows. This feature lowers the cost of data processing as duplicate elimination is considered expensive. Furthermore, a so-called *null value*, marked `ni`, can occur in any column of any row in an SQL table. The null value indicates either non-existing, or existing but unknown, information. This feature of SQL makes it easy to enter new information into the database, since information is not always complete in practice. Null value occurrences can be forbidden for entire columns by declaring the corresponding column header `NOT NULL`. With these new features in mind we now revisit Example 1.

Example 2. Consider the SQL table Schedule from Example 1 with the same set Σ of constraints and where the column headers *Time* and *Location* are `NOT NULL`. The SQL table t from Example 1 on the right is an Armstrong table for Σ and the `NOT NULL` constraints. The BCHNF condition does not guarantee the absence of redundant data value occurrences over SQL tables. For example, the value of *Grothendieck* in the last row of table t is redundant: it is determined by the remaining values in the table t and the FD *Location, Time* → *Speaker*. □

Another important class of constraints are *uniqueness constraints* (UCs). The UC *unique*(X) restricts tables to those that do not have two distinct rows that are non-null and equal on every attribute in X. In the relational model UCs are not studied separately because any set of tuples over R satisfies the UC *unique*(X) if and only if it satisfies the FD $X \rightarrow R$. However, this equivalence no longer holds over SQL tables, as illustrated in Example 2. Indeed, if $X = \{\mathit{Location}, \mathit{Time}\}$, then table t satisfies $X \rightarrow$ Schedule, but not *unique*(X). This means that, in the context of SQL tables, the combined class of UCs and FDs should be studied, preferably in the context of `NOT NULL` constraints. Moreover, Example

2 motivates our pursuit of a normal form condition for SQL table definitions that eliminates redundant data value occurrences.

Contributions and Organization. We summarize previous work in Section 2 and give preliminary definitions in Section 3. In Section 4 we establish a finite axiomatization for the combined class of UCs and FDs in the presence of `NOT NULL` constraints. In Section 5 we show that the implication problem of this class is equivalent to that of goal and definite clauses in Cadoli and Schaerf's para-consistent family of $\mathcal{S}$-3 logics. In Section 6 we propose a new syntactic normal form condition for SQL table definitions. Finally, in Section 7 we justify our condition semantically by showing that it is necessary and sufficient for the absence of redundant data value occurrences in any SQL tables. We also show that our condition can be checked in time quadratic in the input, and is independent of the representation of the constraints. We conclude in Section 8.

2 Related Work

Data dependencies and normalization are essential to the design of the target database, the maintenance of the database during its lifetime, and all major data processing tasks, cf. [1].

In the relational model, a UC $unique(X)$ over relation schema R is satisfied by a relation if and only if the relation satisfies the FD $X \rightarrow R$. Hence, in this context it suffices to study the class of FDs alone. Armstrong [4] established the first axiomatization for FDs. The implication problem of FDs can be decided in time linear in the input [10]. Boyce and Codd [8] and Heath [14] introduced what is now known as the Boyce-Codd-Heath normal form for relation schemata. Vincent showed that BCHNF is a sufficient and necessary condition to eliminate all possible redundant data value occurrences as well as data processing difficulties in terms of FDs [22]. Arenas and Libkin also justified the BCHNF condition in terms of information-theoretic measures [3].

One of the most important extensions of Codd's basic relational model [7] is incomplete information [15]. This is mainly due to the high demand for the correct handling of such information in real-world applications. While there are several possible interpretations of a null value, most of the previous work on data dependencies is based on Zaniolo's no information interpretation [24]. Atzeni and Morfuni established an axiomatization of FDs in the presence of `NOT NULL` constraints under the no information interpretation [5]. They did not consider bags, which commonly appear in SQL, nor normalization. Köhler and Link investigated UCs and FDs over bags, but did not allow null values [17]. Finally, Hartmann and Link established the equivalence of the implication problem for the combined class of FDs and multivalued dependencies in the presence of `NOT NULL` constraints to that of a propositional fragment of Cadoli and Schaerf's family of $\mathcal{S}$-3 logics [13]. However, they only looked at relations where UCs are subsumed by FDs and did not consider bags. The equivalences cover those by Sagiv et al. [20] established for the special case where $\mathcal{S}$ covers all variables.

3 SQL table definitions

We summarize the basic notions. Let $\mathfrak{A} = \{H_1, H_2, \ldots\}$ be a (countably) infinite set of distinct symbols, called (column) headers. An *SQL table definition* is a finite non-empty subset T of $\mathfrak{A}$. Each header H of a table definition T is associated with a countably infinite domain $dom(H)$ which represents the possible values that can occur in the column H denotes. To encompass incomplete information every column may have a null value, denoted by $\texttt{ni} \in dom(H)$. The intention of $\texttt{ni}$ is to mean "no information". This interpretation can therefore model non-existing as well as existing but unknown information [5, 24].

For header sets X and Y we may write XY for $X \cup Y$. If $X = \{H_1, \ldots, H_m\}$, then we may write $H_1 \cdots H_m$ for X. In particular, we may write simply H to represent the singleton $\{H\}$. A *row* over T (T-row or simply row, if T is understood) is a function $r : T \rightarrow \bigcup\limits_{H \in T} dom(H)$ with $r(H) \in dom(H)$ for all $H \in R$. The null value occurrence $r(H) = \texttt{ni}$ associated with a header H in a row r means that no information is available about the header H for the row r. For $X \subseteq T$ let $r[H]$ denote the restriction of the row r over T to X. An *SQL table* t over T is a finite multi-set of rows over R. For a row r over T and a set $X \subseteq T$, r is said to be X-total if for all $H \in X$, $r(H) \neq \texttt{ni}$. Similar, a table t over T is said to be X-total, if every row r of t is X-total. A table t over T is said to be a *total table* if it is T-total.

Following the SQL standard a *uniqueness constraint* (UC) over an SQL table definition T is an expression $unique(X)$ where $X \subseteq T$. An SQL table t over T is said to satisfy the uniqueness constraint $unique(X)$ over T ($\models_t unqiue(X)$) if and only if for all distinct rows $r_1, r_2 \in t$ the following holds: if r_1 and r_2 are X-total, then there is some $H \in X$ such that $r_1(H) \neq r_2(H)$.

Functional dependencies are important for the relational [7] and other data models [2, 11, 12, 23]. Following Lien [18], a *functional dependency* (FD) over T is a statement $X \rightarrow Y$ where $X, Y \subseteq T$. The FD $X \rightarrow Y$ over T is satisfied by a table t over T ($\models_t X \rightarrow Y$) if and only if for all $r_1, r_2 \in t$ the following holds: if r_1 and r_2 are X-total and $r_1[X] = r_2[X]$, then $r_1[Y] = r_2[Y]$. We call $X \rightarrow Y$ *trivial* whenever $Y \subseteq X$, and non-trivial otherwise. For total tables the FD definition reduces to the standard definition of a functional dependency [1], and so is a sound generalization. It is also consistent with the no-information interpretation [5, 18].

Following Atzeni and Morfuni [5], a *null-free sub-definition* (NFS) over the table definition T is a an expression T_s where $T_s \subseteq T$. The NFS T_s over T is satisfied by a table t over T ($\models_t T_s$) if and only if t is T_s-total. SQL allows the specification of column headers as `NOT NULL`. Hence, the set of headers declared `NOT NULL` forms the single NFS over the underlying SQL table definition.

For a set Σ of constraints over some table definition T, we say that a table t over T *satisfies* Σ ($\models_t \Sigma$) if t satisfies every $\sigma \in \Sigma$. If for some $\sigma \in \Sigma$ the table t does not satisfy σ we say that t violates σ (and violates Σ) and write $\not\models_t \sigma$ ($\not\models_t \Sigma$). We are interested in the combined class $\mathcal{C}$ of uniqueness constraints and FDs in the presence of an NFS.

Constraints interact with one another. Let T be an SQL table definition, let $T_s \subseteq T$ denote an NFS over T, and let $\Sigma \cup \{\varphi\}$ be a set of uniqueness constraints and FDs over T. We say that Σ *implies* φ in the presence of T_s ($\Sigma \models_{T_s} \varphi$) if every table t over T that satisfies Σ and T_s also satisfies φ. If Σ does not imply φ in the presence of T_s we may also write $\Sigma \not\models_{T_s} \varphi$. For Σ we let $\Sigma^*_{T_s} = \{\varphi \mid \Sigma \models_{T_s} \varphi\}$ be the *semantic closure* of Σ, i.e., the set of all uniqueness constraints and FDs implied by Σ in the presence of T_s. In order to determine the logical consequences we use a syntactic approach by applying inference rules, e.g. those in Table 1. These inference rules have the form

$$\frac{\text{premise}}{\text{conclusion}}\,\text{condition},$$

and inference rules without any premises are called axioms. An inference rule is called sound, if whenever the set of constraints in the premise of the rule and the NFS are satisfied by some table over T and the constraints and NFS satisfy the conditions of the rule, then the table also satisfies the constraint in the conclusion of the rule. We let $\Sigma \vdash_{\mathfrak{R}} \varphi$ denote the *inference* of φ from Σ by $\mathfrak{R}$. That is, there is some sequence $\gamma = [\sigma_1, \ldots, \sigma_n]$ of constraints such that $\sigma_n = \varphi$ and every σ_i is an element of Σ or results from an application of an inference rule in $\mathfrak{R}$ to some elements in $\{\sigma_1, \ldots, \sigma_{i-1}\}$. For a finite set Σ, let $\Sigma^+_{\mathfrak{R}} = \{\varphi \mid \Sigma \vdash_{\mathfrak{R}} \varphi\}$ be its *syntactic closure* under inferences by $\mathfrak{R}$. A set $\mathfrak{R}$ of inference rules is said to be *sound* (*complete*) for the implication of uniqueness constraints and FDs in the presence of an NFS if for every table definition T, for every NFS T_s over T and for every set Σ of uniqueness constraints and FDs over T we have $\Sigma^+_{\mathfrak{R}} \subseteq \Sigma^*_{T_s}$ ($\Sigma^*_{T_s} \subseteq \Sigma^+_{\mathfrak{R}}$). The (finite) set $\mathfrak{R}$ is said to be a (finite) *axiomatization* for the implication of uniqueness constraints and FDs in the presence of an NFS if $\mathfrak{R}$ is both sound and complete.

Example 3. The SQL table in Example 2 satisfies the FD *Location, Time* → *Speaker*, but violates the UC *unique*(*Location, Time*). The table

Location	*Time*	*Speaker*
Red Room	`ni`	Gauss
Red Room	`ni`	Grothendieck

satisfies the NFS $\{$*Location, Speaker*$\}$, the UC *unique*(*Location, Time*) and the FDs *Location* → *Time* and *Time* → *Speaker*. The table violates the NFS $\{$*Time*$\}$, the UC *unique*(*Location*) and the FD *Location* → *Speaker*. □

4 Axiomatic and algorithmic characterization

Let $\mathfrak{S}$ denote the set of inference rules in Table 1. The soundness of the rules in $\mathfrak{S}$ is not difficult to show. For the completeness of $\mathfrak{S}$ we use the result that the set $\mathfrak{M}$ consisting of the reflexivity axiom, the union, decomposition and null transitivity rule is sound and complete for FDs in the presence of an NFS [5]. In fact, the completeness of $\mathfrak{S}$ follows from that of $\mathfrak{M}$ and the following

$\dfrac{\mathit{unique}(X)}{X \to Y}$ (demotion)	$\dfrac{}{XY \to X}$ (reflexivity)	$\dfrac{X \to YZ}{X \to Y}$ (decomposition)
$\dfrac{X \to Y \quad \mathit{unique}(Y)}{\mathit{unique}(X)}\, Y \subseteq XT_s$ (null pullback)	$\dfrac{X \to Y \quad Y \to Z}{X \to Z}\, Y \subseteq XT_s$ (null transitivity)	$\dfrac{X \to Y \quad X \to Z}{X \to YZ}$ (union)

Table 1. Axiomatization of UCs and FDs in the presence of an NFS.

lemma. For a set $\Sigma = \Sigma_{\text{UC}} \cup \Sigma_{\text{FD}}$ of UCs and FDs over table definition T let $\Sigma_{\text{UC}}^{\text{FD}} = \{X \to T \mid \mathit{unique}(X) \in \Sigma_{\text{UC}}\}$ be the set of FDs associated with Σ_{UC} and let $\Sigma[\text{FD}] := \Sigma_{\text{UC}}^{\text{FD}} \cup \Sigma_{\text{FD}}$ be the set of FDs associated with Σ.

Lemma 1. *Let T be an SQL table definition, T_s an NFS, and Σ a set of UCs and FDs over T. Then the following hold:*

1. *$\Sigma \models_{T_s} X \to Y$ if and only if $\Sigma[FD] \models_{T_s} X \to Y$,*
2. *$\Sigma \models_{T_s} \mathit{unique}(X)$ if and only if $\Sigma[FD] \models_{T_s} X \to T$ and there is some $\mathit{unique}(Z) \in \Sigma$ such that $Z \subseteq XT_s$.* □

Theorem 1. *The set $\mathfrak{S}$ is a finite axiomatization for the implication of UCs and FDs in the presence of an NFS.* □

Lemma 1 establishes an algorithmic characterization of the associated implication problem. In fact, it suffices to compute the *header set closure* $X^*_{\Sigma[\text{FD}],T_s} := \{H \in T \mid \Sigma[\text{FD}] \models_{T_s} X \to H\}$ of X with respect to $\Sigma[\text{FD}]$ and T_s [5]. The size $|\varphi|$ of φ is the total number of attributes occurring in φ, and the size $||\Sigma||$ of Σ is the sum of $|\sigma|$ over all elements $\sigma \in \Sigma$.

Theorem 2. *The problem whether a UC or FD φ is implied by a set Σ of UCs and FDs can be decided in $\mathcal{O}(||\Sigma \cup \{\varphi\}||)$ time.* □

5 Equivalence to goal and definite clauses in $\boldsymbol{\mathcal{S}}$-3 logics

Here we refine the correspondence between the implication of FDs in the presence of NFSs and the implication of Horn clauses in Cadoli and Schaerf's family of $\mathcal{S}$-3 logics, established for tables that are sets of rows [13].

$\boldsymbol{\mathcal{S}}$-3 semantics. Schaerf and Cadoli [21] introduced $\mathcal{S}$-3 logics as "a semantically well-founded logical framework for sound approximate reasoning, which is justifiable from the intuitive point of view, and to provide fast algorithms for dealing with it even when using expressive languages".

For a finite set $\mathcal{L}$ of propositional variables let $\mathcal{L}^\ell$ denote the set of all literals over $\mathcal{L}$, i.e., $\mathcal{L}^\ell = \mathcal{L} \cup \{\neg H' \mid H' \in \mathcal{L}\} \subseteq \mathcal{L}^*$ where $\mathcal{L}^*$ denotes the propositional language over $\mathcal{L}$. Let $\mathcal{S} \subseteq \mathcal{L}$. An $\mathcal{S}$-3 interpretation of $\mathcal{L}$ is a total function

$\hat{\omega} : \mathcal{L}^{\ell} \to \{\mathbb{F}, \mathbb{T}\}$ that maps every variable $H' \in \mathcal{S}$ and its negation $\neg H'$ into opposite values ($\hat{\omega}(H') = \mathbb{T}$ if and only if $\hat{\omega}(\neg H') = \mathbb{F}$), and that does not map both a variable $H' \in \mathcal{L} - \mathcal{S}$ and its negation $\neg H'$ into $\mathbb{F}$ (we must not have $\hat{\omega}(H') = \mathbb{F} = \hat{\omega}(\neg H')$ for any $H' \in \mathcal{L} - \mathcal{S}$). An $\mathcal{S}$-3 interpretation $\hat{\omega} : \mathcal{L}^{\ell} \to \{\mathbb{F}, \mathbb{T}\}$ of $\mathcal{L}$ can be lifted to a total function $\hat{\Omega} : \mathcal{L}^* \to \{\mathbb{F}, \mathbb{T}\}$ by means of simple rules [21]. Since we are only interested in Horn clauses here we require the following two rules for assigning truth values to a Horn clause: (1) $\hat{\Omega}(\varphi') = \hat{\omega}(\varphi')$, if $\varphi' \in \mathcal{L}^{\ell}$, and (2) $\hat{\Omega}(\varphi' \vee \psi') = \mathbb{T}$, if $\hat{\Omega}(\varphi') = \mathbb{T}$ or $\hat{\Omega}(\psi') = \mathbb{T}$. An $\mathcal{S}$-3 interpretation $\hat{\omega}$ is a *model* of a set Σ' of $\mathcal{L}$-formulae, if $\hat{\Omega}(\sigma') = \mathbb{T}$ holds for every $\sigma' \in \Sigma'$. We say that Σ' *$\mathcal{S}$-3 implies* an $\mathcal{L}$-formula φ', denoted by $\Sigma' \models^3_{\mathcal{S}} \varphi'$, if every $\mathcal{S}$-3 interpretation that is a model of Σ' is also a model of φ'.

Mappings between constraints and formulae. In the first step, we define the fragment of $\mathcal{L}$-formulae that corresponds to UCs and FDs in the presence of an NFS T_s over a table definition T. Let $\phi : T \to \mathcal{L}$ denote a bijection between T and the set $\mathcal{L} = \{H' \mid H \in T\}$ of propositional variables that corresponds to T. For an NFS T_s over T let $\mathcal{S} = \phi(T_s)$ be the set of propositional variables in $\mathcal{L}$ that corresponds to T_s. Hence, the variables in S are the images of those column headers of T declared `NOT NULL`. We now extend ϕ to a mapping Φ from the set of UCs and FDs over T. For a UC $\mathit{unique}(H_1, \ldots, H_n)$ over T, let $\Phi(\mathit{unique}(H_1, \ldots, H_n))$ denote the goal clause $\neg H'_1 \vee \cdots \vee \neg H'_n$. For an FD $H_1, \ldots, H_n \to H$ over T, let $\Phi(H_1, \ldots, H_n \to H)$ denote the definite clause $\neg H'_1 \vee \cdots \vee \neg H'_n \vee H'$. For the sake of presentation, but without loss of generality, we assume that FDs have only a single column header on their right-hand side. As usual, disjunctions over zero disjuncts are interpreted as $\mathbb{F}$. In what follows, we may simply denote $\Phi(\varphi) = \varphi'$ and $\Phi(\Sigma) = \{\sigma' \mid \sigma \in \Sigma\} = \Sigma'$.

The equivalence. Our aim is to show that for every SQL table definition T, for every set $\Sigma \cup \{\varphi\}$ of UCs and FDs and for every NFS T_s over T, there is some T_s-total table t that satisfies Σ and violates φ if and only if there is an $\mathcal{S}$-3 model $\hat{\omega}_t$ of Σ' that is not an $\mathcal{S}$-3 model of φ'. For an arbitrary table t it is not obvious how to define the $\mathcal{S}$-3 interpretation $\hat{\omega}_t$. However, for deciding the implication problem $\Sigma \models_{T_s} \varphi$ it suffices to examine two-row tables, instead of arbitrary tables. For two-row tables $\{r_1, r_2\}$ we define the *special-3-interpretation* of $\mathcal{L}$ by

- $\hat{\omega}_{\{r_1,r_2\}}(H') = \mathbb{T}$ and $\hat{\omega}_{\{r_1,r_2\}}(\neg H') = \mathbb{F}$, if $\texttt{ni} \neq r_1(H) = r_2(H) \neq \texttt{ni}$,
- $\hat{\omega}_{\{r_1,r_2\}}(H') = \mathbb{T}$ and $\hat{\omega}_{\{r_1,r_2\}}(\neg H') = \mathbb{T}$, if $r_1(H) = \texttt{ni} = r_2(H)$,
- $\hat{\omega}'_{\{r_1,r_2\}}(H') = \mathbb{F}$ and $\hat{\omega}'_{\{r_1,r_2\}}(\neg H') = \mathbb{T}$, if $r_1(H) \neq r_2(H)$

for all $H' \in \mathcal{L}$. If $\{r_1, r_2\}$ is T_s-total, then $\hat{\omega}_{\{r_1,r_2\}}$ is an $\mathcal{S}$-3 interpretation.

Theorem 3. *Let $\Sigma \cup \{\varphi\}$ be a set of UCs and FDs over the SQL table definition T, and let T_s denote an NFS over T. Let $\mathcal{L}$ denote the set of propositional variables that corresponds to T, $\mathcal{S}$ the set of variables that corresponds to T_s, and $\Sigma' \cup \{\varphi'\}$ the set of goal and definite clauses over $\mathcal{L}$ that corresponds to $\Sigma \cup \{\varphi\}$. Then $\Sigma \models_{T_s} \varphi$ if and only if $\Sigma' \models^3_{\mathcal{S}} \varphi'$.* □

An example of the equivalence. Consider the table definition SCHEDULE with $\text{SCHEDULE}_s = \{\mathit{Location}\}$ and $\Sigma = \{\mathit{Speaker} \to \mathit{Location}, \mathit{Location} \to$

Time$\}$. Suppose we wonder if the FD $\varphi_1 = \mathit{Speaker} \rightarrow \mathit{Time}$ is implied by Σ in the presence of $\textsc{Schedule}_s$. According to Theorem 3 the problem $\Sigma \models_{\textsc{Schedule}_s} \varphi_1$ is equivalent to $\Sigma' \models^3_{\mathcal{S}} \varphi'_1$ where $\mathcal{S} = \{\mathit{Location}'\}$. Suppose an $\mathcal{S}$-3 interpretation $\hat{\omega}$ is not a model of φ'_1. Then $\hat{\omega}(\neg \mathit{Speaker}') = \mathbb{F} = \hat{\omega}(\mathit{Time}')$. For $\hat{\omega}$ to be an $\mathcal{S}$-3 model of Σ' we must thus have $\hat{\omega}(\mathit{Location}') = \mathbb{T} = \hat{\omega}(\neg \mathit{Location}')$, but $\mathit{Location}' \in \mathcal{S}$. We conclude that $\Sigma' \models^3_{\mathcal{S}} \varphi'_1$ and by Theorem 3 also $\Sigma \models_{\textsc{Schedule}_s} \varphi_1$. Let now be $\varphi_2 = \mathit{unique}(\mathit{Speaker})$. Then $\Sigma \not\models_{T_s} \varphi_1$ as the following SQL table t demonstrates:

Speaker	*Location*	*Time*
Grothendieck	Red Room	ni
Grothendieck	Red Room	ni

.

Indeed, the special $\mathcal{S}$-3 interpretation $\hat{\omega}_t$ where for all $L \in \mathcal{L}^\ell$, $\hat{\omega}_t(L) = \mathbb{F}$ iff $L \in \{\neg \mathit{Speaker}', \neg \mathit{Location}'\}$ is a model of Σ' but not a model of φ'_2.

6 The Boyce-Codd-Heath normal form for SQL

Boyce and Codd [8] and Heath [14] introduced a normal form condition on relation schemata that characterizes the absence of certain processing difficulties with any relation over the schema [22]. For SQL table definitions, no normal forms have been proposed to the best of our knowledge. We now propose an extension of the classical Boyce-Codd-Heath normal form to SQL table definitions.

Definition 1. *Let T denote an SQL table definition, T_s a null-free subdefinition, and Σ a set of UCs and FDs over T. Then T is said to be in* Boyce-Codd-Heath normal form *with respect to Σ and T_s if and only if for all non-trivial functional dependencies $X \rightarrow Y \in \Sigma^+_{\mathfrak{S}}$ we have $\mathit{unique}(X) \in \Sigma^+_{\mathfrak{S}}$.* □

Schema $\textsc{Schedule}$ of Example 2 is not in BCHNF with respect to Σ and T_s. However, if we replace the two FDs in Σ by the two UCs *unique*(*Location, Time*) and *unique*(*Speaker*, *Time*), then $\textsc{Schedule}$ is indeed in BCHNF with respect to Σ and T_s. It is very important to note here that the UCs are much stronger than the FDs. If the FD $X \rightarrow H$ is meaningful over T, then the table definition with projected column header set $T' = XH$ still carries the FD $X \rightarrow H$, but the UC $\mathit{unique}(X)$ may not be meaningful over T'. That is, decomposition and synthesis approaches [6, 16, 19] deserve new attention in the context of SQL. In general, duplicates should only be tolerated when they are meaningful, or updates are less expensive than duplicate elimination.

7 Semantic justification

We will now justify our syntactic definition of BCHNF semantically by showing that the condition is sufficient and necessary for the absence of redundant data value occurrences in any future tables. Following Vincent [22] we will make the notion of data redundancy explicit. Let T be an SQL table definition, H a column header of T, and r a row over T. A *replacement* of $r(H)$ is a row r' over T that satisfies the following conditions: i) for all $H' \in T - \{H\}$ we have $r'(H') = r(H)$, and ii) $r'(H) \neq r(H)$. Intuitively, a data value occurrence in some Σ-satisfying table is redundant if the occurrence cannot be replaced by any other data value without violating some constraint in Σ.

Definition 2. *Let T be an SQL table definition, $H \in T$ a column header, T_s an NFS and Σ a set of UCs and FDs over T, t a table over T that satisfies Σ and T_s, and r a row in t. We say that the data value occurrence $r(H)$ is* redundant *if and only if* every *replacement r' of $r(H)$ results in a table $t' := (t - \{r\}) \cup \{r'\}$ that violates Σ. We say that T is in* Redundancy-Free Normal Form *(RFNF) with respect to Σ and T_s if and only if there is no table t over T such that i) t satisfies Σ and T_s, and ii) t contains a row r such that for some column header H of T the data value occurrence $r(H)$ is redundant.* □

We show that the syntactic BCHNF condition of Definition 1 captures the semantic RFNF condition of Definition 2.

Theorem 4. *Let T be an SQL table definition, T_s an NFS and Σ a set of UCs and FDs over T. Then T is in RFNF with respect to Σ and T_s if and only if T is in BCHNF with respect to Σ and T_s.* □

Definition 1 refers to the syntactic closure $\Sigma^+_{\mathfrak{S}}$ of Σ and T_s under $\mathfrak{S}$, which can be exponential in the size of Σ. Therefore, the question remains if the problem whether an SQL table definition is in BCHNF with respect to Σ and T_s can be decided efficiently.

Theorem 5. *Let T be an SQL table definition, T_s an NFS and Σ a set of UCs and FDs over T. Then the following conditions are equivalent:*

1. *T is in BCHNF with respect to Σ and T_s,*
2. *for all non-trivial FDs $X \to Y \in \Sigma$ we have: unique$(X) \in \Sigma^+_{\mathfrak{S}}$,*
3. *for all non-trivial FDs $X \to Y \in \Sigma$ we have: $X \to T \in \Sigma^+_{\mathfrak{S}}$ and there is some unique$(Z) \in \Sigma$ such that $Z \subseteq XT_s$.* □

The following result follows directly from Theorem 5 and Theorem 2.

Theorem 6. *The problem whether an SQL table definition T is in Boyce-Codd-Heath Normal Form with respect to an NFS T_s and a set Σ of UCs and FDs over T can be decided in $\mathcal{O}(||\Sigma|| \times |\Sigma|)$ time.* □

If we define a primary key for an SQL table definition, i.e., there is some $X \subseteq T$ such that $X \subseteq T_s$ and *unique*$(X) \in \Sigma$, then the BCHNF condition for SQL table definitions reduces to the BCHNF condition for relation schemata: T is in BCHNF with respect to Σ and T_s if and only if for all non-trivial FDs $X \to Y \in \Sigma$ we have $X \to T \in \Sigma^+_{\mathfrak{S}}$. However, the presence of primary keys does not mean that the decomposition or synthesis approach [6, 16, 19] can eliminate any data redundancy.

Example 4. Consider the SQL table definition $T = \{Address, City, ZIP\}$ with $T_s = \{Address, ZIP\}$ and

$$\Sigma = \{unique(Address, City), unique(Address,\ ZIP), ZIP \to\ City\}.$$

Hence, we have a primary key $\{Address,\ ZIP\}$. A synthesis into the following table definitions:

- $\{City,\ ZIP\}$ with $ZIP \to City$ and NFS $\{ZIP\}$,
- $\{Address, City\}$ with $unique(Address, City)$ and NFS $\{Address\}$, and
- $\{Address,\ ZIP\}$ with $unique(Address,\ ZIP)$ and NFS $\{Address,\ ZIP\}$

is dependency-preserving, but neither lossless nor is the first resulting table definition in BCHNF. The tables

Address	*City*	*ZIP*
03 Hudson St	ni	10001
70 King St	ni	10001

City	*ZIP*
ni	10001
ni	10001

Address	*ZIP*
03 Hudson St	10001
70 King St	10001

Address	*City*
03 Hudson St	ni
70 King St	ni

show the synthesis on the semantic level. For this example, it appears to be sensible to replace the FD $ZIP \to City$ on $\{City,\ ZIP\}$ by the UC $unique(ZIP)$. The resulting synthesis would then be lossless, all schemata would be in BCHNF and the second table from the left would only contain one row. □

The example illustrates that the approaches of synthesis and decomposition to database normalization require new attention when we consider the features of SQL that allow duplicate and partial information. The presence of duplicates requires uniqueness constraints in addition to functional dependencies, but uniqueness constraints are not preserved when performing joins. Hence, it is not clear what *dependency-preservation* means. The presence of null values requires join attributes to be `NOT NULL` when *lossless* decompositions are to be achieved. Furthermore, projection becomes more difficult to define when duplicates are to be eliminated only sometimes.

8 Conclusion

The class of uniqueness constraints is not subsumed by the class of functional dependencies over SQL tables, in contrast to relations. For this purpose, we have characterized the implication problem for the combined class of UCs and FDs in the presence of `NOT NULL` constraints axiomatically, algorithmically and logically. We have further proposed a syntactic Boyce-Codd-Heath normal form condition for SQL table definitions, and justified this condition semantically. That is, the condition characterizes the absence of redundant data value occurrences in all possible SQL tables. On one hand, the semantics of SQL really calls for a comprehensive support to specify and maintain FDs to guarantee consistency and locate data redundancy. On the other hand, the SQL features motivate a thorough study of the decomposition and synthesis approaches towards achieving normalization.

References

1. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases.* Addison-Wesley, 1995.
2. M. Arenas and L. Libkin. A normal form for XML documents. *ACM Trans. Database Syst.*, 29(1):195–232, 2004.
3. M. Arenas and L. Libkin. An information-theoretic approach to normal forms for relational and XML data. *J. ACM*, 52(2):246–283, 2005.
4. W. W. Armstrong. Dependency structures of database relationships. *Information Processing*, 74:580–583, 1974.
5. P. Atzeni and N. Morfuni. Functional dependencies and constraints on null values in database relations. *Information and Control*, 70(1):1–31, 1986.
6. J. Biskup, U. Dayal, and P. Bernstein. Synthesizing independent database schemas. In *SIGMOD Conference*, pages 143–151, 1979.
7. E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, 1970.
8. E. F. Codd. Recent investigations in relational data base systems. In *IFIP Congress*, pages 1017–1021, 1974.
9. C. Date and H. Darwen. *A guide to the SQL standard.* Addison-Wesley Professional, Reading, MA, USA, 1997.
10. J. Diederich and J. Milton. New methods and fast algorithms for database normalization. *ACM Trans. Database Syst.*, 13(3):339–365, 1988.
11. S. Hartmann and S. Link. Efficient reasoning about a robust XML key fragment. *ACM Trans. Database Syst.*, 34(2), 2009.
12. S. Hartmann and S. Link. Numerical constraints on XML data. *Inf. Comput.*, 208(5):521–544, 2010.
13. S. Hartmann and S. Link. When data dependencies over SQL tables meet the Logics of Paradox and S-3. In *PODS Conference*, 2010.
14. I. J. Heath. Unacceptable file operations in a relational data base. In *SIGFIDET Workshop*, pages 19–33, 1971.
15. T. Imielinski and W. Lipski Jr. Incomplete information in relational databases. *J. ACM*, 31(4):761–791, 1984.
16. H. Köhler. Finding faithful Boyce-Codd normal form decompositions. In *AAIM Conference*, volume 4041 of *Lecture Notes in Computer Science*, pages 102–113. Springer, 2006.
17. H. Köhler and S. Link. Armstrong axioms and Boyce-Codd-Heath normal form under bag semantics. *Inf. Process. Lett.*, 110(16):717–724, 2010.
18. E. Lien. On the equivalence of database models. *J. ACM*, 29(2):333–362, 1982.
19. J. A. Makowsky and E. V. Ravve. Dependency preserving refinements and the fundamental problem of database design. *Data Knowl. Eng.*, 24(3):277–312, 1998.
20. Y. Sagiv, C. Delobel, D. S. Parker Jr., and R. Fagin. An equivalence between relational database dependencies and a fragment of propositional logic. *J. ACM*, 28(3):435–453, 1981.
21. M. Schaerf and M. Cadoli. Tractable reasoning via approximation. *Artif. Intell.*, 74:249–310, 1995.
22. M. Vincent. Semantic foundation of 4NF in relational database design. *Acta Inf.*, 36:1–41, 1999.
23. M. Vincent, J. Liu, and C. Liu. Strong functional dependencies and their application to normal forms in XML. *ACM Trans. Database Syst.*, 29(3):445–462, 2004.
24. C. Zaniolo. Database relations with null values. *J. Comput. Syst. Sci.*, 28(1):142–166, 1984.

9 Appendix - Proofs

9.1 Axiomatic and algorithmic characterization

Lemma 2. *The inference rules in $\mathfrak{S}$ are sound for the implication of uniqueness constraints and functional dependencies in the presence of a null-free subdefinition.*

Proof. The soundness of the rules in $\mathfrak{M}$ can be proven for SQL tables in the same way it was shown for sets of tuples [5]. It remains to show the soundness of the demotion and null pullback rules.

For the soundness of the demotion rule let T be an SQL table definition and $X \to Y$ an FD over T. Suppose that there is a table t that violates $X \to Y$. That is, there are two rows $r, r' \in t$ such that r, r' are X-total and $r[X] = r'[X]$ and there is some $H \in Y$ such that $r(H) \neq r'(H)$. In particular, r, r' are distinct rows of the table t. We conclude that t also violates the uniqueness constraint $\mathit{unique}(X)$.

For the soundness of the null pullback rule let T be an SQL table definition, $\mathit{unique}(X)$, $\mathit{unique}(Y)$ uniqueness constraints over T, $X \to Y$ an FD over T and T_s an NFS over T such that $Y \subseteq XT_s$. Suppose that there is a table t that violates $\mathit{unique}(X)$. That is, there are two distinct rows $r, r' \in t$ such that r, r' are X-total and $r[X] = r'[X]$. If t satisfies the FD $X \to Y$, then $r[Y] = r'[Y]$. However, $Y \subseteq XT_s$ implies that r, r' are also both Y-total. That is, t violates the uniqueness constraint $\mathit{unique}(Y)$. □

Lemma 3. *Let T be an SQL table definition, T_s an NFS, Σ_{UC} a set of uniqueness constraints and Σ_{FD} a set of FDs over T. If $\Sigma_{UC} \cup \Sigma_{FD} \models_{T_s} X \to Y$, then $\Sigma_{UC}^{FD} \cup \Sigma_{FD} \models_{T_s} X \to Y$.*

Proof. Let t be a T_s-total SQL table over T such that $\models_t \Sigma_{\text{UC}}^{\text{FD}} \cup \Sigma_{\text{FD}}$ and $\not\models_t X \to Y$. Then there are two rows $r, r' \in t$ such that $r[X] = r'[X]$ and r, r' are both X-total and there is some $H \in Y$ such that $r(H) \neq r'(H)$. Suppose there is some $\mathit{unique}(Z) \in \Sigma_{UC}$ such that $\not\models_{\{r,r'\}} \mathit{unique}(Z)$. Then r, r' are Z-total and $r[Z] = r'[Z]$. However, $\models_{\{r,r'\}} Z \to T$ and thus $r[T] = r'[T]$, a contradiction. Consequently, $\Sigma_{\text{UC}} \cup \Sigma_{\text{FD}} \not\models_{T_s} \mathit{unique}(X)$. □

Lemma 4. *Let T be an SQL table definition, T_s an NFS, Σ_{UC} a set of uniqueness constraints, and Σ_{FD} a set of FDs over T. Then $\Sigma_{UC} \cup \Sigma_{FD} \models_{T_s} \mathit{unique}(X)$ if and only if $\Sigma_{UC}^{FD} \cup \Sigma_{FD} \models_{T_s} X \to T$ and there is some $\mathit{unique}(Z) \in \Sigma_{UC}$ such that $Z \subseteq XT_s$.*

Proof. **Sufficiency.** Let $\Sigma = \Sigma_{\text{UC}} \cup \Sigma_{\text{FD}}$. From $X \to T \in \left(\Sigma_{\text{UC}}^{\text{FD}} \cup \Sigma_{\text{FD}}\right)^*_{T_s}$ we conclude $X \to T \in \Sigma^*_{T_s}$ since $\left(\Sigma_{\text{UC}}^{\text{FD}} \cup \Sigma_{\text{FD}}\right)^*_{T_s} \subseteq \Sigma^*_{T_s}$. From $X \to T \in \Sigma^*_{T_s}$ we conclude $X \to Z \in \Sigma^*_{T_s}$ by soundness of the decomposition rule. From $X \to Z \in \Sigma^*_{T_s}$ and $\mathit{unique}(Z) \in \Sigma_{\text{UC}}$ such that $Z \subseteq XT_s$ we conclude $\mathit{unique}(Z) \in \Sigma^*_{T_s}$ by soundness of the null pullback rule.

Necessity. From $\Sigma_{\text{UC}} \cup \Sigma_{\text{FD}} \models_{T_s} \mathit{unique}(X)$ we conclude $\Sigma_{\text{UC}} \cup \Sigma_{\text{FD}} \models_{T_s} X \to T$ by soundness of the demotion rule. From $\Sigma_{\text{UC}} \cup \Sigma_{\text{FD}} \models_{T_s} X \to T$ we conclude $\Sigma_{\mathrm{UC}}^{\mathrm{FD}} \cup \Sigma_{\mathrm{FD}} \models_{T_s} X \rightarrow T$ by Lemma 3. It remains to show that there is some *unique*$(Z) \in \Sigma_{\mathrm{UC}}$ such that $Z \not\subseteq XT_s$. Assume to the contrary that for all *unique*$(Z) \in \Sigma_{\mathrm{UC}}$ we have $Z \not\subseteq XT_s$. Under this assumption we will derive the contradiction that $\Sigma_{\mathrm{UC}} \cup \Sigma_{\mathrm{FD}} \not\models_{T_s}$ *unique*(X) by constructing a T_s-total two-row table t that satisfies $\Sigma_{\mathrm{UC}} \cup \Sigma_{\mathrm{FD}}$ and violates *unique*(X).

For $\Sigma = \Sigma_{\mathrm{UC}} \cup \Sigma_{\mathrm{FD}}$ and $X^*_{\Sigma,T_s} = \{H \in T \mid \Sigma \models_{T_s} X \rightarrow H\}$ let $t = \{r, r'\}$ such that

- $\mathtt{ni} \neq r(H) = r'(H) \neq \mathtt{ni}$ for all $H \in X(X^*_{\Sigma,T_s} \cap T_s)$,
- $r(H) = \mathtt{ni} = r'(H)$ for all $H \in (X^*_{\Sigma,T_s} - XT_s) \cup (T - X^*_{\Sigma,T_s}T_s)$,
- $\mathtt{ni} \neq r(H) \neq r'(H) \neq \mathtt{ni}$ for all $H \in T_s - X^*_{\Sigma,T_s}$.

For example, the table t may look as follows:

	$X(X^*_{\Sigma,T_s} \cap T_s)$	$(X^*_{\Sigma,T_s} - XT_s) \cup (T - X^*_{\Sigma,T_s}T_s)$	$T_s - X^*_{\Sigma,T_s}$
r	$0 \cdots 0$	$\mathtt{ni} \cdots \mathtt{ni}$	$0 \cdots 0$
r'	$0 \cdots 0$	$\mathtt{ni} \cdots \mathtt{ni}$	$1 \cdots 1$

.

Since r, r' are both X-total and $r[X] = r'[X]$ we conclude that $\not\models_t$ *unique*(X). We also conclude that t is T_s-total. We show now that $\models_t \Sigma$.

Let *unique*$(Z) \in \Sigma$. According to our assumption there is some $H \in Z \cap (T - XT_s)$. Consequently, $r(H) = \mathtt{ni} = r'(H)$ and, thus, $\models_t$ *unique*(Z).

Let $U \rightarrow V \in \Sigma$. Suppose that $r[U] = r'[U]$ and r, r' are U-total. Then $U \subseteq X(X^*_{\Sigma,T_s} \cap T_s)$. From $\Sigma \models_{T_s} X \rightarrow X^*_{\Sigma,T_s}$ and $U \subseteq X^*_{\Sigma,T_s}$ we conclude that $\Sigma \models_{T_s} X \rightarrow U$ by soundness of the decomposition rule. From $\Sigma \models_{T_s} X \rightarrow U$, $\Sigma \models_{T_s} U \rightarrow V$ and $U \subseteq XT_s$ we conclude $\Sigma \models_{T_s} X \rightarrow V$ by soundness of the null transitivity rule. Hence, $V \subseteq X^*_{\Sigma,T_s}$ and $r[V] = r'[V]$.

We have just derived the contradiction that $\Sigma_{\mathrm{UC}} \cup \Sigma_{\mathrm{FD}} \not\models_{T_s}$ *unique*(X). Hence, our assumption must have been wrong. Consequently, there is some *unique*$(Z) \in \Sigma_{\mathrm{UC}}$ such that $Z \subseteq XT_s$. □

Theorem 7 (Theorem 1 restated). *The set $\mathfrak{S}$ is a finite axiomatization for the implication of uniqueness constraints and functional dependencies in the presence of null-free subdefinitions.*

Proof. The soundness of $\mathfrak{S}$ was shown in Lemma 2. We establish the completeness of $\mathfrak{S}$ by showing for an arbitrary table definition T, an arbitrary NFS T_s and an arbitrary set $\Sigma_{\mathrm{UC}} \cup \Sigma_{\mathrm{FD}} \cup \{\varphi\}$ of uniqueness constraints and functional dependencies over T the following holds: if $\Sigma_{\mathrm{UC}} \cup \Sigma_{\mathrm{FD}} \models_{T_s} \varphi$, then $\Sigma_{\mathrm{UC}} \cup \Sigma_{\mathrm{FD}} \vdash_{\mathfrak{S}} \varphi$. We consider two cases. In case (1) φ denotes the FD $X \rightarrow Y$. Then we know by Lemma 3 that $\Sigma_{\mathrm{UC}}^{\mathrm{FD}} \cup \Sigma_{\mathrm{FD}} \models_{T_s} \varphi$ holds. From the completeness of $\mathfrak{M}$ for the implication of functional dependencies in the presence of an NFS we conclude that $\Sigma_{\mathrm{UC}}^{\mathrm{FD}} \cup \Sigma_{\mathrm{FD}} \vdash_{\mathfrak{M}} \varphi$. Since $\mathfrak{M} \subseteq \mathfrak{S}$ holds we know that $\Sigma_{\mathrm{UC}}^{\mathrm{FD}} \cup \Sigma_{\mathrm{FD}} \vdash_{\mathfrak{S}} \varphi$ holds, too. The *demotion rule* shows for all $\sigma \in \Sigma_{\mathrm{UC}}^{\mathrm{FD}}$ that $\Sigma_{\mathrm{UC}} \vdash_{\mathfrak{S}} \sigma$ holds. Consequently, we have $\Sigma_{\mathrm{UC}} \cup \Sigma_{\mathrm{FD}} \vdash_{\mathfrak{S}} \varphi$. This concludes case (1). In case (2) φ denotes the UC *unique*(X). From $\Sigma_{\mathrm{UC}} \cup \Sigma_{\mathrm{FD}} \models_{T_s}$ *unique*(X) we conclude by Lemma 4 that there is some *unique*$(Z) \in \Sigma_{\mathrm{UC}}$ such that $Z \subseteq XT_s$ holds. We also conclude from $\Sigma_{\mathrm{UC}} \cup \Sigma_{\mathrm{FD}} \models_{T_s}$ *unique*(X) that $\Sigma_{\mathrm{UC}} \cup \Sigma_{\mathrm{FD}} \models_{T_s} X \rightarrow Z$ holds by soundness of the demotion rule. From case (1) it follows that $\Sigma_{\text{UC}} \cup \Sigma_{\text{FD}} \vdash_{\mathfrak{S}} X \to Z$ holds. A final application of the *null pullback rule* shows that $\Sigma_{\text{UC}} \cup \Sigma_{\text{FD}} \vdash_{\mathfrak{S}} \varphi$ holds. □

Algorithm 8 (NFSClosure(X,Σ[FD],T_s,T))

Input: header set X, FD set Σ[FD], NFS T_s over SQL table definition T
Output: header set closure $X^*_{\Sigma[\text{FD}],T_s}$ of X with respect to Σ[FD] and T_s
Method:

```
(A0) CLOSURE := $X$;
(A1) repeat
       OLDCLOSURE := CLOSURE;
       for all $V \to W \in \Sigma$[FD] do
         if $V \subseteq$ CLOSURE $\cap\, XT_s$ then
           CLOSURE := CLOSURE $\cup\, W$;
         endif;
       enddo;
     until OLDCLOSURE = CLOSURE;
(A2) return CLOSURE;
```

□

9.2 Logical characterization

Lemma 5. *Let T be some SQL table, t a two-row table over T and T_s an NFS over T. Let $\mathcal{L}$ be the set of propositional variables that corresponds to T, and $\mathcal{S}$ the set of propositional variables that corresponds to T_s. If t satisfies T_s, then $\hat{\omega}_t$ is an $\mathcal{S}$-3 interpretation of $\mathcal{L}$.*

Proof. If t satisfies T_s, then the two rows of t are T_s-total. According to the definition of the special-3-interpretation $\hat{\omega}_t$ it cannot be the case that $\hat{\omega}_t(H') = \mathbb{T}$ and $\hat{\omega}_t(\neg H') = \mathbb{T}$ for any $H' \in \mathcal{S}$. □

The converse of Lemma 5 is not valid. In fact, let $T = AB$ and $T_s = A$, and let $t = \{(a, b), (\texttt{ni}, b')\}$. We have $\hat{\omega}_t(A') = \mathbb{F} = \hat{\omega}_t(B')$ and $\hat{\omega}_t(\neg A') = \mathbb{T} = \hat{\omega}_t(\neg B')$, but t violates $T_s = A$. However, note that we can replace the null value occurrence ni in t by a non-null value different from a. The resulting table t' satisfies $T_s = A$ and $\hat{\omega}_{t'} = \hat{\omega}_t$. This strategy is always applicable.

Lemma 6. *Let T be some SQL table definition, and T_s an NFS over T. Let $\mathcal{L}$ be the set of propositional variables that corresponds to T, and $\mathcal{S}$ the set of propositional variables that corresponds to T_s. Let $\hat{\omega}$ be an $\mathcal{S}$-3 interpretation of $\mathcal{L}$. Then there is a two-row table t over T such that t satisfies T_s and $\hat{\omega}_t = \hat{\omega}$.*

Proof. Define a row r over T as follows: for $H \in T$ let $r_1(H) := a \in dom(H) - \{\texttt{ni}\}$, if $\hat{\omega}(H') = \mathbb{F}$ or $\hat{\omega}(\neg H') = \mathbb{F}$, and $r_1(H) := \texttt{ni}$ otherwise. Define another row r_2 over T as follows: i) let $r_2(H) := r_1(H)$, if $\hat{\omega}(H') = \mathbb{T}$, and ii) let $r_2(H) := a' \in dom(H) - \{\texttt{ni}, r_1(H)\}$ otherwise. Let $t := \{r_1, r_2\}$. From this definition it follows that t is T_s-total. Moreover, for all $H' \in \mathcal{L}$ we have $\hat{\omega}_t(H') = \hat{\omega}(H')$. □

The following lemma justifies the definitions of the corresponding fragment of $\mathcal{L}$-formulae and the special-3-interpretation of $\mathcal{L}$.

Lemma 7. *Let t be a two-row table over the SQL table definition T, and let φ be a UC or an FD over T. Then t satisfies φ if and only if $\hat{\omega}_t$ is a model of φ'.*

Proof. Let $t = \{r_1, r_2\}$. If $\varphi = \mathit{unique}(\emptyset)$, then a table satisfies φ if and only if the table contains less than two rows. Note that, in particular, t violates φ and $\varphi' = \mathit{false}$.

Sufficiency. Assume that $\hat{\omega}_t$ is a model of φ'. We show that t satisfies φ.

First, let φ denote the UC $\mathit{unique}(X)$ where $X = \{H_1, \ldots, H_n\}$. If t violated φ, then $r_1[X] = r_2[X]$ and r_1, r_2 are X-total. Hence, $\hat{\omega}_t(\neg H_i') = \mathbb{F}$ for all $i = 1, \ldots, n$. This, however, is a contradiction since $\hat{\omega}_t$ is a model of φ'. Consequently, t satisfies φ. Now, let φ denote the FD $X \to H$ where $X = \{H_1, \ldots, H_n\}$. If $r_1[X] = r_2[X]$ and r_1, r_2 are X-total, then $\hat{\omega}_t(\neg H_i') = \mathbb{F}$ for all $i = 1, \ldots, n$. Since $\hat{\omega}_t$ is a model of φ' it follows that $\hat{\omega}_t(H) = \mathbb{T}$. Consequently, $r_1(H) = r_2(H)$. Thus, t satisfies φ.

Necessity. Assume that t satisfies φ. We show that $\hat{\omega}_t$ is a model of φ'.

First, let φ' denote the formula $\neg H_1' \vee \cdots \vee \neg H_n'$. Suppose that $\hat{\omega}_t(\neg H_i') = \mathbb{F}$ for all $i = 1, \ldots, n$. It would follow that $r_1[X] = r_2[X]$ and r_1, r_2 are X-total. This would contradict that t satisfies φ. Consequently, $\hat{\omega}_t(\neg H_i') = \mathbb{T}$ for some $i \in \{1, \ldots, n\}$. Therefore, $\hat{\omega}_t$ is a model of φ'. Now, let φ' denote the formula $\neg H_1' \vee \cdots \vee \neg H_n' \vee H$. Suppose that $\hat{\omega}_t(\neg H_i') = \mathbb{F}$ for all $i = 1, \ldots, n$ (otherwise $\hat{\omega}_t$ is a model of φ'). It follows that $r_1[X] = r_2[X]$ and r_1, r_2 are X-total. Since r satisfies φ it follows that $r_1(H) = r_2(H)$. Consequently, $\hat{\omega}_t(H') = \mathbb{T}$. Therefore, $\hat{\omega}_t$ is a model of φ'. □

In fact, Lemmata 5, 6 and 7 allow us to establish the anticipated equivalence between the implication of UCs and FDs in the presence of an NFS T_s and the $\mathcal{S}$-3 implication of their corresponding goal and definite clauses in $\mathcal{L}$.

Theorem 9 (Theorem 3 restated). *Let $\Sigma \cup \{\varphi\}$ be a set of UCs and FDs over the SQL table definition T, and let T_s denote an NFS over T. Let $\mathcal{L}$ denote the set of propositional variables that corresponds to T, $\mathcal{S}$ the set of variables that corresponds to T_s, and $\Sigma' \cup \{\varphi'\}$ the set of goal and definite clauses over $\mathcal{L}$ that corresponds to $\Sigma \cup \{\varphi\}$. Then $\Sigma \models_{T_s} \varphi$ if and only if $\Sigma' \models^3_{\mathcal{S}} \varphi'$.*

Proof. It follows immediately from the definition of satisfaction for UCs and FDs in the presence of an NFS that $\Sigma \models_{T_s} \varphi$ if and only if $\Sigma \models_{2-T_s} \varphi$ (the latter problem is to decide if every *two-row* table over T that satisfies Σ and T_s also satisfies φ). Hence, it suffices to show that $\Sigma \models_{2-T_s} \varphi$ if and only if $\Sigma' \models^3_{\mathcal{S}} \varphi'$.

We show first that if $\Sigma' \models^3_{\mathcal{S}} \varphi'$ holds, then $\Sigma \models_{2-T_s} \varphi$ holds, too. For this purpose, suppose that $\Sigma \models_{2-T_s} \varphi$ does not hold. Consequently, there is some two-row table t over T that satisfies Σ and T_s but violates φ. Following Lemma 5, $\hat{\omega}_t$ is an $\mathcal{S}$-3 interpretation. According to Lemma 7, $\hat{\omega}_t$ is an $\mathcal{S}$-3 model of Σ' but not an $\mathcal{S}$-3 model of φ'. Consequently, $\Sigma' \models^3_{\mathcal{S}} \varphi'$ does also not hold.

It now remains to show that if $\Sigma \models_{2-T_s} \varphi$ holds, then $\Sigma' \models^3_{\mathcal{S}} \varphi'$ holds, too. For this purpose, suppose that $\Sigma' \models^3_{\mathcal{S}} \varphi'$ does not hold. Consequently, there is some $\mathcal{S}$-3 interpretation $\hat{\omega}$ of $\mathcal{L}$ that is a model of Σ' but not a model of φ'. According to Lemma 6 there is some two-row table t that satisfies T_s and $\hat{\omega}_t = \hat{\omega}$.

Hence, Lemma 7 guarantees that t satisfies Σ but violates φ. We conclude that $\Sigma \models_{2-T_s} \varphi$ does also not hold. $\square$

9.3 Semantic justification of BCHNF

Theorem 10 (Theorem 4 restated). *Let T be an SQL table definition, T_s an NFS and Σ a set of UCs and FDs over T. Then T is in RFNF with respect to Σ and T_s if and only if T is in BCHNF with respect to Σ and T_s.*

Proof. Let T not be in RFNF with respect to Σ and T_s. Then there is some T_s-total table t over T that satisfies Σ, some row $r \in t$ and some header $H \in T$ such that $r(H)$ is redundant. We need to show that there is some non-trivial FD $X \to Y \in \Sigma^+_{\mathfrak{G}}$ such that *unique*$(X) \notin \Sigma^+_{\mathfrak{G}}$. Let $t[H] := \{\bar{r}(H)|\bar{r} \in t\}$. Define a replacement r' of $r(H)$ such that $r'(H) \in dom(H) - (t[H] \cup \{\mathtt{ni}\})$. Furthermore, let $t' := (t - \{r\}) \cup \{r'\}$. Since $r(H)$ is redundant it follows that t' violates Σ. Since $\models_t \Sigma$ and t' agrees with t except on $r'(H) \notin t[H] \cup \{\mathtt{ni}\}$ it follows that t' cannot violate any UC in Σ. Let t' violate the FD $X \to Y \in \Sigma$. From the definition of r' and the properties of t and t' it follows that $H \in Y - X$. Hence, $X \to Y$ is non-trivial. Since t' violates $X \to Y \in \Sigma$ there is some $r'' \in t' - \{r'\}$ such that $r''[X] = r'[X]$, and r'', r' are X-total. Moreover, $r'' \in t$, $r''[X] = r[X]$ and r'', r are X-total since $H \notin X$. Therefore, t satisfies Σ but t violates the UC *unique*(X). Hence, *unique*$(X) \notin \Sigma^*_{T_s}$ and by the soundness of $\mathfrak{G}$ we conclude *unqiue*$(X) \notin \Sigma^+_{\mathfrak{G}}$. It follows that T is not in BCHNF with respect to Σ and T_s.

Vice versa, let T not be in BCHNF with respect to Σ and T_s. Then there is some non-trivial FD $X \to Y \in \Sigma^+_{\mathfrak{G}}$ such that *unique*$(X) \notin \Sigma^+_{\mathfrak{G}}$. We need to show that there is some table t over T that satisfies Σ and T_s, some row $r \in t$ and some header $H \in T$ such that $r(H)$ is redundant. Let $t := \{r, r'\}$ consist of two rows r and r' over T such that for all $H' \in T$, i) $\mathtt{ni} \neq r(H') = r'(H') \neq \mathtt{ni}$ holds if and only if $H' \in X(X^+_{\Sigma,T_s} \cap T_s)$, ii) $r(H') = \mathtt{ni} = r'(H')$ if and only if $H' \in X^+_{\Sigma,T_s} - XT_s$, and iii) $\mathtt{ni} \neq r(H') \neq r'(H') \neq \mathtt{ni}$ if and only if $H' \in T - X^+_{\Sigma,T_s}$. Here,

$$X^+_{\Sigma,T_s} := \{H' \in T | X \to H' \in \Sigma^+_{\mathfrak{G}}\}.$$

It follows immediately that t is T_s-total. We show that t satisfies Σ.

Let $U \to V \in \Sigma$ and let $r[U] = r'[U]$ such that r, r' are U-total. It follows that $U \subseteq X(X^+_{\Sigma,T_s} \cap T_s)$. From $X \to X^+_{\Sigma,T_s} \in \Sigma^+_{\mathfrak{G}}$ and $U \subseteq X^+_{\Sigma,T_s}$ follows $X \to U \in \Sigma^+_{\mathfrak{G}}$ by the *decomposition rule*. From $X \to U \in \Sigma^+_{\mathfrak{G}}$, $U \to V \in \Sigma$ and $U \subseteq XT_s$ we conclude $X \to V \in \Sigma^+_{\mathfrak{G}}$ by means of the *null transitivity rule*. Consequently, $V \subseteq X^+_{\Sigma,T_s}$ and therefore $r[V] = r'[V]$. We conclude that t satisfies $U \to V$.

Let *unique*$(U) \in \Sigma$, and assume that $r[U] = r'[U]$ holds for the distinct U-total rows r and r'. We conclude that $U \subseteq X(X^+_{\Sigma,T_s} \cap T_s)$ holds. From $X \to X^+_{\Sigma} \in \Sigma^+_{\mathfrak{G}}$ we infer $X \to U \in \Sigma^+_{\mathfrak{G}}$ by means of the *decomposition rule*. From *unique*$(U) \in \Sigma$, $X \to U \in \Sigma^+_{\mathfrak{G}}$ and $U \subseteq XT_s$ we infer that *unique*$(X) \in \Sigma^+_{\mathfrak{G}}$ by an application of the *null pullback rule*. This, however, is a contradiction since *unique*$(X) \notin \Sigma^+_{\mathfrak{G}}$. Consequently, t satisfies *unique*(U). Hence, t satisfies Σ.

Now let $H \in Y - X$. Since $Y \subseteq X^+_{\Sigma,T_s}$ it follows that $r(H)$ is redundant. Therefore, T is not in RFNF with respect to Σ and T_s. □

Theorem 11 (Theorem 5 restated). *Let T be an SQL table definition, T_s an NFS and Σ a set of UCs and FDs over T. Then the following conditions are equivalent:*

1. *T is in BCHNF with respect to Σ and T_s,*
2. *for all non-trivial FDs $X \to Y \in \Sigma$ we have: $unique(X) \in \Sigma^+_{\mathfrak{S}}$,*
3. *for all non-trivial FDs $X \to Y \in \Sigma$ we have: $X \to T \in \Sigma^+_{\mathfrak{S}}$ and there is some $unique(Z) \in \Sigma$ such that $Z \subseteq XT_s$.*

Proof. We show first the equivalence between *1.* and *2.* Condition *1.* implies condition *2.* since $\Sigma \subseteq \Sigma^+_{\mathfrak{S}}$. We show next that condition *2.* implies condition *1.* Assume that T is not in BCHNF with respect to Σ and T_s. That is, there is some non-trivial FD $X \to Y \in \Sigma^+_{\mathfrak{S}}$ such that $unique(X) \notin \Sigma^+_{\mathfrak{S}}$. We need to show that there is some non-trivial FD $X' \to Y' \in \Sigma$ such that $unique(X') \notin \Sigma^+_{\mathfrak{S}}$. Let

$$\Sigma = \Sigma_0 \subset \Sigma_1 \subset \cdots \subset \Sigma_k = \Sigma^+_{\mathfrak{S}}$$

be a proper chain where for all $j = 1, \ldots, k$ the set Σ_j results from Σ_{j-1} by a single application of an inference rule in $\mathfrak{S}$. We show that if there is some non-trivial FD $X \to Y \in \Sigma_j$ such that $unique(X) \notin \Sigma^+_{\mathfrak{S}}$, then there is some non-trivial FD $X' \to Y' \in \Sigma_{j-1}$ such that $unique(X') \notin \Sigma^+_{\mathfrak{S}}$. For $j > 0$ let $X \to Y \in \Sigma_j - \Sigma_{j-1}$ be non-trivial such that $unique(X) \notin \Sigma^+_{\mathfrak{S}}$. Then $X \to Y$ has been inferred either by means of the decomposition, union or null transitivity rule. In case of the decomposition rule we have $X \to YZ \in \Sigma_{j-1}$ with $unique(X) \notin \Sigma^+_{\mathfrak{S}}$. In case of the union rule we have $X \to U \in \Sigma_{j-1}$ and $X \to W \in \Sigma_{j-1}$ with $Y = UW$ and $unique(X) \notin \Sigma^+_{\mathfrak{S}}$. In case of the null transitivity rule we know that there are $X \to Z$ and $Z \to Y$ in Σ_{j-1} with $Z \subseteq XT_s$. If $X \to Z$ is non-trivial, then we are done. If $X \to Z$ is trivial, then $Z \to Y$ is non-trivial since otherwise $X \to Y$ would be trivial, too. If $unique(Z) \in \Sigma^+_{\mathfrak{S}}$, then an application of the null pullback rule to $unique(Z) \in \Sigma^+_{\mathfrak{S}}$, $X \to Z \in \Sigma^+_{\mathfrak{S}}$ and $Z \subseteq XT_s$ shows that $unique(X) \in \Sigma^+_{\mathfrak{S}}$ holds as well. This is a contradiction, i.e., $unique(Z) \notin \Sigma^+_{\mathfrak{S}}$. We have just shown that there is some non-trivial FD $X' \to Y' \in \Sigma$ such that $unique(X') \notin \Sigma^+_{\mathfrak{S}}$.

The equivalence between conditions *2.* and *3* follows immediately from Lemma 4. □