Journal of Universal Computer Science, vol. 4, no. 12 (1998), 883-897
submitted: 16/10/98, accepted: 1/12/98, appeared: 28/12/98

Linear Derivations for Keys of a Database Relation Schema

Ralf Wastl
(University of Würzburg, Germany
wastl@informatik.uni-wuerzburg.de)

Abstract: In [Wastl 1998] we have introduced the Hilbert style inference system $\mathbb{K}$ for deriving all keys of a database relation schema. In this paper we investigate formal $\mathbb{K}$-derivations more closely using the concept of tableaux. The analysis gives insight into the process of deriving keys of a relation schema. Also, the concept of tableaux gives a proof procedure for computing all keys of a relation schema. In practice, the methods developed here will be usefull for computing keys or for deciding whether an attribute is a key attribute, respectively non-key attribute. This decision problem becomes relevant when checking whether a relation schema is in third normal form, or when applying the well–known 3NF-decomposition algorithm (a.k.a. 3NF-synthesis algorithm).

Key Words: Database relation schema, Keys, Automated deduction.

Category: H.2.1, H.2.8

1 Introduction

Semantic tableaux are a well-known refutation proof procedure in automated theorem proving. We will use tableaux for deriving keys of a relation schema using the inference system $\mathbb{K}$. The methods developed here are constructive in the sense that branches in a tableau will correspond to formal $\mathbb{K}$-derivations. It will be shown that these derivations are linear. The paper is organized as follows: In this section we will give an example to show that minimal keys are needed for relational database design, and we will give the basic items concerning relation schemas (cf. [Maier 1983], [Ullman 1988]). In section 2 we will shortly discuss the inference system $\mathbb{K}$ (cf. [Wastl 1998]). In section 3 we will informally introduce the concept of $\mathbb{K}$-tableaux for deriving keys. A formal treatment is then given in section 4. In the last section we will consider linear derivations for keys.

Example 1 ([Elmasri et al. 1994])
A manufacturer wants to describe lots in his database. Therefore, he designs a relation schema $R = \langle U, F\rangle$ with

$$U = \{\mathit{PropertyId}, \mathit{CountryName}, \mathit{LotNumber}, \mathit{Area}, \mathit{Price}\}$$

and

$$\begin{array}{rcl} F = \{ & & \\ \mathit{PropertyId} & \to & \mathit{CountryName}, \mathit{LotNumber}, \mathit{Area}, \mathit{Price}, \\ \mathit{CountryName}, \mathit{LotNumber} & \to & \mathit{PropertyId}, \mathit{Area}, \mathit{Price}, \\ \mathit{Area} & \to & \mathit{Price}\,\}. \end{array}$$

Intuitively, the *PropertyId* identifies a lot; this is coded in the first functional dependency. The *CountryName* and *LotNumber* determine the *PropertyId*, *Area* and *Price*. Finally, given the *Area*, the manufacturer can tell the *Price*.

To avoid update, insert or delete anomalies (to a certain degree), the manufacturer wants to test whether his relation schema is in third normal form (3NF). This requires to check whether for all functional dependencies in F the left hand side is a super key or the right hand side is a prime attribute (that is, occurs in a minimal key). For the first functional dependency we observe that the transitive closure $\mathit{PropertyId}^+$ equals U; thus *PropertyId* is a super key (and obviously minimal). Also, $\{\mathit{CountryName}, \mathit{LotNumber}\}^+$ equals U. For the third functional dependency we compute $\mathit{Area}^+ = \{\mathit{Area}, \mathit{Price}\}$, which is a proper subset of U. Hence, we have to check whether the right hand side occurs in any minimal key. This is the point where one has e. g. to know all keys of a relation schema. The minimal keys of R are $\{\mathit{PropertyId}\}$ and $\{\mathit{CountryName}, \mathit{LotNumber}\}$. Since *Price* does not occur in any key, the relation schema is not 3NF.

In the rest of this section we collect the necessary terms from relational database theory. An attribute A is an identifier for an element of some domain D. Let U be a set of attributes. An attribute set X over U is a subset of U. A functional dependency over U is an expression of the form $X \to Y$, where X, Y are attribute sets. Intuitively, a functional dependency $X \to Y$ means that the attribute set X determines the attribute set Y. If X, Y are attribute sets, then we write XY for $X \cup Y$. We denote by $\mathit{attr}(F)$ the set of all attributes occurring in F. All attribute sets and all sets of functional dependencies are finite. The cardinality of a set X is denoted by $|X|$.

We will use the following naming conventions: $A, B, C, D, \ldots$ for attributes, $X, Y, U, V, \ldots$ for attribute sets, and $F, G, \ldots$ for sets of functional dependencies.

A relation schema $R = \langle U, F\rangle$ consists of an attribute set U and a set F of functional dependencies over U. There are distinguished subsets $K \subseteq U$, called superkeys. To define superkeys we use the algorithm *transitive closure* below. The algorithm *transitive closure* computes for an attribute set X the set $X^+ \supseteq X$ of all attributes which are functional determined by X.

Algorithm *transitive closure*.
Input: A relation schema $R = \langle U, F \rangle$ and an attribute set $X \subseteq U$.
Output: X^+.

[INIT]	$X^+ := X;$
[LOOP]	while $(\exists (Y \to Z) \in F : Y \subseteq X^+ \ \& \ Z \not\subseteq X^+)$
	$\quad X^+ := X^+ \cup Z;$
[RESULT]	return $X^+;$

Figure 1: Algorithm *transitive closure*

Now an attribute set $K \subseteq U$ is a *superkey* of R, if $K^+ = U$. A superkey K of R is a *key* of R, if K is minimal with respect to set inclusion. Keys are also known as candidate keys. We denote by $\mathcal{K}_R$ the set of all keys of the relation schema R. For a computation of X^+ we denote the LOOP–steps by $X^{(0)}, X^{(1)}, X^{(2)}, \ldots$ and so on.

When the right hand side of a functional dependency is a singleton set, then we use the notation $X \to A$, $Y \to B$, $Z \to C$ or similar. We call such functional dependencies *unit* functional dependencies. A functional dependency $X \to Y$ is trivial, if $Y \subseteq X$.
Let $Y \to B$ be a unit functional dependency. To indicate that the attribute A occurs in the left hand side of $Y \to B$, we write $YA \to B$. Additionally, when we use the notation $YA \to B$, then we assume $A \notin Y$, that is, the union YA is disjoint. In this paper we work with unit functional dependencies. It is no restriction to consider only unit functional dependencies, see [Maier 1983] p. 77 Lemma 5.3. Further, for a relation schema $R = \langle U, F \rangle$ we always assume that $U = attr(F)$. This is no restriction when considering keys, because the attributes in $U - attr(F)$ have to be in every key of R. Summing up: For all relation schemas $R = \langle U, F \rangle$ in this paper we assume that

- $U = attr(F)$ and
- F is a set of non–trivial unit functional dependencies.

2 The Inference System $\mathbb{K}$

Let $R = \langle U, F \rangle$ be a relation schema. In [Wastl 1998] the Hilbert style inference system $\mathbb{K}$ has been introduced for deriving all keys of a database relation schema R. The entities which are derived with $\mathbb{K}$ are functional dependencies. The inference system $\mathbb{K}$ depends on R. The axioms and rules of inference are given below. Note that by our convention the functional dependencies in F are non-trivial unit functional dependencies.

Axioms of $\mathbb{K}$

$\varnothing \to \varnothing$

$X \to A \quad$ if $(X \to A) \in F$

Rules of inference of $\mathbb{K}$

$$\mathbb{K}1. \qquad \frac{X \to A \qquad YA \to B}{XY \to B}$$

$$\mathbb{K}2. \qquad \frac{X \to A \qquad Y \to B}{XY \to B}$$

The axioms of $\mathbb{K}$ are essentially the functional dependencies of F. The axiom of the form $\varnothing \to \varnothing$ is only needed when $F = \varnothing$. Then $\varnothing$ is the only key of R. Note that $U = attr(F)$ and so, $F = \varnothing$ implies $U = \varnothing$. Note also, that in the inference rule $\mathbb{K}2$ the two functional dependencies in the premise can be swapped and thus, one can also derive the functional dependency $XY \to A$.

The inference rules of $\mathbb{K}$ have two premises and one conclusion. A $\mathbb{K}$-derivation of the functional dependency $X \to A$ from F, denoted by $F \vdash_{\mathbb{K}} X \to A$, is defined in the usual way. That is, a derivation $F \vdash_{\mathbb{K}} X \to A$ starts with axioms from $\mathbb{K}$. Then one derives functional dependencies using axioms from $\mathbb{K}$ or functional dependencies which have been derived by previous steps. The length of a derivation $F \vdash_{\mathbb{K}} X \to A$ is defined as the number of inference steps with $\mathbb{K}1$ or $\mathbb{K}2$. By a $\mathbb{K}1$-derivation ($\mathbb{K}2$-derivation) we mean a $\mathbb{K}$–derivation where only the inference rule $\mathbb{K}1$ ($\mathbb{K}2$) is used.

Example 2
Let $R = \langle U, F \rangle$, where $U = \{A, B, C, D, E, H, K\}$ and

$$F = \{ \begin{array}{lcl} AB & \to & C, \\ DC & \to & E, \\ H & \to & K \,\}. \end{array}$$

The following is a derivation of length 2 for the unique key $ABDH$. Note that the entities in a derivation are functional dependencies, and that deriving the key $ABDH$ means deriving a functional dependency with left hand side $ABDH$.

$$[\mathbb{K}2] \quad \frac{[\mathbb{K}1] \quad \dfrac{AB \to C \qquad DC \to E}{ABD \to E} \qquad H \to K}{ABDH \to K}$$

A relation schema $R = \langle U, F\rangle$ is transitive, if F is closed with respect to the inference rule $\mathbb{K}1$. That is, for any two functional dependencies $X \to A$, $YA \to B$ from F, the functional dependency $XY \to B$ is in F, provided that $XY \to B$ is not trivial. In general, a relation schema $R = \langle U, F\rangle$ is not transitive. But there exists a unique transitive relation schema $R^+ = \langle U, tc(F)\rangle$ such that

- $F \subseteq tc(F)$, and
- K is a key of R if and only if K is a key of R^+.

The following statements are taken from [Wastl 1998].

Lemma 3
Let $R = \langle U, F\rangle$ be a transitive relation schema and K be a key of R. Then,

$$K^+ = K \uplus \{A \in U \mid \exists X \to A \in F \text{ such that } X \subseteq K\}.$$

Remark: the symbol $\uplus$ means disjoint union. So, the relation above gives a partition of U.

Theorem 4 (Completeness of $\vdash_{\mathbb{K}}$)
Let $R = \langle U, F\rangle$ be a relation schema. Then, for every key K of R there exists a derivation $F \vdash_{\mathbb{K}} K \to A$, where $A \in U$ or $A = \varnothing$. $\square$

3 The Concept of Tableau Proof for Keys

We will first sketch the idea of using the inference rules $\mathbb{K}1$ and $\mathbb{K}2$ for defining a tableau whose branches are formal derivations (in the inference system $\mathbb{K}$) for keys of a relation schema. The tableau T to be defined for a relation schema $R = \langle U, F\rangle$ should be characterized by the following three items:

- The nodes of T are functional dependencies over U.
- Each branch in T is a formal derivation in the inference system $\mathbb{K}$.
- For each node $X \to A$ in T the path from the root of T to $X \to A$ is a derivation $F \cup \{\text{root of } T\} \vdash_{\mathbb{K}} X \to A$, where only the inference rule $\mathbb{K}1$ is used.

The aim is then that the tableau T should enjoy the following properties:

- For each leaf $X \to A$ of T, X is a super key of R.
- For every key K there is a leaf node $K \to A$ (completeness property).

Now, the construction of a tableau T for the relation schema $R = \langle U, F\rangle$ goes as follows: First determine the root of T. To this end, let

$$F = \{X_1 \to A_1, X_2 \to A_2, \ldots, X_{n-1} \to A_{n-1}, X_n \to A_n\};$$

Build a $\mathbb{K}2$-derivation

$$[\mathbb{K}2] \quad \dfrac{X_1 \to A_1 \qquad X_2 \to A_2}{X_1X_2 \to A_2}$$

$$\ddots$$

$$[\mathbb{K}2] \quad \dfrac{X_1 \ldots X_{n-1} \to A_{n-1} \qquad X_n \to A_n}{X_1 \ldots X_n \to A_n},$$

where only the inference rule $\mathbb{K}2$ is used to collect the left hand sides of all functional dependencies in F. The left hand side of the resulting functional dependency $X_1 \ldots X_n \to A_n$ has the properties

- $(X_1 \ldots X_n)^+ = U$, that is, $X_1 \ldots X_n$ is a superkey, and
- $\bigcup_{K \in \mathcal{K}_R} \subseteq X_1 \ldots X_n$, that is, all keys of R are included in $X_1 \ldots X_n$.

This is the point where tableaux come into play. Using tableaux we specialize $X_1 \ldots X_n$ to all keys of R using only the inference rule $\mathbb{K}1$. Remark: a similar idea has been used in [Gottlob 1987] for computing covers for embedded functional dependencies.

The root of the tableau T is the functional dependency $X_1 \ldots X_n \to A_n$. Let $Z \to C$ be any node in T. Then $Z \to C$ has the functional dependency $(Z - A_i)X_i \to C$ as a successor node and the edge between $Z \to C$ and $(Z - A_i)X_i \to C$ is labeled with A_i, if and only if there exists a functional dependency $X_i \to A_i \in F$ such that condition (1) and (2) holds:

(1) $A_i \in Z$,

(2) $X_i \cap L = \varnothing$, where L is the union of the edge labels on the (unique) path from the root to the node $Z \to C$.

Graphically this looks like

$$\begin{array}{c} Z \to C \\ \Big\vert\, A_i \\ (Z - A_i)X_i \to C\,. \end{array}$$

Observe that this conforms with one inference step with the inference rule $\mathbb{K}1$. Let $Z = Z'A_i$. Then we have

$$[\mathbb{K}1] \quad \frac{X_i \to A_i \qquad Z'A_i \to C}{X_i Z' \to C}.$$

Condition (1) is needed to apply the inference rule $\mathbb{K}1$, and condition (2) ensures that no attribute which has been removed from $X_1 \dots X_n$ on the path from the root to $Z \to C$ will be introduced again through X_i. This can be seen as a kind of loop checking (cf. [Lifschitz et al. 1995]). The set of attributes removed from $X_1 \dots X_n$ on the path from the root to $Z \to C$, denoted by L, is the union of the edge labels on that path.

So far we have explained the concepts. We give an example.

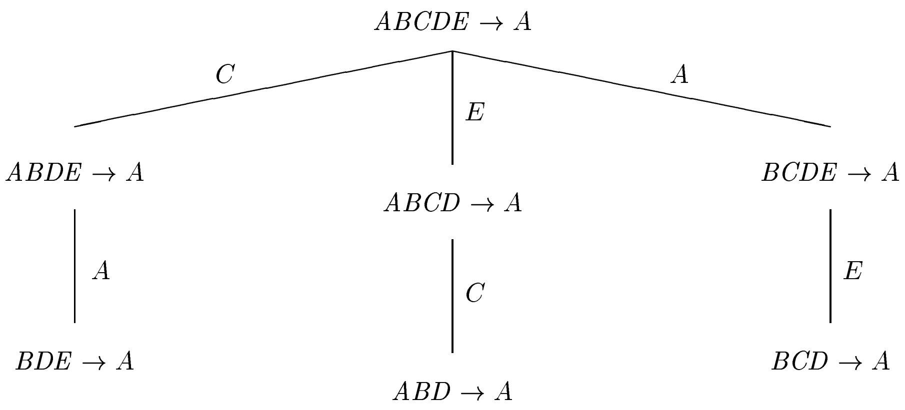
Example 5
Let $R = \langle U, F \rangle$, where $U = \{A, B, C, D, E\}$ and

$$\begin{array}{rlcl} F = \{ & AB & \to & C, \\ & CD & \to & E, \\ & E & \to & A\}. \end{array}$$

There are three keys: BDE, BCD and ABD. In a first step we determine the root of the tableau T with the $\mathbb{K}2$–derivation below.

$$[\mathbb{K}2] \quad \cfrac{[\mathbb{K}2] \quad \cfrac{AB \to C \qquad CD \to E}{ABCD \to E} \qquad E \to A}{ABCDE \to A}$$

The tableau T is the following tree:



The the left hand sides of the leafs of T are exactly the keys of R. Each branch is a $\mathbb{K}1$–derivation from $F \cup \{ABCDE \to A\}$. We discuss the left branch; the argumentation for the other branches is similar. The node $ABDE \to A$ is a successor node of the root $ABCDE \to A$, because there exists a functional dependency $AB \to C$ in F such that (1) $C \in ABCDE$ and (2) $AB \cap L = \varnothing$, because $L = \varnothing$. Recall that the attribute set L is the union of the edge labels from the root to $ABCDE \to A$, which is the root in this case. When considering the successor nodes of $ABDE \to A$, we have $L = \{C\}$. There is only one functional dependency in F whose right hand side occurs in $ABDE$ and whose left hand side is disjoint to L, namely $E \to A$. Thus, node $ABDE \to A$ has the sole successor node $BDE \to A$, and L will be $\{A, C\}$. Now, none of the functional dependencies in F fulfills condition (1) and (2), and so, $BDE \to A$ has no successor nodes.

We close this section with a technical remark. We have defined the nodes of a tableau as functional dependencies. This is good for the proofs in the following, because each branch in a tableau is a $\mathbb{K}$-derivation which starts in a leaf and ends in the root. On the other hand, when applying tableau one can simply work with the left hand sides. Note that the tableau in the example above could have also been constructed whitout the right hand sides. So, in the following examples we will omit the right hand sides.

4 $\mathbb{K}$-Tableaux

The definition of a $\mathbb{K}$-tableau for a relation schema below collects the properties which have been developed in the previous section.

Definition 6
Let $R = \langle U, F \rangle$ be a relation schema. A $\mathbb{K}$-*tableau* for R is a non-empty tree whose nodes are functional dependencies over U and whose edges are labeled with attributes from U such that the following two conditions hold:

(1) The root $V \to D$ of T fulfills

- $\bigcup_{K \in \mathcal{K}_R} \subseteq V$ (this implies $V^+ = U$),

and

(2) for every node $Z \to C$ of T the following condition is true: the functional dependency $(Z - A_i)X_i \to C$ is a successor node of $Z \to C$ if and only if there exists a functional dependency $X_i \to A_i$ in F such that

(2a) $A_i \in Z$,

(2b) $X_i \cap L = \varnothing$, where L is the union of the edge labels on the path from the root to the node $Z \to C$, and

(2c) the edge from $Z \to C$ to $(Z - A_i)X_i \to C$ is labeled with A_i.

In the following we will consider $\mathbb{K}$-tableaux T for relation schemas $R = \langle U, F\rangle$ where the left hand side of the root of T is the union of the left hand sides of the functional dependencies in F. The union of the left hand sides of the functional dependencies in F trivially fulfills condition (1) of Definition 6. Then, the $\mathbb{K}$-tableaux for relation schemas are unique modulo permutation of branches, and therefore, we will speak of *the* $\mathbb{K}$-tableau of a relation schema. As a special case the empty relation schema $R = \langle\varnothing, \varnothing\rangle$ has tableau $\varnothing \to \varnothing$. The edge lable set L depends on the nodes in T. Sometimes it will be appropriate to make this dependency explicit; then we index the set L with nodes, for example, $L = L_{Z\to C}$.

Lemma 7
Let $R = \langle U, F\rangle$ be a relation schema and T be the $\mathbb{K}$-tableau for R. Then, the left hand side of every node in T is a superkey of R.

Proof. We make an induction on the depth $d \geq 1$ of the tableau[1] T. For the inductive basis let $d = 1$. Then, the statement of the lemma holds, because the transitive closure of the left hand side of the root equals U. So, assume as inductive hypothesis that the lemma holds for tableaux of depth ≥ 1. Let R be a relation schema whose tableau T has depth $d + 1$. Let $(Z - A_i)X_i \to C$ be a leaf of T which has depth $d + 1$, and let $Z \to C$ be its parent node (which exists, because $d + 1 > 1$). Let A_i be the edge label between parent $Z \to C$ and successor $(Z - A_i)X_i \to C$. Then, according to Definition 6, there exists a functional dependency $X_i \to A_i$ in F such that $A_i \in Z$, and we have

$$[\mathbb{K}1] \quad \frac{X_i \to A_i \qquad \overbrace{Z'A_i}^{=Z} \to C}{(Z - A_i)X_i \to C}\,.$$

By inductive hypothesis, Z is a superkey of R, that is $Z^+ = U$. We must show $\Big((Z - A_i)X_i\Big)^+ = U$. Therefore, it is enough to observe that $X_i \subseteq (Z - A_i)X_i$. This yields $A_i \in \Big((Z - A_i)X_i\Big)^+$. Now $Z^+ = U$ implies $\Big((Z - A_i)X_i\Big)^+ = U$. $\square$

Proposition 8
Let $R = \langle U, F\rangle$ be a relation schema and T be the $\mathbb{K}$-tableau for R. Then, for every node $Z \to C$ of T there is

(a) $Z \cap L_{Z\to C} = \varnothing$, and

(b) $Z \uplus L_{Z\to C} =$ left hand side of the root node.

[1] We usually write tableau instead of $\mathbb{K}$-tableau.

Proof. A straightforward induction on the depth of the tableau T. Note that $L_{Z \to C}$ is uniquely determined through $Z \to C$. For (a) note also that F does not contain trivial functional dependencies. $\square$

The theorem below states that the $\mathbb{K}$-tableau for a *transitive* relation schema is sound and complete for keys.

Theorem 9
Let $R = \langle U, F \rangle$ be a transitive relation schema and T be the $\mathbb{K}$-tableau for R.

(a) For every leaf $X \to A$ in T, X is a $\subseteq$–minimal key of R.

(b) For every key K of R there exists a leaf $K \to A$ in T.

Proof. (a) Let $X \to A$ be a leaf of T. By Lemma 7 we have $X^+ = U$. So, it remains to show that X is $\subseteq$–minimal. We show this by contradiction and assume to this end that X is a proper superkey. Then there exists a key K of R such that $K \subset X$. Since R is transitive, by Lemma 3, there exists a functional dependency $Y \to B$ in F such that (i) $B \in X - K$ and (ii) $Y \subseteq K \subset X$. By Proposition 8 (a) we have $X \cap L_{X \to A} = \varnothing$. Together with (ii) we obtain $Y \cap L_{X \to A} = \varnothing$. Since by (i) $B \in X$, we get

$$[\mathbb{K}1] \quad \frac{Y \to B \qquad \overbrace{X'B}^{=X} \to A}{(X - B)Y \to A},$$

that is, the leaf $X \to A$ has the successor node $(X - B)Y \to A$. This is a contradiction.

(b) Choose a key K of R. We have to find a branch from the root of T to a leaf $K \to A$. Let $V \to D$ be the root of T, and let $V - K = \{A_1, \dots, A_n\}$. Then, by Lemma 3 there exists n functional dependencies

$$X_1 \to A_1, \dots, X_n \to A_n$$

in F such that

- $X_i \subseteq K$, for all $1 \leq i \leq n$.

If $n = 0$, then V is the unique key of R. Hence, $K = V$ and the theorem is proved. We proceed under the assumption $n > 0$. Then, we get a $\mathbb{K}1$–derivation $F \cup \{V \to D\} \vdash_{\mathbb{K}} K \to D$ as follows:

$$[\mathbb{K}1] \quad \frac{X_1 \to A_1 \qquad \overbrace{V'A_1}^{=V} \to D}{(V - A_1)X_1 \to D}$$

$$\ddots$$

$$[\mathbb{K}1] \quad \frac{X_n \to A_n \qquad \overbrace{(V - A_1 \dots A_{n-1})}^{\text{contains } A_n} X_1 \dots X_{n-1} \to D}{\underbrace{(V - A_1 \dots A_{n-1} A_n) X_1 \dots X_{n-1} X_n}_{=K} \to D}.$$

Since $A_i \in U - K$ and $X_i \subseteq K$ for all $1 \leq i \leq n$ we get

- $X_i \cap L_{(V-A_1 \dots A_i)X_1 \dots X_i \to D} = \varnothing$.

Hence, the $\mathbb{K}1$–derivation displayed above is a branch in T by Definition 6. To complete the proof we must show that there exists no functional dependency $Y \to B$ in F with right hand side in K such that $Y \cap L_{K \to D} = \varnothing$. Intuitively, the functional dependency $Y \to B$ could be used to lengthen the $\mathbb{K}1$–derivation above and then, the left hand side of the successor node would not include K. We falsify the existence of such a functional dependency in the claim below. This will prove the theorem.

Claim: There exists no functional dependency $Y \to B$ in F with right hand side in K such that $Y \cap L_{K \to D} = \varnothing$.

We prove the claim by contradiction. To this end let $Y \to B \in F$ with right hand side in K such that $Y \cap L_{K \to D} = \varnothing$. We show $Y \subseteq K$. First note that V equals the union of the left hand sides of the functional dependencies in F; Therefore, $Y \subseteq V$. By Proposition 8 (b) we have $K \uplus L_{K \to D} = V$. Together, this shows, $Y \subseteq K$. Further, $Y \to B \in F$ implies that the functional dependency $Y \to B$ is not trivial, that is, $B \notin Y$. Using the hypothesis $Y \cap L_{K \to D} = \varnothing$ we conclude with Definition 6 that the node

$$K \to D$$

has the successor node

$$(K - B)Y \to D.$$

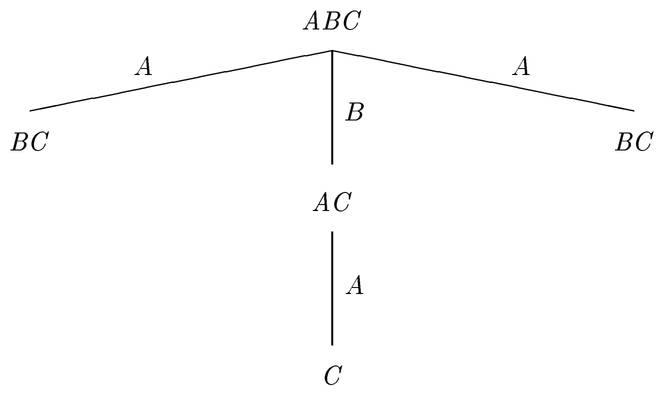
This node can be written as $(K - B) \to D$, because $B \notin Y$ and $Y \subseteq K$, as observed above. By Lemma 7, $K - B$ is a super key of R. This is a contradiction. $\square$

It is interesting to note that Theorem 9 (a) fails, if the relation schema is not transitive.

Example 10
Let $R = \langle U, F \rangle$, where $U = \{A, B, C\}$ and

$$F = \{ \begin{array}{ccc} C & \to & A, \\ A & \to & B, \\ B & \to & A \}. \end{array}$$

There is a unique key C. The tableau T is the following tree:



The transitive closure of F is $tc(F) = F \cup \{C \to \underline{B}\}$. Thus, when adding the the following two edges and nodes to the tree T above we get a tableau for R^+.

Theorem 11
Let $R = \langle U, F \rangle$ be a relation schema and T be the $\mathbb{K}$-tableau for R.

(a) For every leaf $X \to A$ in T, X is a super key of R.

(b) For every key K of R there exists a leaf $K \to A$ in T.

Proof. (a) By Lemma 7.

(b) Fix a key K of R. Let $V \to D$ be the root of T. Assume that

$$V - K = \{A_1, \ldots, A_n\}.$$

If $n = 0$, then V is the unique key of R. Hence, $K = V$ and the theorem is proved. We proceed under the assumption $n > 0$. Since K is a key there exists functional dependencies

$$Z_1 \to A_1, \ldots, Z_n \to A_n$$

in F such that

- $A_i \neq A_j$ for all $1 \leq i < j \leq n$,

- $K^{(n)} = V$, and
- $Z_i \subseteq KA_1 \ldots A_{i-1}$ for all $1 \leq i \leq n$.

This yields a $\mathbb{K}1$–derivation

$$[\mathbb{K}1] \quad \frac{Z_n \to A_n \qquad \overbrace{V'A_n}^{=V} \to D}{(V - A_n)Z_n \to D}$$

$$\ddots$$

$$[\mathbb{K}1] \quad \frac{Z_1 \to A_1 \qquad \overbrace{(\ldots((V - A_n)Z_n - A_{n-1})Z_{n-1} - \ldots - A_2)Z_2}^{\text{contains } A_1} \to D}{\underbrace{(\ldots(((V - A_n)Z_n - A_{n-1})Z_{n-1} - \ldots - A_2)Z_2 - A_1)Z_1}_{=K} \to D},$$

such that for all $1 \leq i \leq n$

- $Z_i \cap L_{(\ldots(V-A_n)Z_n-\ldots-A_i)Z_i \to D} = \varnothing$
(Note that $L_{(\ldots(V-A_n)Z_n-\ldots-A_i)Z_i \to D} = \{A_i, \ldots, A_n\}$ and $L_{V \to D} = \varnothing$).

By Definition 6 this $\mathbb{K}1$–derivation corresponds to a branch in T beginning in the root $V \to D$ of T. To complete the proof we must show that there exists no functional dependency $Y \to B$ in F with right hand side in K such that $Y \cap L_{K \to D} = \varnothing$. This is done as in the proof of Theorem 9. □

5 Linear Derivations for Keys

A $\mathbb{K}$-derivation is *linear* if the length is zero and the derivation consists of exactly one axiom, or the length is > 0 and the derivation has the form

$$\text{Axiom, Axiom, } \mathbb{K}1/\mathbb{K}2,\ \text{Axiom, } \mathbb{K}1/\mathbb{K}2, \ldots, \text{Axiom, } \mathbb{K}1/\mathbb{K}2,$$

that is, the derivation starts with two axioms from F followed by an application of $\mathbb{K}1$ or $\mathbb{K}2$; the derived functional dependency together with an axiom from F are then the premises of another application of $\mathbb{K}1$ or $\mathbb{K}2$, and so on.

A linear $\mathbb{K}$-derivation is a *linear* $\mathbb{K}1$*-derivation* ($\mathbb{K}2$*-derivation*), if only the inference rule $\mathbb{K}1$ ($\mathbb{K}2$) is used.

Theorem 12
Let $R = \langle U, F \rangle$ be a relation schema. Then, for every key K of R and every attribute $D \in U$ there exists a linear

(a) $\mathbb{K}1$-derivation $F \cup \{U \to D\} \vdash_{\mathbb{K}} K \to D$ of length $\leq |F|$.

(b) $\mathbb{K}$-derivation $F \vdash_{\mathbb{K}} K \to D$ of length $\leq 2|F|$.

Proof. (a) Let T be the tableau of R and $V \to D$ be its root. Then V equals the union of the left hand sides of the functional dependencies in F. Hence, attributes in $U-V$ occur only in the right hand side of functional dependencies in F. Now choose a key K of R. At first we make a linear $\mathbb{K}1$-derivation $F \cup \{U \to D\} \vdash_{\mathbb{K}} V \to D$. This derivation is obtained by removing all attributes in $U-V$ with $|U-V|$ $\mathbb{K}1$ steps. This is possible because attributes in $U-V$ occur only in the right hand side of functional dependencies in F. In a second step we apply Theorem 11 (b) and get a linear $\mathbb{K}1$-derivation $F \cup \{V \to D\} \vdash_{\mathbb{K}} K \to D$. Together, this yields a linear $\mathbb{K}1$-derivation $F \cup \{U \to D\} \vdash_{\mathbb{K}} K \to D$ of length $\leq |F|$.

(b) At first make a linear $\mathbb{K}2$-derivation $F \vdash_{\mathbb{K}} V \to D$. Then apply Theorem 11 (b). $\square$

Example 13
Let $R = \langle U, F\rangle$, where $U = \{A, B, C, D\}$ and

$$F = \{\begin{array}{rcl} A & \to & B, \\ B & \to & A, \\ BC & \to & D, \\ D & \to & A\,\}. \end{array}$$

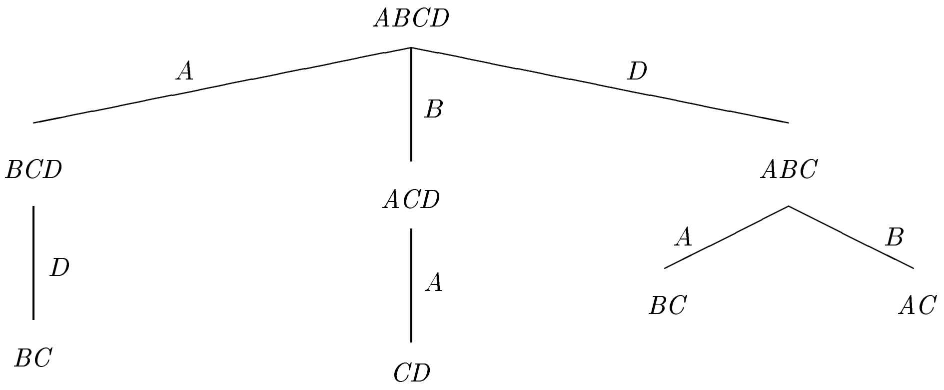
There are three keys: BC, AC and CD. The following is a linear $\mathbb{K}2$-derivation $F \vdash_{\mathbb{K}} ABCD \to A$ for determining the root of the tableau T:

$$[\mathbb{K}2]\quad \dfrac{[\mathbb{K}2]\quad \dfrac{A \to B \qquad BC \to D}{ABC \to D} \qquad D \to A}{ABCD \to A}$$

The following is a linear $\mathbb{K}1$-derivation $F \cup \{ABCD \to A\} \vdash_{\mathbb{K}} CD \to A$:

$$[\mathbb{K}1]\quad \dfrac{D \to A \qquad [\mathbb{K}1]\quad \dfrac{A \to B \qquad ABCD \to A}{ACD \to A}}{CD \to A}$$

The tableau T is the following tree:



6 Conclusion

Using the concept of tableaux and the inference system $\mathbb{K}$ we have shown that $\mathbb{K}$-tableaux are complete for keys of a relation schema. The branches of a $\mathbb{K}$-tableau are linear $\mathbb{K}1$-derivations. So, we get linear derivations for keys whose length is bounded by the number of axioms.

Acknowledgement. I am grateful to the referees for their suggestions to improve the paper. In particular, the second referee has given several detailed hints which improve the readability of the paper a lot.

References

[Elmasri et al. 1994] Elmasri, R., Navathe, S.B.: "Fundamentals of Database Systems"; Second edition, Benjamin/Cummings Pub. - Redwood City, Calif. (1994).

[Gottlob 1987] Gottlob, G.: "Computing Covers for Embedded Functional Dependencies"; *Symposium on Principles of Database Systems (PODS) '87*, 58-69 (1987).

[Lifschitz et al. 1995] Lifschitz, V., McCain, N., Przymusinski, T.C. and Stärk, R.F.: "Loop checking and the well-founded semantics"; In V.W. Marek et. al. (eds), *Proceedings of the Third International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)* '95, Lexington, Kentucky, USA, Springer-Verlag, Lecture Notes in Artificial Intelligence 928, 127-142 (1995).

[Maier 1983] Maier, D.: "The Theory of Relational Databases"; Computer Science Press (1983).

[Ullman 1988] Ullman, J.D.: "Principles of Databases and Knowledge Bases, Vol. 1"; Computer Science Press (1988).

[Wastl 1998] Wastl, R.: "On the Number of Keys of a Relational Database Schema"; *Journal of Universal Computer Science (J.UCS)*, Vol. 4(5) (1998).