Functional Dependencies Between Variables

Natasha Alechina
School of Computer Science and Information Technology
University of Nottingham
Nottingham, NG7 2RD, England
E-mail: n.alechina@cs.nott.ac.uk

Abstract

We consider a predicate logic L_{fd} where not all assignments of values to individual variables are possible. Some variables are functionally dependent on other variables. This makes sense if the models of logic are assumed to correspond to databases or states. We show that L_{fd} is undecidable but has a complete and sound sequent calculus formalisation.

Keywords: quantifiers, assignments, functional dependencies.

1 Introduction

It is a standard practice to define the truth conditions for a first-order formula with a quantifier in terms of assignments to individual variables. For example, let M be a model and s a function (assignment) which assigns to variables elements in the domain of M. Then a formula $\exists x\phi(x)$ is true in M under s iff there is an assignment s' which differs from s at most in the value it assigns to x, such that $\phi(x)$ is true on M under s'. However, in classical predicate logic assignments are not essential; the same truth definition can be reformulated without mentioning them at all. For example, we can be slightly sloppy and just say: $\exists x\phi(x)$ is true in M if there is an element a in the domain of M, such that $\phi(a)$ is true in M. This is so because in classical predicate logic variables are just anonymous placeholders; every value which can be assigned to x can just as well be assigned to y, and the two formulas $\exists x P(x)$ and $\exists y P(y)$ have precisely the same meaning: there exists an element with the property P.

In some cases, we do not want variables to be completely anonymous. For example, in many-sorted logic some variables are assigned values from one domain (e.g. numbers) and other variables are assigned values from another domain (e.g. people). Furthermore, we might want the value of one variable to be dependent on the value of some other variable. This means that not all assignments of values to variables are possible.

The study of logics with restricted sets of assignments was initiated in [5] (in close connection with the study of cylindric relativised set algebras) and continued in many other papers. Usually, the restrictions on the set of assignments are motivated algebraically or from modal logic perspective and have a form of some closure conditions, for example closure under permutation. Many such logics, unlike standard first order logic, are decidable (see, for example, [3]).

Here, we look at the set of assignments satisfying some functional dependencies between variables. The most important motivation for this is the connection between models of first order logic and relational databases, which often satisfy some set of functional dependencies between attributes.

A relational database can be seen as a (collection of) tables. For example, a telephone directory database could be a table with the columns called `Name` and `Telephone Number`. These names of columns are called *attributes*. One can see the rows in the table as the collection of all possible values for the attributes, or all possible assignments to the variables x_1 and x_2 corresponding to the columns one and two. Assuming that every person has only one telephone number, there is an obvious a priori restriction on the set of assignments: there is no two assignments which agree on the value for x_1 and differ on the value for x_2. This is called a *functional dependency* between `Name` and `Telephone Number`, denoted by

$$\texttt{Name} \longrightarrow \texttt{TelephoneNumber}.$$

In what follows, we use both logical terminology (individual variables) and databases terminology (columns in a table and attribute names), in the spirit of [1].

In section 2 we introduce a logic L_{fd} where the set of possible assignments is restricted by functional dependencies. L_{fd} is strictly weaker than the classical predicate logic. In section 3 we prove that, unlike many other logics with restricted sets of assignments, L_{fd} is undecidable. In section 4 we show that again somewhat surprisingly L_{fd} admits complete and sound sequent calculus with natural introduction and elimination rules.

2 Language and models

Consider a first order many-sorted language $\mathcal{L}(I)$ which contains individual variables $(v_i)_{i\in I}$, where each variable belongs to a different sort (attribute names); constants $(a^i_n)_{n<\omega}$, for each sort $i \in I$; predicate symbols, including $=$, boolean connectives and quantifiers $\forall$ and $\exists$. The definition of a well formed formula is standard. In particular, if P is a predicate (intuitively) over sorts 1 and 2, we do not forbid expressions like $P(a, b)$ where a or b or both are of the wrong sort. This expression is going to be false, but meaningful. To see the reason for this, observe that any many-sorted system is in all aspects (but complexity) equivalent to a one-sorted system, via translation which takes $P(a, b)$ to $P(a, b) \wedge A_1(a) \wedge A_2(b)$ (where A_i are predicates used in the embedding language to denote the attribute domains).

An *fd-model* is of the form $M = \langle (A_i)_{i\in I}, \sigma, F\rangle$. Here, A_i are non-empty sets (attribute domains, not necessarily disjoint) and F is a set of functions between the A_i's, which are used to define the set of allowed assignments S:

$$S = \{s \in \prod_{i\in I} A_i : \forall i, j(f_{ij} \in F \rightarrow s_j = f_{ij}(s_i))\}.$$

Sometimes we will use S instead of F to specify a model. Finally, σ is a function which assigns predicates and constants their meaning: $\sigma(a) \in A_i$, if a is a constant of sort i, and $\sigma(P) \subseteq S\lceil i_1, \ldots, i_n$, where P is a predicate over attributes $i_1, \ldots, i_n$. In other words, σ assigns P the set of tuples from $A_{i_1} \times \cdots \times A_{i_n}$ satisfying the restrictions imposed by F.

An fd-model corresponds to a database possibly satisfying some functional dependencies between attributes. If a function $f : A_1 \rightarrow A_2$ is in F, then the database satisfies a functional dependency $A_1 \rightarrow A_2$. It is incorrect to call f itself a *functional dependency* since it is a particular instance of a dependency; but for the sake of brevity, where there can be no misunderstanding about the meaning, such concrete functions will be called functional dependencies.

Now we define what it means that $M = \langle (A_i)_{i \in I}, \sigma, S\rangle$ satisfies a formula φ.

$M, s \models P(t_1, \ldots, t_n) \Leftrightarrow \langle [t_1]_{M,s}, \ldots, [t_n]_{M,s}\rangle \in \sigma(P)$, where $[t_i]_{M,s}$ is $s(x)$ if t_i is an individual variable x and $\sigma(c)$ if t_i is a constant c.

$M, s \models t_1 = t_2 \Leftrightarrow [t_1]_{M,s} = [t_2]_{M,s}$.

The clauses for the booleans are standard. For the quantifiers, we have

$M, s \models \exists v_i \psi \Leftrightarrow \exists s' \in S(s' =_i s \wedge M, s' \models \psi)$, where $s' =_i s$ means that s' agrees with s on all coordinates except for (possibly) the ith coordinate.

$M, s \models \forall v_i \psi \Leftrightarrow \forall s' \in S(s' =_i s \Rightarrow M, s' \models \psi)$.

As usual, $M \models \varphi$ (φ is true in M) if for all $s \in S$, $M, s \models \varphi$; $\models \varphi$ (φ is valid) if φ is true in all models; $\Gamma \models \varphi$, where Γ is a set of formulas, if $\forall M \forall s(M, s \models \Gamma \Longrightarrow M, s \models \varphi)$. Let us call the corresponding logic L_{fd}.

Observe that the notion of validity defined above is rather weak. A formula is valid if it is true in all models where the set of assignments is restricted by some set of functions (which may be empty).

Sometimes we shall also refer to validity in a class of models satisfying some functional dependency of the form $A_i \to A_j$, for all possible concrete functions f_{ij}.

3 Undecidability

Before proving that the set of validities of L_{fd} is undecidable, we study some properties of fd-models which will be useful later.

Normally, logics with restricted sets of assignments do not have the property of locality, which says that if two assignments s and z agree on the free variables of a formula φ, then they agree on the truth value of φ. Also L_{fd} does not have this property. Consider for example a model where the set of allowed assignments is $S = \{aa, bb\}$ (the domain contains two elements a and b and a functional dependency of the form $v_0 = v_1$ holds). Let P hold of b and not hold of a. Then both assignments agree on the free variables of $\exists v_0 P(v_0)$ (as there are no free variables there), but bb satisfies it and aa not (because ba is not in S). But we'll see later that locality almost holds, that is, every

fd-satisfiable formula without constants has an fd-model which is local. In this example, the model consists of $S = \{bb\}$, $A = \{b\}$, and $F = \{id\}$.

In what follows, $s^{x_1,\ldots,x_n}_{d_1,\ldots,d_n}$ denotes the assignment which assigns d_i to x_i for all $i \in \{1, \ldots, n\}$ and agrees with s on the values for all other variables.

The following lemma shows that, given the presence of some assignments in an fd-model, we can infer that some other assignments are present, too.

Lemma 3.1 *Let $M = \langle (A_i)_{i\in I}, \sigma, F\rangle$ be an fd-model, S its set of assignments, and $s \in S$. Then $s \in S$, $s^x_d \in S$ implies $s^{xy}_{de} \in S$ iff $s^y_e \in S$.*

Proof. Suppose s^x_d and s^{xy}_{de} are in S and s^y_e is not. The latter means that s^y_e does not satisfy some functional dependency, which demands, for some i, either that $f(s(y)) = s_i$ or that $f(s_i) = s(y)$. Suppose that x is not v_i. Then, since s^{xy}_{de} does not differ from s^y_e on the value of v_i, the same functional dependency would be violated there; but it is not. So, there is a dependency between x and y. But then if it demands that $f(s(x)) = s(y)$, and s^x_d satisfies this dependency, $f(d) = s(y)$, and since s^{xy}_{de} satisfies it, $f(d) = e = s(y)$, so obviously $s^y_e = s \in S$. If the dependency demands $f(s(y)) = s(x)$, we have $f(e) = d$ and $f(s(y)) = d$, so $f(s(y)) = f(e)$ and f^y_e satisfies this dependency.

Suppose s^x_d and s^y_e are in S, and s^{xy}_{de} is not. Again this means that some functional dependency is violated; and again it must involve both x and y, otherwise it would have been violated in s^x_d or s^y_e. Suppose $f(s(x)) = s(y)$. Then $f(d) = s(y)$ (from $s^x_d \in S$) and $f(s(x)) = e$ (from $s^y_e \in S$). This implies that $f(d) = f(s(x)) = e$, so the dependency is satisfied in s^{xy}_{de}. Analogously for $f(s(y)) = s(x)$. □

Let $s, z \in S$. Then z is *accessible* from s in S in n steps if $z = s^{x_1,\ldots,x_n}_{d_1,\ldots,d_n}$ and $s^{x_1}_{d_1}$, $s^{x_1,x_2}_{d_1,d_2}$, $\ldots$, $s^{x_1,\ldots,x_{n-1}}_{d_1,\ldots,d_{n-1}} \in S$. The previous lemma implies that if $s^{x_i}_{d_i} \in S$ for every i, $1 \leq i \leq n$, then $s^{x_1,\ldots,x_n}_{d_1,\ldots,d_n} \in S$ and it is accessible from s.

The next lemma shows that if z is accessible from s and $z = s^{x_1,\ldots,x_n}_{d_1,\ldots,d_n}$, then $s^{x_i}_{d_i} \in S$ for all i, $1 \leq i \leq n$.

Lemma 3.2 *If $s^{x_1}_{d_1} \in S$, $s^{x_1,x_2}_{d_1,d_2} \in S$, $\ldots$, $s^{x_1,\ldots,x_{n-1}}_{d_1,\ldots,d_{n-1}} \in S$, $s^{x_1,\ldots,x_n}_{d_1,\ldots,d_n} \in S$, then $s^{x_n}_{d_n} \in S$.*

Proof. The previous lemma gives the proof for $n = 2$. Assume $n \geq 2$. We prove that if x_n can be changed to the value d_n after $n - 1$ changes to s, then it can be changed after $n - 2$ changes to s. Let $s' = s^{x_1,\ldots x_{n-2}}_{d_1,\ldots,d_{n-2}}$. By assumption, $s'^{x_{n-1}}_{d_{n-1}}$ and $s'^{x_{n-1},x_n}_{d_{n-1},d_n}$ are in S. By the previous lemma, $s'^{x_n}_{d_n}$ is in S. The same argument repeated $n-1$ times gives $s^{x_n}_{d_n} \in S$. $\square$

Lemma 3.3 *Every fd-satisfiable formula without constants has an fd-model where I is finite and equals to the number of variables used in the formula.*

Proof. Let $M, s \models \varphi$ for some fd-model $M = ((A_i)_{i \in I}, \sigma, F)$. For convenience, assume that $Var(\varphi) = \{v_1, \ldots, v_n\}$. Consider the set of sequences $S' = \{s' \in S : \forall j > n(s'_j = s_j)\}$ – that is, the set of all assignments in S which assign the same values to $v_{n+1}, v_{n+2}, \ldots$, as s. Let M' be a model where the only allowed assignments are those in S'. It is easy to check that $M, s \models \varphi$ iff $M', s \models \varphi$, since only the assignments in S' are relevant for determining the truth value of φ on s. Our next step is to transform the model further by restricting the set I to $\{1, \ldots, n\}$. Let $M" = ((A"_i)_{i \leq n}, \sigma", F")$, where $A"_i = \{e \in A_i : \exists s' \in S'(s'_i = e)\}$, and $\sigma"$ and $F"$ are the restrictions of σ and F to those smaller sets (note that σ' is undefined for some constants, therefore we need the assumption that φ does not contain constants). Observe that any two assignments in S' are accessible from each other iff their restrictions to the first n coordinates are, since they all have the same 'tails'. Again, an easy inductive argument shows that for every sequence $s' \in S'$, and for every subformula ψ of φ, $M', s' \models \psi$ iff $M", s' \lceil \{v_1, \ldots, v_n\} \models \psi$. This gives $M, s \models \varphi$ iff $M", s \lceil \{v_1, \ldots, v_n\} \models \varphi$. $\square$

Remark. Observe that the same proof works if φ does contain constants, but for every constant a in φ $\exists y(y = a)$ holds on M, s, where y is the variable of the same sort as a. Then $s^y_a \in S$ for all such a. Let now n be not the number of variables in φ, but of terms of different sorts. Then $s^y_a \in S'$ and the rest of the proof is as above.

Lemma 3.4 *If s is an assignment in S and $z \in S$ is accessible from s, then for every formula φ such that s and z agree on the free variables of φ, $s \models \varphi \Leftrightarrow z \models \varphi$.*

Proof. Assume s and z are accessible from each other, but there is a formula φ such that s and z agree on the free variables of φ, but $M, s \models \varphi$ and $M, z \not\models \varphi$. Without loss of generality we may assume that s, z and φ form a minimal such counterexample, i.e. no subformula of φ is a counterexample to locality (this is possible because atomic formulas can never be counterexamples to locality). An easy check shows that φ must be of the form $\exists x\psi$. Then there is an element d such that $M, s^x_d \models \psi$ and z^x_d is not an allowed assignment (otherwise it would have satisfied ψ, since we have assumed that φ is a minimal counterexample). But s and z are accessible from each other, so $s = z^{x_1,\ldots,x_n}_{d_1,\ldots,d_n}$, and $s^x_d = z^{x_1,\ldots,x_n,x}_{d_1,\ldots,d_n,d}$. But then by the lemmata above, z^x_d must be an allowed assignment. □

This implies that if in a fd-model all assignments are accessible from each other, then this model satisfies the locality property.

Lemma 3.5 *Any fd-satisfiable formula without constants φ with $Var(\varphi) = \{v_1, \ldots, v_n\}$ has an fd-model which satisfies locality and where the index set is $\{1, \ldots, n\}$. If φ contains constants, but for every constant a in φ $\exists y(y = a)$ is consistent with φ, then φ has a local model with I equal to the number of sorts in φ.*

Proof. Let M be an fd-model, and $s \in S$ an assignment in M. Then an *s-generated submodel* M_s of M is defined as follows. Let S' be the set of assignments in S which are accessible from s in a finite number of steps. For each A_i, $i \in I$, let A'_i be the set of elements of A_i which occur in the sequences from S'. Restrict σ and the functions in F to A'_i. (Note that for some constants σ' is not defined anymore. However, it is defined for all constants a such that $M, s \models \exists y(a = y)$, where y is the variable of the same sort as a, since in this case $\sigma(a)$ belongs to M_s.) The resulting structure $\langle (A'_i)_{i \in I}, \sigma', F' \rangle$ is clearly an fd-model. We show that if I is finite, than the set of assignments in M_s (denoted by S_s) is precisely S'. Clearly, every assignment in S' is in S_s. Assume that z is S_s. Note that every element d from A'_i occurs as s'_i in some s' accessible from s in a finite number of steps. This means that $z = s^{x_1,\ldots,x_I}_{d_1,\ldots,d_I}$, and $s^{x_1}_{d_1}, \ldots, s^{x_I}_{d_I}$ are S'. But then by the lemmata above, so is $z = s^{x_1,\ldots,x_I}_{d_1,\ldots,d_I}$. In a generated submodel any assignment is accessible from s. Since the accessibility relation is reflexive, symmetric and transitive, any assignment is accessible from any other assignment. So the generated model has the locality property. □

Theorem 3.6 *The set of validities in L_{fd} is undecidable.*

Proof. We show that for every first order sentence without constants φ there is an $\mathcal{L}(I)$ formula φ' such that φ is satisfiable in ordinary first-order logic iff φ' is L_{fd}-satisfiable. This means that a first-order formula $\neg\varphi$ is valid iff $\neg\varphi'$ is, and since the set of valid sentences of first-order logic is undecidable, so is the set of L_{fd}-validities.

The proof is now very easy (given the lemmata above) and repeats an argument given in [2] for another logic. Let φ be a first-order sentence without constants containing variables $x_1, \dots, x_n$. Then $\varphi' = \varphi \wedge \bigwedge_{1 \leq i,j \leq n} \forall x_i \exists x_j (x_i = x_j)$. Suppose φ is satisfiable in first-order logic. Then trivially φ' is satisfiable in first-order logic. A first-order model is a special case of an fd-model (where all A_i are the same and F is empty). So φ' is fd-satisfiable.

Suppose φ' is fd-satisfiable. There is some $M, s \models \varphi'$, where the set of indices equals $\{1, \dots, n\}$ and all assignments are accessible from s. Since $M, s \models \bigwedge_{i,j} \forall x_i \exists x_j (x_i = x_j)$, for every $i, j \leq n$, $\{e : s^i_e \in S\} = \{e : s^j_e \in S\}$. Since all assignments in S are accessible from s, this means that $\{e : \exists s' \in S(s'_i = e)\} = \{e' : \exists s' \in S(s'_j = e')\}$. Let M' be the model where $A_i = A_j = \{e' : \exists s' \in S(s'_i = e)\}$ and $F' = \emptyset$. It is easy to verify the claim that all assignments possible in M' are already possible in M, and $M', s \models \varphi$. But M' is an ordinary first-order model. $\square$

4 Sequent calculus

In this section we introduce a sequent calculus S_{fd} for the logic L_{fd}. S_{fd} contains an ordinary sequent calculus for first order logic, where the standard quantifier left- and right-introduction rules are replaced by the rules defined below. The main difference from the standard rules is that we require constants to be 'accessible' by introducing additional premises of the form $\exists y(y = a)$[1].

Let A be the set of constants occurring in $\Gamma \cup \Delta \cup \{\varphi(c)\}$, and c is a constant of the same sort as x_i.

$$\frac{\exists x_i(x_i = c), \Gamma \Longrightarrow \varphi(c), \Delta}{\exists x_i(x_i = c), \Gamma \Longrightarrow \exists x_i \varphi(x_i), \Delta} \exists r \qquad \frac{\bigwedge_{a \in A} \exists y(y = a), \Gamma, \varphi(c) \Longrightarrow \Delta}{\bigwedge_{a \in A} \exists y(y = a), \Gamma, \exists x_i \varphi(x_i) \Longrightarrow \Delta} \exists l$$

$$\frac{\bigwedge_{a \in A} \exists y(y = a), \Gamma \Longrightarrow \varphi(c), \Delta}{\bigwedge_{a \in A} \exists y(y = a), \Gamma \Longrightarrow \forall x_i \varphi(x_i), \Delta} \forall r \qquad \frac{\exists x_i(x_i = c), \Gamma, \varphi(c) \Longrightarrow \Delta}{\exists x_i(x_i = c), \Gamma, \forall x_i \varphi(x_i) \Longrightarrow \Delta} \forall l$$

Here, as usual, we require that c is free for x_i in φ, and in $\forall r$ and $\exists l$ c and x_i do not occur free in Γ and Δ.

[1] Premises of the form $\exists y(y = a)$ are reminiscent of E-logic (see [4]).

In addition to $\exists r, l$ and $\forall r, l$, S_{fd} contains two more axioms, $\varphi(x_i) \to \exists x_i \varphi(x_i)$ and $\forall x_i \varphi(x_i) \to \varphi(x_i)$, or two additional rules:

$$\frac{\Gamma \Longrightarrow \varphi(x_i), \Delta}{\Gamma \Longrightarrow \exists x_i \varphi(x_i), \Delta} \exists r' \qquad \frac{\Gamma, \varphi(x_i) \Longrightarrow \Delta}{\Gamma, \forall x_i \varphi(x_i) \Longrightarrow \Delta} \forall r'$$

Theorem 4.1 *1. S_{fd} is sound and complete for L_{fd}.*

2. For a class of fd-models characterised by specific functional dependencies a sound and complete calculus is obtained by adding to S_{fd} the following 'collapse' rule for all j such that $A_i \to A_j$ is in the set of dependencies: $\exists x_j \varphi \to \forall x_j \varphi$.

Proof. We start by proving (1). First we show that S_{fd} is sound. Before embarking on a formal proof, observe that the quantifiers are introduced only on the constants (variables) of the right sort; this corresponds to the fact that the renaming of bound variables is not valid in L_{fd}. The splitting of $\exists r$ and $\forall l$ rules is due to the fact that although $\varphi(x_i) \to \exists x_i \varphi(x_i)$ is valid in L_{fd}, $\varphi(c) \to \exists x_i \varphi[c/x_i]$ is, somewhat embarrassingly, not valid. The reason is that $s^{x_i}_{\sigma(c)}$ might not be in the set of allowed assignments. However, if $\exists x_i(x_i = c)$ is satisfied on s, $s^{x_i}_{\sigma(c)}$ is available.

We check that $\exists r$ and $\exists l$ preserve validity; the rules for the universal quantifier are dual.

First we show that if the premise of $\exists r$ holds, then so does the conclusion. Suppose Γ and $\exists x_i(x_i = c)$ are true on some M, s, and either a formula from Δ is true, or $\varphi(c)$ is true on M, s. If some formula from Δ is true, then the conclusion holds on M, s. Suppose all formulas from Δ are false and $\varphi(c)$ is true. We show that $\exists x_i \varphi(x_i)$ holds on M, s. Since $M, s \models \exists x_i(x_i = c)$, $s^{x_i}_{\sigma(c)} \in S$, $s^{x_i}_{\sigma(c)}$ is accessible from s, s and $s^{x_i}_{\sigma(c)}$ agree on the free variables of $\varphi(c)$, so $s^{x_i}_{\sigma(c)}$ satisfies $\varphi(c)$ by Lemma 3.4. An easy induction shows that $s^{x_i}_{\sigma(c)}$ satisfies $\varphi(x_i)$. Therefore $M, s \models \exists x_i \varphi(x_i)$.

Now we check the $\exists l$ rule. Suppose that on M, s Γ, $\exists x_i \varphi(x_i)$ and the conjunction of formulas stating that all constants in Γ, Δ and $\varphi(c)$ are 'accessible', holds, and all formulas in Δ fail (that is, the conclusion of the rule is not a valid inference). We show that then the premise is not a valid inference. The conjunction of formulas stating that all constants are accessible allows us to assume that M is a local model (by Lemma 3.5). Since $M, s \models \exists x_i \varphi(x_i)$, there is an element d such that $M, s^x_d \models \varphi(x_i)$. Change the interpretation of c in M so that $\sigma'(c) = d$. Again, an easy induction shows that $M', s^x_d \models \varphi(c)$, and by locality $M', s \models \varphi(c)$. Since c does not occur in Γ and Δ, the truth values of the formulas there are the same in M' as in M, so $M', s \models \Gamma, \varphi(c)$ and falsifies Δ: the premise is not a valid inference.

Next, we prove that the calculus is complete for the class of all fd-models. The proof uses the standard Henkin argument. Suppose $\Gamma \not\vdash \varphi$, then we build a model for $\Gamma \cup \{\neg\varphi\}$, which shows that $\Gamma \not\models \varphi$.

Let Δ be an S_{fd}-consistent set of sentences. We are building a term model for Δ. Fix some (well)-ordering of sentences in the language, $\varphi_1, \varphi_2, \ldots,$. Let

$\Sigma_0 = \Delta$,

$\vdots$

$\Sigma_n = \Sigma_{n-1} \cup \{\varphi_n\}$, if the resulting set is consistent, and Σ_{n-1} otherwise. If φ_n is of the form $\exists x\psi$, we also add $\psi(c)$, for a fresh constant c of the same sort as x, and, for every formula ψ with one free variable x, $\forall x\psi \to \psi(c)$, and $\exists x(c = x)$.

Let $\Sigma = \cup_n \Sigma_n$. First we show that it is consistent. Suppose it is not. Then a contradiction is derivable from some finite subset of Σ, which implies that for some n, Σ_n is inconsistent. Take the first such Σ_n. The contradiction can only arise as a result of adding a witness for some $\exists x\psi(x)$ and is derivable from some finite conjunction $\bigwedge_j \varphi_j$ of formulas from Σ, finite conjunction $\bigwedge_i(\forall x\theta_i \to \theta_i(c))$, $\exists x\psi(x)$ and $\psi(c)$, that is

$$\bigwedge_j \varphi_j, \exists x\psi(x) \Longrightarrow \bigwedge_i(\forall x\theta_i \to \theta_i(c)) \to \neg\psi(c)$$

We can assume that all formulas of the form $\exists y(y = a)$, where y is a variable of the same sort as a, are in $\bigwedge_j \varphi_j$, for every constant a occurring in the derivation. By the $\forall r$ rule (c and x are not free in the formulas left from the turn style)

$$\bigwedge_j \varphi_j, \exists x\psi(x) \Longrightarrow \forall x(\bigwedge_i(\forall x\theta_i \to \theta_i(x)) \to \neg\psi(x))$$

Since $\bigwedge_i(\forall x\theta_i \to \theta_i(x))$ is derivable,

$$\bigwedge_j \varphi_j, \exists x\psi(x) \Longrightarrow \forall x\neg\psi(x))$$

which contradicts the assumption that $\Sigma_n \cup \{\exists x \psi(x)\}$ is consistent.

As usual, we set the universe of the model to be $\cup_{i \in I} A_i$, where each A_i is the set of equivalence classes of constants of type i (c and c' are in the same equivalence class iff $(c = c') \in \Sigma$). Note that A_i might have non-empty intersection. The interpretation function is also easy to define:

$$\langle [c_1], \ldots, [c_n] \rangle \in \sigma(P) \Leftrightarrow P(c_1, \ldots, c_n) \in \Sigma$$

This works well because of the equality axiom $t = t' \wedge \varphi(t) \rightarrow \varphi(t')$.

F is empty, and all assignments from the product of A_i are possible (since all new constants are accessible, and Δ does not contain constants). The truth lemma ($\varphi \in \Sigma \Leftrightarrow \Sigma \models \varphi$) is easy to prove.

(2) Now consider the models where some functional dependency $A_i \rightarrow A_j$ holds. In fact that means that x_j can be eliminated and replaced by $f(x_i)$. Since we do not have functional symbols in the language, the same effect is obtained by adding the 'collapse' axiom $\exists x_j \varphi \rightarrow \forall x_j \varphi$.

To show that $\exists x_j \varphi \rightarrow \forall x_j \varphi$ is true in all models where a functional dependency $A_i \rightarrow A_j$ holds, suppose that $M, s \models \exists x_j \varphi(x_j)$, and $f_{ij} \in F$. Then the value of x_j cannot be changed while the value of x_i is fixed; the only element d such that $s_d^{x_j} \in S$ is $s(x_j)$. This means that $M, s \models \varphi(x_j)$, and all assignments which are accessible from s by changing the x_j coordinate, satisfy φ. But then $M, s \models \forall x_j \varphi(x_j)$. This proves that in all models satisfying the dependency $A_i \rightarrow A_j$, $\exists x_j \varphi \rightarrow \forall x_j \varphi$ is true.

For completeness, we can use the same construction as before. Suppose for some v_i $\exists v_i \varphi \rightarrow \forall v_i \varphi$ is in Δ for every φ. Then the A_i consists of just one element. Any functional dependency of the form $s_i = f(s_j)$ is trivially satisfied. $\square$

5 Conclusions

The paper provides some new results on quantifier logics with restricted sets of assignments, the study of which was initiated in [5]. The logic L_{fd} where the set of assignments is restricted by means of functional dependencies between variables, is shown to be undecidable, but enjoys a complete and sound sequent calculus formalisation. The undecidability result is somewhat disappointing, since many other logics with restricted sets of assignments are decidable, and so is modal logic with functional accessibility relation.

The motivation behind restricting the set of assignments in that way is the interest in databases satisfying some set of functional dependencies. The way of modelling functional dependencies between attributes chosen in this paper is, of course, not the only one possible. Other ways might be of interest as well, such as directly introducing functions in the language (although then the signature of the logical language does not correspond to the database schema) or restricting only extensions of relations so that they satisfy functional dependencies and leaving the set of assignments intact. It is not quite clear how to go about studying a logic defined is the latter way.

Acknowledgements I am grateful to Achim Jung and Michiel van Lambalgen for their comments on the draft of this paper.

On a more general level, I am grateful to Vladimir Aleksandrovich Smirnov who was the first one to teach me about sequent calculi and inductive proofs.

References

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

[2] N. Alechina. *Modal Quantifiers.* PhD thesis, ILLC, University of Amsterdam, 1995.

[3] H. Andréka, J. van Benthem, and I. Németi. Back and forth between modal logic and classical logic. *Bulletin of the Interest Group in Pure and Applied Logic*, 3:685 – 720, 1995.

[4] D. Scott. Identity and Existence in Intuitionistic Logic. In M. P. Fourman, C. J. Mulvey, and D. S. Scott, editors, *Applications of Sheaves*, Springer Lecture Notes in Mathematics 753, 660–696 1979.

[5] I. Németi. Decidability of weakened versions of first order logic and cylindric relativized set algebras. In L. Csirmaz, D. M. Gabbay, and M. de Rijke, editors, *Logic Colloquium'92.* CSLI, Stanford, 1992.