Translating a first-order modal language to relational algebra

Yeb Havinga*
Portavita B.V., The Netherlands

December 9, 2013

1 Introduction

Definition 1.1 (Kripke structure)**.** A *Kripke structure* is a tuple $\mathfrak{F}$ whose first component is a non-empty set $\mathcal{G}$ called the *universe* of $\mathfrak{F}$ and whose remaining components are binary relations on $\mathcal{G}$. We assume that every Kripke structure has at least one relation.

This paper is about Kripke structures that are

1. inside a relational database.
2. queried with a *modal* language.

At first the modal language that is used is introduced, followed by a definition of the database and relational algebra. Based on these definitions two things are described:

1. a mapping from components of the model structure to a relational database schema and instance.
2. a translation from queries in the modal language to relational algebra queries.

2 The modal language

2.1 Language

The modal language used is an adaptation of the language used in [FM99]. The most prominent difference is the absence of predicates.

*Work supervised by Balder ten Cate, Institute for Logic, Language and Computation, Universiteit van Amsterdam.

Definition 2.1 (Lexicon)**.** The lexicon consists of:

1. *basic symbols*: $\neg \quad \wedge \quad \vee \quad \rightarrow \quad \exists \ \forall \ (\ )$
2. *modal operators*: for every accessibility relation π the modal operators $\langle\pi\rangle$ and $[\pi]$
3. a collection of *constants symbols*. There are two kinds of

   constants: constants that denote individual objects and constants that denote individual concepts. There is a concept constant symbol *id*, which will be given a special meaning in definition 4.1.
4. a collection of *variable symbols*. Like constants, there are two kinds of variables. I'll use lowercase Latin letters x, y, z as object variables and lowercase Greek letters α, β, γ as concept variables.
5. the relation symbols $=$ and $\neq$.

Definition 2.2 (Term)**.** A *term* denotes an individual object or concept. The definition of *term* is as follows:

1. Constants and variables are *terms*. A term is an *object term* if it is an individual object variable or constant. Similarly for *concept terms*.
2. If t is a concept term, $\downarrow t$ is an object term. $\downarrow t$ is intended to designate the object denoted by t, in a particular state.
3. Nothing else is a term.

Definition 2.3 (Formula)**.** A *formula* expresses some fact about the (possibly virtual) reality. A formula without free variables is called a *sentence*. Sentences are the things of which we can say that they are true or false. The definition of formula is as follows:

1. If t_1 and t_2 are both object terms[1], then $t_1 = t_2$ and $t_1 \neq t_2$ are atomic formulas.
2. If φ is a formula, then $\neg\varphi$ is a formula.
3. If φ is a formula and π an accessibility relation, then $\langle\pi\rangle\varphi$ and $[\pi]\varphi$ are formulas.
4. If φ and ψ are formulas, so are $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$.
5. If φ is a formula and ϱ is a variable of either kind, then $\forall\varrho\ \varphi$ and $\exists\varrho\ \varphi$ are formulas.
6. If φ is a formula, ϱ is a variable of either kind, and t is a term of the same kind as ϱ, then $\langle\lambda\varrho.\varphi\rangle(t)$ is a formula.
7. Nothing else is a formula.

[1] The exclusion of concept terms is intentional.

2.2 Semantics

Definition 2.4 (Augmented Frame)**.** The frames we need to build first-order modal models with are enhanced versions of frames used for the semantics of propositional modal logic. Let Π be a set of accessibility relations. An *augmented frame* $\mathfrak{F}$ is a structure $\langle \mathcal{G}, \{R_\pi | \pi \in \Pi\}, \mathcal{D}_o, \mathcal{D}_c \rangle$ that consists of the following ingredients:

1. a non-empty set $\mathcal{G}$ of states. (worlds)
2. for every π in Π, a binary relation R_π on $\mathcal{G} \times \mathcal{G}$.
3. a non-empty set of objects $\mathcal{D}_o$, called the *domain of the frame*.
4. $\mathcal{D}_c$ is a non-empty set of functions from $\mathcal{G}$ to $\mathcal{D}_o$, called *individual concepts*.

The domain of an augmented frame is the set of things over which quantifiers can range, no matter at which state. $\mathcal{R}$ will be used as shorthand notation for $\{R_r | r \in \Pi\}$.

Definition 2.5 (Interpretation)**.** $\mathcal{I}$ is an *interpretation* in an augmented frame $(\mathcal{G}, \mathcal{R}, \mathcal{D}_o, \mathcal{D}_c)$ if $\mathcal{I}$ is a mapping that assigns:

1. to each individual object constant symbol some member of $\mathcal{D}_o$.
2. to each individual concept constant symbol some member of $\mathcal{D}_c$.

This interpretation gives rise to a *constant domain*, that is, a domain (of interpreted constants) that is invariable between states. It is assumed that individual objects and concepts have *unique names*. In other words, no two different constant symbols denote the same object. This allows us to use constant symbols to identify objects and concepts and vice versa.

Definition 2.6 (Model)**.** A *first-order modal model* is a pair $\mathfrak{M} = (\mathfrak{F}, \mathcal{I})$ where $\mathfrak{F}$ is an augmented frame and $\mathcal{I}$ is an interpretation in it.

Definition 2.7 (Assignment)**.** Let $\mathfrak{M} = (\mathfrak{F}, \mathcal{I})$ be a first-order modal model. A *assignment* v in the model $\mathfrak{M}$ is a mapping that assigns to each free individual object variable some member of $\mathcal{D}_o$ and to each free individual concept variable some member of $\mathcal{D}_c$.

Definition 2.8 (Term evaluation)**.** Let $\mathfrak{F} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}_o, \mathcal{D}_c \rangle$ be an augmented frame, $\mathfrak{M} = (\mathfrak{F}, \mathcal{I})$ be a model based on $\mathfrak{F}$ and v be an assignment in $\mathfrak{M}$. A mapping $(v * \mathcal{I})$ is defined, assigning a meaning to each term, at each possible state. Let $\Gamma \in \mathcal{G}$.

1. if ϱ is a variable, $(v * \mathcal{I})(\varrho, \Gamma) = v(\varrho)$.
2. if c is a constant symbol, $(v * \mathcal{I})(c, \Gamma) = \mathcal{I}(c)$.
3. if $\downarrow t$ is a relativized term, $(v * \mathcal{I})(\downarrow t, \Gamma) = (v * \mathcal{I})(t)(\Gamma)$.

To make reading easier, the following special notation is used. Let $\varrho_1, \dots, \varrho_k$ be variables of any type, and let $d_1, \dots, d_k$ be members of $\mathcal{D}_o \cup \mathcal{D}_c$, with $d_i \in \mathcal{D}_o$ if the variable ϱ_i is of object type, and $d_i \in \mathcal{D}_c$ if ϱ_i is of concept type. Then

$$\mathfrak{M}, \Gamma \Vdash_v \varphi[\varrho_i/d_1, \dots, \varrho_k/d_k]$$

abbreviates: $\mathfrak{M}, \Gamma \Vdash_{v'} \varphi$ where v' is the assignment that is like v on all variables except $\varrho_1, \dots \varrho_l$, and $v'(\varrho_1) = d_1, \dots v'(\varrho_k) = d_k$.

Definition 2.9 (Truth in a model)**.** Let $\mathfrak{F} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}_o, \mathcal{D}_c \rangle$ be an augmented frame, $\mathfrak{M} = (\mathfrak{F}, \mathcal{I})$ be a model based on $\mathfrak{F}$ and v be a assignment in $\mathfrak{M}$. We now inductively define the notion of a formula φ being *satisfied* (true) in $\mathfrak{M}$ at state Γ as follows:

1. $\mathfrak{M}, \Gamma \Vdash_v t_1 = t_2$ iff $(v * \mathcal{I})(t_1, \Gamma) = (v * \mathcal{I})(t_2, \Gamma)$.
2. $\mathfrak{M}, \Gamma \Vdash_v t_1 \neq t_2$ iff $(v * \mathcal{I})(t_1, \Gamma) \neq (v * \mathcal{I})(t_2, \Gamma)$.
3. $\mathfrak{M}, \Gamma \Vdash_v \neg\varphi$ iff $\mathfrak{M}, \Gamma \nVdash_v \varphi$.
4. $\mathfrak{M}, \Gamma \Vdash_v (\varphi \wedge \psi)$ iff $\mathfrak{M}, \Gamma \Vdash_v \varphi$ and $\mathfrak{M}, \Gamma \Vdash_v \psi$.
5. $\mathfrak{M}, \Gamma \Vdash_v (\varphi \vee \psi)$ iff $\mathfrak{M}, \Gamma \Vdash_v \varphi$ or $\mathfrak{M}, \Gamma \Vdash_v \psi$.
6. $\mathfrak{M}, \Gamma \Vdash_v \varphi \to \psi$ iff $\mathfrak{M}, \Gamma \nVdash_v \varphi$ or $\mathfrak{M}, \Gamma \Vdash_v \psi$.
7. $\mathfrak{M}, \Gamma \Vdash_v \forall x\, \varphi$ iff $\mathfrak{M}, \Gamma \Vdash_v \varphi[x/d]$ for all $d \in \mathcal{D}_o$.
8. $\mathfrak{M}, \Gamma \Vdash_v \forall \alpha\, \varphi$ iff $\mathfrak{M}, \Gamma \Vdash_v \varphi[\alpha/d]$ for all $d \in \mathcal{D}_c$.
9. $\mathfrak{M}, \Gamma \Vdash_v \exists x\, \varphi$ iff $\mathfrak{M}, \Gamma \Vdash_v \varphi[x/d]$ for some $d \in \mathcal{D}_o$.
10. $\mathfrak{M}, \Gamma \Vdash_v \exists \alpha\, \varphi$ iff $\mathfrak{M}, \Gamma \Vdash_v \varphi[\alpha/d]$ for some $d \in \mathcal{D}_c$.
11. $\mathfrak{M}, \Gamma \Vdash_v [\pi]\varphi$ iff for all $\Delta \in \mathcal{G}$, if $\pi(\Gamma, \Delta)$ then $\mathfrak{M}, \Delta \Vdash_v \varphi$.
12. $\mathfrak{M}, \Gamma \Vdash_v \langle\pi\rangle\varphi$ iff for some $\Delta \in \mathcal{G}$, if $\pi(\Gamma, \Delta)$ then $\mathfrak{M}, \Delta \Vdash_v \varphi$.
13. $\mathfrak{M}, \Gamma \Vdash_v \langle\lambda\varrho.\varphi\rangle(t)$ if $\mathfrak{M}, \Gamma \Vdash_v \varphi[\varrho/d]$ where $d = (v * \mathcal{I})(t, \Gamma)$.

Definition 2.10 (Modal query)**.** $\varphi(\varrho_1, \dots, \varrho_n)$ is a *modal query*, iff

1. φ is a wff of the modal language.
2. $\varrho_1, \dots, \varrho_n$ are distinct variables of either kind
3. $\varrho_1, \dots, \varrho_n$ are the only free variables in φ.

$\varrho_1, \dots, \varrho_n$ is called the *target list*.

3 Database and algebra

We adopt the unnamed conventional perspective of the relational modal, which is described in detail in chapter 3 of [AHV95]. The unnamed perspective is preferred over the named perspective, because it's easier to work with in the translation procedure and correspondence proof later in this section.

3.1 Database

Definition 3.1 (Database)**.** **dom** is a countably infinite set of individual objects. **relname** is a countably infinite set of relation names. A *relation scheme* is a relation name (symbol) R along with a positive integer called the *degree* (arity) of R. If R has degree n, the n attributes of R are identified by the numbers $1, \ldots, n$. A *relation instance* I, also associated with a degree n, is a finite set of n-tuples.

symbol	used for
t,u	tuple variables
a,b,c	constant symbols
R,S	relation names
I,J	relation instances
q	queries
R	database schema
I	database instance

3.2 Relational Algebra

Five primitive algebra operators form the *unnamed relational algebra*: projection, selection and cross product, union and set difference. The sixth operator, intersection, is added because it is the natural algebra counterpart of the conjunction logical connective.

Definition 3.2 (Selection)**.** Let j, k be positive integers and $c \in \mathbf{dom}$. Then $\sigma_{j=c}$ and $\sigma_{j=k}$ are selection operators. These operators applies to any relation instance I with degree$(I) \geq \max\{j, k\}$. The operator $\sigma_{j=c}$ is defined as follows:

$$\sigma_{j=c}(I) = \{t \in I | t(j) = c\}$$

producing output of degree(I).

Definition 3.3 (Projection)**.** The projection operator has the form $\pi_{j_1,\ldots,j_n}$ where $j_1, \ldots, j_n$ is a possibly empty sequence of positive integers, possibly with repeats. This operator takes as input any relation instance with degree $\geq \max\{j_1, \ldots, j_n\}$, and returns an instance with degree n, in particular,

$$\pi_{j_1,\ldots,j_n}(I) = \{\langle t(j_1), \ldots, t(j_n)\rangle | t \in I\}$$

.

Definition 3.4 (Cartesian (cross) product)**.** Let I and J be two relation instances., with arities n and m, respectively. The *cartesian product* returns a relation instance with a degree of $n + m$ and is defined as follows

$$I \times J = \{\langle t(1), \ldots, t(n), u(1), \ldots, u(m)\rangle | t \in I \text{ and } u \in J\}$$

The relation instance $\{\langle\rangle\}$ behaves as left and right identity:

$$I \times \{\langle\rangle\} = \{\langle\rangle\} \times I = I$$

Because cross-product is associative, it can be viewed as polyadic operator and written as $I_1 \times \ldots \times I_n$.

Definition 3.5 (Union-compatible)**.** Two relations are *union compatible* if they are of the same degree.

Definition 3.6 (Union)**.** By adding union to the algebra, it becomes possible to express disjunctive information in algebra expressions. Let I and J be two relation instances that are union-compatible. The union of I and J, noted $I \cup J$, is defined as follows:

$$I \cup J = \{t | t \in I \vee t \in J\}$$

Definition 3.7 (Difference)**.** Set difference adds negation to the algebra. Let I and J be two relation instances that are union-compatible. The set difference of I minus J, noted $I - J$, is defined as follows:

$$I - J = \{t | t \in I \wedge t \notin J\}$$

Definition 3.8 (Intersection)**.** Let I and J be two relation instances that are union-compatible. The intersection of I and J, noted $I \cap J$, is defined as follows:

$$I \cap J = \{t | t \in I \wedge t \in J\}$$

Definition 3.9 (Algebra query)**.** The *base algebra queries* are inductively defined as follows:

1. *Unary singleton constant* : If $c \in \mathbf{dom}$, then $\{\langle c\rangle\}$ is a query with *degree* 1.

2. *Input relation*: If R is a relation, the expression R is a query with degree equal to $degree(R)$.

The family of *algebra queries* is inductively defined as follows:

1. All base algebra queries are algebra queries.

2. *Selection*: Let $j, k \leq degree(q_1)$ and $c \in \mathbf{dom}$. If q_1 is a algebra query, then $\sigma_{j=c}(q_1)$ and $\sigma_{j=k}(q_1)$ are algebra queries with degrees equal to $degree(q_1)$,

3. *Projection*: If q_1 is a algebra query and each $j_1, \ldots, j_n \leq degree(q_1)$, then $\pi_{j_1,\ldots,j_n}(q_1)$ is a algebra query, with degree n.

4. *Cross product*: If q_1, q_2 are algebra queries with degrees n respectively m, then $q_1 \times q_2$ is a algebra query, with degree $n + m$.

5. *Union*: If q_1, q_2 are algebra queries that are union compatible; they are of the same degree, then $q_1 \cup q_2$ is a algebra query with degree $degree(q_1)$.

6. *Intersection*: If q_1, q_2 are algebra queries that are union compatible, then $q_1 \cap q_2$ is a algebra query with degree $degree(q_1)$.

7. *Difference*: If q_1, q_2 are algebra queries that are union compatible, then $q_1 - q_2$ is a algebra query with degree $degree(q_1)$.

4 Translation

Definition 4.1 (correspondence)**.** Let $\mathfrak{M} = \langle \mathfrak{F}, \mathcal{I} \rangle$ be a model and $\mathfrak{C}$ a bijective mapping that assigns to each concept in $\mathcal{D}_c$ a number between 1 and $|\mathcal{D}_c|$. Associated with $\mathfrak{M}$ are a unique database schema $\mathbf{R}_\mathfrak{M}$ and instance $\mathbf{I}_\mathfrak{M}$ for which the following condition holds:

1. $\mathbf{R}_\mathfrak{M} = \{Sta, Rel, Con, Obj\}$.

2. Con has degree 1 and $Con(\mathbf{I}_\mathfrak{M}) = \mathcal{D}_c$.

3. Obj has degree 1 and $Obj(\mathbf{I}_\mathfrak{M}) = \mathcal{D}_o$.

4. The domain **dom** of the database is $\mathcal{D}_c \cup \mathcal{D}_o \cup \Pi$.

5. The modal language and the database share the same collection of object constant symbols.

6. Unique names are assumed, in particular, the object denoted by a database object constant symbol c is equivalent to it's interpretation in the modal language $\mathcal{I}(c)$.

7. degree$(Sta) = |\mathcal{D}_c|$

8. Γ is a state in $\mathcal{G}$ iff there is a tuple $\langle a_1, \ldots, a_n \rangle$ in $Sta(\mathbf{I}_\mathfrak{M})$ such that for every $i \in \{1, \ldots, n\}$ holds $\mathcal{I}(a_i) = (v * \mathcal{I})(\downarrow c)(\Gamma)$, where $\mathfrak{C}(c) = i$ and $n = $ degree(Sta).

9. there is a concept named id in $\mathcal{D}_c$ such that $Sta : id \to c_1 \ldots c_k$ where $\bigcup_{i=1\ldots k} c_i = \mathcal{D}_c - id$. $\mathfrak{C}(id) = 1$.

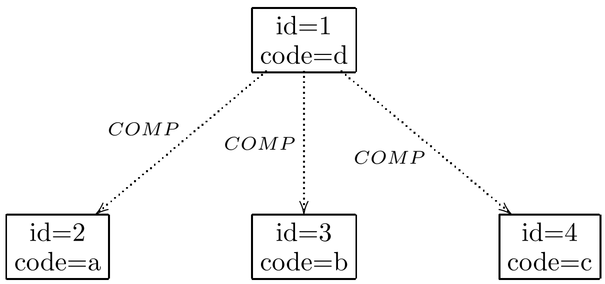
10. $Dom(typeCode) = \Pi$.

11. Rel has degree 3.

12. for every $\pi \in \Pi$ holds:
    $\pi(\Gamma, \Delta)$ iff there is a tuple $\langle a, b, c \rangle$ in $Rel(\mathbf{I}_\mathfrak{M})$ such that $\mathcal{I}(a) = (v * \mathcal{I})(\downarrow id)(\Gamma)$ and $\mathcal{I}(b) = (v * \mathcal{I})(\downarrow id)(\Delta)$ and $\mathcal{I}(c) = \pi$.

Example 4.1. This example shows a model and it's corresponding database instance:

Model:



Database:

Sta	id	code
	1	d
	2	a
	3	b
	4	c

Rel	source	target	typeCode
	1	2	COMP
	1	3	COMP
	1	4	COMP

Con	1
	id
	code

Obj	1
	1
	2
	3
	4
	a
	b
	c
	d

Definition 4.2 (Formula translation)**.** The following translation takes as input a query $\varphi(\varrho_1, \ldots, \varrho_n)$ in the modal language and results in a relational algebra expression. The translation consists of a set of syntactic translation rules. The basic idea is that each atomic subformula, with free variables $\varrho_1, \ldots, \varrho_n$ is translated to a query on *Sta*, *Obj* and *Con* that has the following structure:

$FT(\varphi(\varrho_1, \ldots, \varrho_n)$	1	$\ldots$	n	id
	$v(\varrho_1)$	$\ldots$	$v(\varrho_n)$	Γ

Example 4.2 (no variable query image)**.** The query image of a translated formula with no variables looks like this:

$FT(\downarrow code = b)$	id
	3

Example 4.3 (one variable query image)**.** The query image of a translated formula with one variables looks like this

$FT((\downarrow id = 3 \wedge code = \varrho_1)(\varrho_1))$	ϱ_1	id
	b	3

Conjunctions and disjunctions result in intersections and unions of queries. Negation of a query is translated to set difference on Sta. The translation of the existential quantifier is done by translating to a query with the quantified variable added to it's target list, which is later removed by projection of the original target list. The universal quantifier is translated by translating into the division of the translation of the remaining subformula of the query, by the concept domain Con or object domain Obj. The diamond modal operator is translated to a query on Rel. Since the translation doesn't require a specific normal form, we can use the dual of the diamond operator to translate the box operator. Lambda abstraction is translated using an extra query that captures the designation of the relativized term.

1. Term translations result in attribute index numbers or constants.

   (a) $TT(c) =$ **'**c**'**, if c an object constant.

   (b) $TT(\varrho_k) = k$, if ϱ_k is a variable of either kind.

   (c) $TT(\downarrow t) = n + \mathfrak{C}(t)$, if $\downarrow t$ is a relativized term. n is the number of variables of the subformula in the current scope.

2. Variables result in domain relations.

   (a) $VT(\varrho_1, \ldots, \varrho_n) = \{\langle\rangle\}$, if $\varrho_1, \ldots, \varrho_n$ is empty.

   (b) $VT(\varrho_1, \ldots, \varrho_n) = D_1 \times \ldots \times D_n$, otherwise, where D_i is the relation Con, if ϱ_i is a concept variable, and Obj if ϱ_i is an object variable, $1 \leq i \leq n$.

3. (sub)formula translations are translated to algebra queries.

   (a) $FT(\ (t_1 = t_2)(\varrho_1, \ldots, \varrho_n)\ ) = \pi_{1,\ldots,n+1}\sigma_{(TT(t_1)=TT(t_2))}(VT(\varrho_1, \ldots, \varrho_n) \times Sta)$

   (b) $FT(\ (t_1 \neq t_2)(\varrho_1, \ldots, \varrho_n)\ ) = \pi_{1,\ldots,n+1}\sigma_{(TT(t_1)\neq TT(t_2))}(VT(\varrho_1, \ldots, \varrho_n) \times Sta)$

(c) $FT(\ \neg\varphi(\varrho_1,\ldots,\varrho_n)\ ) = (VT(\varrho_1,\ldots,\varrho_n)\times\pi_1(Sta)) - FT(\varphi(\varrho_1,\ldots,\varrho_n))$

(d) $FT(\ \langle\pi\rangle\varphi(\varrho_1,\ldots,\varrho_n)\ ) = \pi_{1,\ldots,n,n+2}\sigma_{(n+4=\pi\wedge n+1=n+3)}(FT(\varphi(\varrho_1,\ldots,\varrho_n))\times Rel)$

(e) $FT(\ [\pi]\varphi(\varrho_1,\ldots,\varrho_n)\ ) = FT(\neg\langle\pi\rangle\neg\varphi(\varrho_1,\ldots,\varrho_n)\ )$

(f) $FT(\ (\varphi\vee\psi)(\varrho_1,\ldots,\varrho_n)\ ) = FT(\varphi(\varrho_1,\ldots,\varrho_n)) \cup FT(\psi(\varrho_1,\ldots,\varrho_n))$

(g) $FT(\ (\varphi\wedge\psi)(\varrho_1,\ldots,\varrho_n)\ ) = FT(\varphi(\varrho_1,\ldots,\varrho_n)) \cap FT(\psi(\varrho_1,\ldots,\varrho_n))$

(h) $FT(\ \exists\varrho\ \varphi(\varrho_1,\ldots,\varrho_n)\ ) = \pi_{2,\ldots,n+2}FT(\varrho,\varrho_1,\ldots,\varrho_n)$

(i) $FT(\ \forall\varrho\ \varphi(\varrho_1,\ldots,\varrho_n)\ ) = \pi_{2,\ldots,n+2}U - \pi_{2,\ldots,n+2}((VT(\varrho)\times\pi_{2,\ldots,n+2}U) - U)$, where $U = FT(\varrho,\varrho_1,\ldots,\varrho_n)$

(j) $FT(\langle\lambda\varrho.\varphi\rangle(t)(\varrho_1,\ldots,\varrho_n)) = \pi_{2,\ldots,n+2}\sigma_{(1=n+3\wedge n+2=n+4)}(FT(\varphi(\varrho,\varrho_1,\ldots,\varrho_n))\times \pi_{TT(t),1}Sta\ )$

4.1 Examples

Example 4.4 (Atomic formula, no variables)**.** Here follows the translation of the variable free query ($\downarrow$ *code* = 'b'). In this example, the translation of $\times VT(\varrho_1,\ldots,\varrho_n)$ with an empty $\varrho_1,\ldots,\varrho_n$ is given explicitly. In the remaining examples, I will omit this explicit translation of empty variable lists and directly write S instead of $S\times VT()$.

$$\begin{aligned} FT(\downarrow code = b) &\Rightarrow \pi_{1,\ldots,1}\sigma_{(TT(\downarrow code)=TT(b))}(VT()\times Sta)\\ &\Rightarrow \pi_1\sigma_{2=\texttt{'b'}}(\{\langle\rangle\}\times Sta)\\ &\Rightarrow \pi_1\sigma_{2=\texttt{'b'}}Sta \end{aligned}$$

Example 4.5 (Diamond operator, no variables)**.** Here follows the translation of the variable free query $\langle COMP\rangle(\downarrow code = \text{'b'})$.

$$\begin{aligned} FT(\langle COMP\rangle(\downarrow code = b)) &\Rightarrow \pi_{1,\ldots,0,2}\sigma_{(4=\texttt{'COMP'}\wedge 1=3)}(FT(\downarrow code = b)\times Rel)\\ &\Rightarrow \pi_2\sigma_{(4=\texttt{'COMP'}\wedge 1=3)}(\pi_1\sigma_{2=\texttt{'b'}}(Sta)\times Rel) \end{aligned}$$

Example 4.6 (Box operator, no variables)**.**

$$\begin{aligned} &FT([COMP](\downarrow code = b))\\ &\Rightarrow FT(\neg\langle COMP\rangle\neg(\downarrow code = b))\\ &\Rightarrow (VT(\varrho_1,\ldots,\varrho_n)\times\pi_1 Sta) - FT(\langle COMP\rangle\neg(\downarrow code = b))\\ &\Rightarrow \pi_1 Sta - (\pi_2\sigma_{(4=\texttt{'COMP'}\wedge 1=3}(Rel\times FT(\neg(\downarrow code = b))))\\ &\Rightarrow \pi_1 Sta - (\pi_2\sigma_{(4=\texttt{'COMP'}\wedge 1=3)}(Rel\times((VT(\varrho_1,\ldots,\varrho_n)\times\pi_1 Sta) - FT((\downarrow code = b)(\varrho_1,\ldots,\varrho_n)))))\\ &\Rightarrow \pi_1 Sta - (\pi_2\sigma_{(4=\texttt{'COMP'}\wedge 1=3)}(Rel\times(\pi_1 Sta - \pi_1\sigma_{2=\texttt{'b'}}Sta))) \end{aligned}$$

Example 4.7 (Predicate abstraction, no variables)**.** Here follows the translation of the variable free query $\langle\lambda y.\langle COMP\rangle(\downarrow code = y)\rangle(\downarrow code)$. Note that the first translation step introduces a variable.

$$\begin{aligned} &FT(\langle\lambda y.\langle COMP\rangle(\downarrow code = \varrho)\rangle(\downarrow code))\\ &\Rightarrow \pi_{2,\ldots,2}\sigma_{(1=3\wedge 2=4)}(FT(\langle COMP\rangle(\downarrow code = \varrho)(\varrho))\times\pi_{TT(code)-0,1}Sta)\\ &\Rightarrow \pi_2\sigma_{(1=3\wedge 2=4)}(\pi_{1,\ldots,1,2}\sigma_{5=\texttt{'COMP'}\wedge 2=5}(FT((\downarrow code = \varrho)(\varrho))\times Rel)\times\pi_{2,1}Sta)\\ &\Rightarrow \pi_2\sigma_{(1=3\wedge 2=4)}(\pi_{1,2}\sigma_{5=\texttt{'COMP'}\wedge 2=5}(\pi_{1,2}\sigma_{TT(\downarrow code)=TT(\varrho)}(VT(\varrho)\times Sta)\times Rel)\times\pi_{2,1}Sta)\\ &\Rightarrow \pi_2\sigma_{(1=3\wedge 2=4)}(\pi_{1,2}\sigma_{5=\texttt{'COMP'}\wedge 2=5}(\pi_{1,2}\sigma_{3=1}(Obj\times Sta)\times Rel)\times\pi_{2,1}Sta) \end{aligned}$$

4.2 Proof of correspondence

Lemma 4.1. *Let $\mathbf{I}_{\mathfrak{M}}$ be a database instance that is associated with a model $\mathfrak{M}$. Then $\bigcup_{i=1,\ldots,n} \pi_i Sta(\mathbf{I}_{\mathfrak{M}}) \subseteq Obj(\mathbf{I}_{\mathfrak{M}})$.*

If this was not the case, it would be a violation of definition 4.1, item 4.

Lemma 4.2. *Let $\mathbf{I}_{\mathfrak{M}}$ be a database instance that is associated with a model $\mathfrak{M}$. Then $Sta(\mathbf{I}_{\mathfrak{M}})$ is not empty.*

This property follows straight from definition 2.4, item 1 and definition 4.1, item 8.

Lemma 4.3. *Fix a model state pair $\mathfrak{M}, \Gamma$. Then for any two arbitrary terms t_1, t_2, assignment v and and object constant i, where i identifies the state in Sta such that $\mathcal{I}(i) = (v * \mathcal{I})(\downarrow id, \Gamma)$, the following holds:*

$(v*\mathcal{I})(t_1, \Gamma) = (v*\mathcal{I})(t_2, \Gamma)$ *iff* $\langle v(\varrho_1), \ldots, v(\varrho_n), i\rangle \in FT((t_1 = t_2)(\varrho_1, \ldots, \varrho_n))(\mathbf{I}_{\mathfrak{M}})$

where $\varrho_1, \ldots, \varrho_n$ is the list of variables in t_1, t_2. [2]

Proof. Let t_1, t_2 be terms, v an assignment and u the tuple $\langle v(\varrho_1), \ldots, v(\varrho_n), i\rangle$ such that the following holds:

$$u \in FT((t_1 = t_2)(\varrho_1, \ldots, \varrho_n))(\mathbf{I}_{\mathfrak{M}})$$

which by translation step 3(a) of definition 4.2 is equal to.

$$u \in \pi_{1,\ldots,n+1}\sigma_{(TT(t_1)=TT(t_2))}(VT(\varrho_1, \ldots, \varrho_n) \times Sta)(\mathbf{I}_{\mathfrak{M}})$$

Thus the following equivalence is to be proved:

$(v*\mathcal{I})(t_1, \Gamma) = (v*\mathcal{I})(t_2, \Gamma)$ iff $u \in \pi_{1,\ldots,n+1}\sigma_{(TT(t_1)=TT(t_2))}(VT(\varrho_1, \ldots, \varrho_n) \times Sta)(\mathbf{I}_{\mathfrak{M}})$.

- t_1 and t_2 are both constants. Since there are no variables, u is the tuple $\langle i\rangle$ and $\pi_{1,\ldots,n+1}\sigma_{(TT(t_1)=TT(t_2))}(VT(\varrho_1, \ldots, \varrho_n) \times Sta)(\mathbf{I}_{\mathfrak{M}})$ is equivalent with $\pi_1\sigma_{TT(t_1)=TT(t_2)}Sta(\mathbf{I}_{\mathfrak{M}})$.

  $\Rightarrow$ By definition 2.8 item 2, $(v*\mathcal{I})(t_1, \Gamma) = (v*\mathcal{I})(t_2, \Gamma)$ iff $\mathcal{I}(t_1) = \mathcal{I}(t_2)$. By the unique names assumption, t_1 and t_2 are the same constant. By definition 4.1, item 5, this constant exists in the database. Hence $TT(t_1) = TT(t_2)$ and therefore $\sigma_{TT(t_1)=TT(t_2)}Sta(\mathbf{I}_{\mathfrak{M}}) = Sta(\mathbf{I}_{\mathfrak{M}})$. By definition 4.1 item 9, $u \in \pi_1\sigma_{TT(t_1)=TT(t_2)}Sta(\mathbf{I}_{\mathfrak{M}})$.

  $\Leftarrow$ Suppose that t_1 and t_2 are different constants. Because of the unique names assumption, $\pi_1\sigma_{TT(t_1)=TT(t_2)}Sta(\mathbf{I}_{\mathfrak{M}})$ is empty. But $u \in \pi_1\sigma_{TT(t_1)=TT(t_2)}Sta(\mathbf{I}_{\mathfrak{M}})$ and $Sta(\mathbf{I}_{\mathfrak{M}})$ is not empty (lemma 4.2). Therefore t_1 and t_2 are the same constant. By definition 4.1, item 5, this constant exists in the modal language. Hence $(v*\mathcal{I})(t_1, \Gamma) = (v*\mathcal{I})(t_2, \Gamma)$.

[2] The object constant symbol i is used on many occasions where $\mathcal{I}(i)$ is more appropriate. From the context it should be clear whether the object itself, a number or word that identifies a state, or it's (unique) symbol is meant.

- t_1 is a constant, t_2 is a relativized concept. For the sake of readability, let t_1 be the constant a and t_2 be the relativized concept $\downarrow c$, with $\mathfrak{C}(c) = 2$. Let $u \in \pi_{1,\ldots,n+1}\sigma_{(TT(t_1)=TT(t_2))}(VT(\varrho_1, \ldots, \varrho_n) \times Sta)(\mathbf{I}_{\mathfrak{M}})$. Because there are no variables, this is equivalent with $u \in \pi_1\sigma_{\text{'a'}=2}(Sta)(\mathbf{I}_{\mathfrak{M}})$, which by definition 3.2 holds iff $\langle i \rangle \in \{t[1] | t \in Sta(\mathbf{I}_{\mathfrak{M}}) \wedge t[2] = a\}$, which holds iff there exists a tuple t in $Sta(\mathbf{I}_{\mathfrak{M}})$ such that $t[1] = i$ and $t[2] = a$. Because $\mathcal{I}(i) = (v * \mathcal{I})(\downarrow id, \Gamma)$ and by definition 4.1 item 8 this holds iff $\mathcal{I}(a) = (v * \mathcal{I})(\downarrow c, \Gamma)$. By definition 2.5, this holds iff $(v * \mathcal{I})(a, \Gamma) = (v * \mathcal{I})(\downarrow c, \Gamma)$.
- t_1 is a variable, t_2 is a relativized concept. For the sake of readability, let t_1 be the variable ϱ and t_2 be the relativized concept $\downarrow c$, with $\mathfrak{C}(c) = k$. Let v be any assignment such that $\langle v(\varrho), i \rangle \in FT((\varrho =\downarrow c)(\varrho))$. This is translated to $\langle v(\varrho), i \rangle \in \pi_{1,\ldots,n+1}\sigma_{(TT(\varrho)=TT(\downarrow c))}(VT(\varrho) \times Sta)(\mathbf{I}_{\mathfrak{M}})$, which is further translated and simplified to the equivalent $\langle v(\varrho), i \rangle \in \pi_{1,2}\sigma_{(1=k+1)}(Obj \times Sta)(\mathbf{I}_{\mathfrak{M}})$, which by definition 3.2 is equal to $\langle v(\varrho), i \rangle \in \{t \in \pi_{1,k+1}(Obj \times Sta)(\mathbf{I}_{\mathfrak{M}}) | t(1) = t(k+1)\}$. Because lemma 4.1 holds on database instance $\mathbf{I}_{\mathfrak{M}}$, the set $\{t | t \in \pi_{1,k+1}(Obj \times Sta)(\mathbf{I}_{\mathfrak{M}}) \wedge t(1) = t(k+1)\}$ is equal to the set $\{t | t \in \pi_{k,1} Sta(\mathbf{I}_{\mathfrak{M}}\}$. Hence $\langle v(\varrho), i \rangle$ is in $\pi_{k,1} Sta(\mathbf{I}_{\mathfrak{M}})$ iff v assigns to ϱ the object denoted by the attribute with index k, which corresponds to relativized concept $\downarrow c$. By definition 4.1 item 8, this holds iff $v(\varrho) = (v * \mathcal{I})(\downarrow c, \Gamma)$, which, in other words, is equal to $(v * \mathcal{I})(t_1, \Gamma) = (v * \mathcal{I})(t_2, \Gamma)$.
- the remaining combinations follow from commutativity and transitivity of $=$.

$\square$

Proposition 4.4 (Correspondence)**.** *Fix a model state pair $\mathfrak{M}, \Gamma$ and object constant i, such that $\mathcal{I}(i) = (v * \mathcal{I})(\downarrow id, \Gamma)$ Then the following holds*

$$\mathfrak{M}, \Gamma \Vdash_v \varphi(\varrho_1, \ldots, \varrho_n) \text{ iff } \langle v(\varrho_1), \ldots, v(\varrho_n), i \rangle \in FT(\varphi(\varrho_1, \ldots, \varrho_n))(\mathbf{I}_{\mathfrak{M}})$$

Proof. By induction on the structure of φ.

- Base case: φ is $t_1 = t_2$, where t_1, t_2 are object terms. This is lemma 4.3.
- Case $\neg\varphi(\varrho_1, \ldots, \varrho_n)$: Assume that $\mathfrak{M}, \Gamma \Vdash_v \neg\varphi(\varrho_1, \ldots, \varrho_n)$. This holds
  iff $\mathfrak{M}, \Gamma \nVdash_v \varphi$ (truth definition)
  iff $\langle v(\varrho_1), \ldots, v(\varrho_n), i \rangle \notin FT(\varphi(\varrho_1, \ldots, \varrho_n))(\mathbf{I}_{\mathfrak{M}})$ (inductive hypothesis)
  iff $\langle v(\varrho_1), \ldots, v(\varrho_n), i \rangle \in (VT(\varrho) \times \pi_1 Sta) - FT(\varphi(\varrho_1, \ldots, \varrho_n))(\mathbf{I}_{\mathfrak{M}})$ (def 3.7)
  iff $\langle v(\varrho_1), \ldots, v(\varrho_n), i \rangle \in FT(\neg\varphi(\varrho_1, \ldots, \varrho_n))(\mathbf{I}_{\mathfrak{M}})$ (def 4.2 item 3(c)).
- Case $(\varphi \wedge \psi)(\varrho_1, \ldots, \varrho_n)$: Assume that $\mathfrak{M}, \Gamma \Vdash_v \varphi \wedge \psi(\varrho_1, \ldots, \varrho_n)$. This holds
  iff $\mathfrak{M}, \Gamma \Vdash_v \varphi$ and $\mathfrak{M}, \Gamma \Vdash_v \psi$ (truth definition)
  iff $\langle v(\varrho_1), \ldots, v(\varrho_n), i \rangle \in FT(\varphi(\varrho_1, \ldots, \varrho_n))(\mathbf{I}_{\mathfrak{M}})$ (IH)
  and $\langle v(\varrho_1), \ldots, v(\varrho_n), i \rangle \in FT(\psi(\varrho_1, \ldots, \varrho_n))(\mathbf{I}_{\mathfrak{M}})$ (IH)
  iff $\langle v(\varrho_1), \ldots, v(\varrho_n), i \rangle \in FT(\varphi(\varrho_1, \ldots, \varrho_n))(\mathbf{I}_{\mathfrak{M}})$
  $\cap\ FT(\psi(\varrho_1, \ldots, \varrho_n))(\mathbf{I}_{\mathfrak{M}})$ (def 3.8)
  iff $\langle v(\varrho_1), \ldots, v(\varrho_n), i \rangle \in FT((\varphi \wedge \psi)(\varrho_1, \ldots, \varrho_n))(\mathbf{I}_{\mathfrak{M}})$ (def 4.2 item 3(g)).

- Case $(\varphi \vee \psi)(\varrho_1, \ldots, \varrho_n)$: similar to the conjunction case.

- Case $\langle\pi\rangle\varphi(\varrho_1, \ldots, \varrho_n)$: Let id_Γ, id_Δ be shorthand notations for $(v * \mathcal{I})(id, \Gamma)$, $(v * \mathcal{I})(id, \Delta)$ respectively. It is easy to see that the object constant i is equal to id_Γ.

  $\Rightarrow$: Assume that $\mathfrak{M}, \Gamma \Vdash_v \langle\pi\rangle\varphi(\varrho_1, \ldots, \varrho_n)$. By the truth definition, there exists a Δ such that $\pi(\Gamma, \Delta)$ and $\mathfrak{M}, \Delta \Vdash_v \varphi(\varrho_1, \ldots, \varrho_n)$. By the IH, $\langle v(\varrho_1), \ldots, v(\varrho_n), id_\Delta\rangle \in FT(\varphi(\varrho_1, \ldots, \varrho_n))(\mathbf{I}_\mathfrak{M})$. Since $\pi(\Gamma, \Delta)$ and because of definition 4.1 item 12, the tuple $\langle id_\Gamma, id_\Delta, \pi\rangle \in Rel(\mathbf{I}_\mathfrak{M})$. The crossproduct $FT(\varphi(\varrho_1, \ldots, \varrho_n)) \times Rel$ contains the following attributes: $1, \ldots, n$ are $\varrho_1, \ldots, \varrho_n$. $n + 1$ the id's of the states in which the subformula φ is true. At index $n + 2$ the Rel relation appears in the cross product: $n + 2$ holds the source state, $n + 3$ the target state and $n + 4$ the typeCode. Hence $\langle v(\varrho_1), \ldots, v(\varrho_n), id_\Gamma\rangle \in \pi_{1,\ldots,n,n+2}\sigma_{(n+4=\pi \wedge n+1=n+3)}(FT(\varphi(\varrho_1, \ldots, \varrho_n)) \times Rel)(\mathbf{I}_\mathfrak{M})$. Since i is equal to id_Γ, this means that, $\langle v(\varrho_1), \ldots, v(\varrho_n), i\rangle \in FT(\langle\pi\rangle\varphi(\varrho_1, \ldots, \varrho_n))(\mathbf{I}_\mathfrak{M})$.

  $\Leftarrow$: Assume that $\langle v(\varrho_1), \ldots, v(\varrho_n), i\rangle \in FT(\langle\pi\rangle\varphi(\varrho_1, \ldots, \varrho_n))(\mathbf{I}_\mathfrak{M})$. Applying translation step 3(d) of definition 4.2 gives $\langle v(\varrho_1), \ldots, v(\varrho_n), i\rangle \in \pi_{1,\ldots,n,n+2}\sigma_{(n+4=\pi \wedge n+1=n+3)}(FT(\varphi(\varrho_1, \ldots, \varrho_n)) \times Rel)(\mathbf{I}_\mathfrak{M})$. This means there exist tuples t, u in respectively $FT(\varphi(\varrho_1, \ldots, \varrho_n))(\mathbf{I}_\mathfrak{M})$ and $Rel(\mathbf{I}_\mathfrak{M})$, such that $t[n + 1] = u[2]$ and $u[3] = \pi$ and $u[1] = i$. Let $u[2] = id_\Delta$. Since $i = id_\Delta$ and by definition 4.1 item 12, $\pi(\Gamma, \Delta)$. Since $u[2] = t[n + 1]$, also holds $\langle v(\varrho_1), \ldots, v(\varrho_n), id_\Delta\rangle \in FT(\varphi(\varrho_1, \ldots, \varrho_n))$. Hence, by the IH, $\mathfrak{M}, \Delta \Vdash_v \varphi(\varrho_1, \ldots, \varrho_n)$. By the truth definition, $\mathfrak{M}, \Gamma \Vdash_v \langle\pi\rangle\varphi(\varrho_1, \ldots, \varrho_n)$.

- Case $\exists\varrho\, \varphi(\varrho_1, \ldots, \varrho_n)$: Let U be the query $FT(\varphi(\varrho, \varrho_1, \ldots, \varrho_n))$. The inductive hypothesis states $\mathfrak{M}, \Gamma \Vdash_v \varphi(\varrho, \varrho_1, \ldots, \varrho_n)$ iff $\langle v(\varrho), v(\varrho_1), \ldots, v(\varrho_n), i\rangle \in U(\mathbf{I}_\mathfrak{M})$.

  $\Rightarrow$: Assume that $\mathfrak{M}, \Gamma \Vdash_v \exists\varrho\, \varphi(\varrho_1, \ldots, \varrho_n)$.

  By definition 2.9 of $\exists$, $\mathfrak{M}, \Gamma \Vdash_v \varphi[\varrho/d](\varrho_1, \ldots, \varrho_n)$ for some $d \in \mathcal{D}_o$[3]. By the IH, for some $d \in VT(\varrho)$, $\langle d, v(\varrho_1), \ldots, v(\varrho_n), i\rangle \in U(\mathbf{I}_\mathfrak{M})$. Hence $\langle v(\varrho_1), \ldots, v(\varrho_n), i\rangle \in \pi_{2,\ldots,n+2}FT(\varphi(\varrho, \varrho_1, \ldots, \varrho_n))(\mathbf{I}_\mathfrak{M})$.

  $\Leftarrow$: Assume that $\langle v(\varrho_1), \ldots, v(\varrho_n), i\rangle \in \pi_{2,\ldots,n+2}FT(\varphi(\varrho, \varrho_1, \ldots, \varrho_n))(\mathbf{I}_\mathfrak{M})$. For the sake of contradition, suppose that $\mathfrak{M}, \Gamma \Vdash_v \nexists\varrho\, \varphi(\varrho_1, \ldots, \varrho_n)$. By definition 2.9, $\mathfrak{M}, \Gamma \nVdash_v \varphi[\varrho/d](\varrho_1, \ldots, \varrho_n)$ for some $d \in \mathcal{D}_o$. By the IH and dual, for no $d \in \mathcal{D}_o$, $\langle d, v(\varrho_1), \ldots, v(\varrho_n), i\rangle \in U(\mathbf{I}_\mathfrak{M})$. Hence for all $d \in \mathcal{D}_o$, $\langle d, v(\varrho_1), \ldots, v(\varrho_n), i\rangle \notin U(\mathbf{I}_\mathfrak{M})$ and hence $\langle v(\varrho_1), \ldots, v(\varrho_n), i\rangle \notin \pi_{2,\ldots,n+2}FT(\varphi(\varrho, \varrho_1, \ldots, \varrho_n))(\mathbf{I}_\mathfrak{M})$. But this is a contradiction, so $\mathfrak{M}, \Gamma \Vdash_v \exists\varrho\, \varphi(\varrho_1, \ldots, \varrho_n)$.

- Case $\forall\varrho\, \varphi(\varrho_1, \ldots, \varrho_n)$: Let U be the query $FT(\varphi(\varrho, \varrho_1, \ldots, \varrho_n))$ and let A be the query $VT(\varrho) \times \pi_{2,\ldots,n+2}U$. The inductive hypothesis states $\mathfrak{M}, \Gamma \Vdash_v \varphi(\varrho, \varrho_1, \ldots, \varrho_n)$ iff $\langle v(\varrho), v(\varrho_1), \ldots, v(\varrho_n), i\rangle \in U(\mathbf{I}_\mathfrak{M})$.

[3]Replace with $\mathcal{D}_c$ if ϱ is a concept variable.

$\Rightarrow$: Assume that $\mathfrak{M}, \Gamma \Vdash_v \forall \varrho\, \varphi(\varrho_1, \ldots, \varrho_n)$.

By definition 2.9 of $\forall$, $\mathfrak{M}, \Gamma \Vdash_v \varphi[\varrho/d](\varrho_1, \ldots, \varrho_n)$ for all $d \in \mathcal{D}_o$[4]. Hence, by the IH, for all $d \in \mathcal{D}_o$, $\langle d, v(\varrho_1), \ldots, v(\varrho_n), i\rangle \in U(\mathbf{I}_{\mathfrak{M}})$ (*). It is now easy to see that $A(\mathbf{I}_{\mathfrak{M}}) = U(\mathbf{I}_{\mathfrak{M}})$. Hence $(A - U)(\mathbf{I}_{\mathfrak{M}}) = \emptyset$. Hence $\pi_{2,\ldots,n+2}U - \pi_{2,\ldots,n+2}(A) - U)(\mathbf{I}_{\mathfrak{M}}) = \pi_{2,\ldots,n+2}U(\mathbf{I}_{\mathfrak{M}})$. Since (*), $\langle v(\varrho_1), \ldots, v(\varrho_n), i\rangle \in \pi_{2,\ldots,n+2}U - \pi_{2,\ldots,n+2}((VT(\varrho) \times \pi_{2,\ldots,n+2}U) - U)(\mathbf{I}_{\mathfrak{M}})$ By definition 4.2, $\langle v(\varrho_1), \ldots, v(\varrho_n), i\rangle \in FT(\forall \varrho\, \varphi(\varrho_1, \ldots, \varrho_n))(\mathbf{I}_{\mathfrak{M}})$

$\Leftarrow$: Assume that $\langle v(\varrho_1), \ldots, v(\varrho_n), i\rangle \in FT(\forall \varrho\, \varphi(\varrho_1, \ldots, \varrho_n))(\mathbf{I}_{\mathfrak{M}})$. By definition 4.2, $\langle v(\varrho_1), \ldots, v(\varrho_n), i\rangle \in \pi_{2,\ldots,n+2}U - \pi_{2,\ldots,n+2}((VT(\varrho) \times \pi_{2,\ldots,n+2}U) - U)(\mathbf{I}_{\mathfrak{M}})$. For the sake of contradiction, suppose that $\mathfrak{M}, \Gamma \nVdash_v \forall \varrho\; \varphi(\varrho_1, \ldots, \varrho_n)$. By definition 2.9, there exists a $d \in \mathcal{D}_o$ such that $v(\varrho) = d$ and $\mathfrak{M}, \Gamma \nVdash_v \varphi(\varrho, \varrho_1, \ldots, \varrho_n)$. By the IH, $\langle v(\varrho), v(\varrho_1), \ldots, v(\varrho_n), i\rangle \notin U(\mathbf{I}_{\mathfrak{M}})$. Hence $\langle v(\varrho), v(\varrho_1), \ldots, v(\varrho_n), i\rangle \in (A - U)(\mathbf{I}_{\mathfrak{M}})$ and hence $\langle v(\varrho_1), \ldots, v(\varrho_n), i\rangle \in \pi_{2,\ldots,n+2}(A - U)(\mathbf{I}_{\mathfrak{M}})$. and hence $\langle v(\varrho_1), \ldots, v(\varrho_n), i\rangle \notin \pi_{2,\ldots,n+2}U - \pi_{2,\ldots,n+2}((VT(\varrho) \times \pi_{2,\ldots,n+2}U) - U)(\mathbf{I}_{\mathfrak{M}})$ which is a contradiction. Therefore $\mathfrak{M}, \Gamma \Vdash_v \forall \varrho\, \varphi(\varrho_1, \ldots, \varrho_n)$.

- Case $\langle \lambda \varrho.\varphi\rangle(t)(\varrho_1, \ldots, \varrho_n)$:

  Assume that $\mathfrak{M}, \Gamma \Vdash_v \langle \lambda \varrho.\varphi\rangle(t)(\varrho_1, \ldots, \varrho_n)$.

  By the truth definition, this holds iff $\mathfrak{M}, \Gamma \Vdash_v \varphi[\varrho/d]$ where $d = (v * \mathcal{I})(t, \Gamma)$. In other words, $\mathfrak{M}, \Gamma \Vdash_{v'} \varphi$ where v' is v except $v'(\varrho) = (v * \mathcal{I})(t, \Gamma)$ (*).

  By the IH and because $\mathfrak{M}, \Gamma \Vdash_{v'} \varphi$, this holds iff $\langle v'(\varrho_1), \ldots, v'(\varrho_n)\rangle \in FT(\varphi(\varrho_1, \ldots, \varrho_n))(\mathbf{I}_{\mathfrak{M}})$ $(*^2)$. Since in subformula φ, ϱ is an unbound variable, by definition 4.2 item 2b, $FT(\varphi(\varrho, \varrho_1, \ldots, \varrho_n)) = VT(\varrho) \times FT(\varphi(\varrho_1, \ldots, \varrho_n))$. Thus $(*^2)$ holds iff $\langle v'(\varrho), v'(\varrho_1), \ldots, v'(\varrho_n)\rangle \in FT(\varphi(\varrho, \varrho_1, \ldots, \varrho_n))(\mathbf{I}_{\mathfrak{M}})$ for any arbitrary valuation v' of ϱ. Since (*), this holds iff $\langle v'(\varrho), v'(\varrho_1), \ldots, v'(\varrho_n) \in \pi_{1,\ldots,n+2}\sigma_{(1=n+3 \wedge n+2=n+4)}$ $(FT(\varphi(\varrho, \varrho_1, \ldots, \varrho_n)) \times \pi_{TT(t),1}Sta)(\mathbf{I}_{\mathfrak{M}})$ $(*^3)$. The last step explained: $n+2 = n+4$ is a join condition on the state identifiers: select only tuples with matching states. The condition $1 = n+3$ ensures that $v'(\varrho)$ is equal to attribute with index $TT(t)$, which by definition 4.2 item 1c, is $\mathfrak{C}(t)$. In other words, only records are selected where $v'(\varrho)$ is equal to the value of attribute $\mathfrak{C}(t)$ in Sta. By definition 4.1 item 8, this means that if and only if $v'(\varrho) = (v' * \mathcal{I})(t, \Gamma)$, the tuple is present in the query image.

  Finally, because v' is v except $v'(\varrho) = (v * \mathcal{I})(t, \Gamma)$ and by definition 4.2, $(*^3)$ holds iff $\langle v(\varrho_1), \ldots, v(\varrho_n)\rangle \in FT(\langle \lambda \varrho.\varphi\rangle(t)(\varrho_1, \ldots, \varrho_n))(\mathbf{I}_{\mathfrak{M}})$.

$\square$

[4]Replace with $\mathcal{D}_c$ if ϱ is a concept variable.

References

[AHV95] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[FM99] Melvin Fitting and Richard L. Mendelsohn. *First-order modal logic*. Kluwer Academic Publishers, Norwell, MA, USA, 1999.