Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 336 (2005) 125–151
Theoretical Computer Science
www.elsevier.com/locate/tcs

On the decidability of termination of query evaluation in transitive-closure logics for polynomial constraint databases ☆

Floris Geerts[1], Bart Kuijpers*

Department of Mathematics, Physics and Computer Science, Department WNI, Limburgs Universitair Centrum, B-3590 Diepenbeek, Belgium

Abstract

The formalism of constraint databases, in which possibly infinite data sets are described by Boolean combinations of polynomial inequality and equality constraints, has its main application area in spatial databases. The standard query language for polynomial constraint databases is first-order logic over the reals. Because of the limited expressive power of this logic with respect to queries that are important in spatial data base applications, various extensions have been introduced. We study extensions of first-order logic with different types of transitive-closure operators and we are in particular interested in deciding the termination of the evaluation of queries expressible in these transitive-closure logics. It turns out that termination is undecidable in general. However, we show that the termination of the transitive closure of a continuous function graph in the two-dimensional plane, viewed as a binary relation over the reals, is decidable, and even expressible in first-order logic over the reals. Based on this result, we identify a particular transitive-closure logic for which termination of query evaluation is decidable and which is more expressive than first-order logic over the reals. Furthermore, we can define a guarded fragment in which exactly the terminating queries of this language are expressible.

Keywords: Data base theory; Constraint databases; Query languages; Query evaluation; Dynamical systems theory; Fixed points

☆ A preliminary version of this work was presented at the 9th International Conference on Data base Theory, Siena, Italy, January 2003.

* Corresponding author.
E-mail addresses: floris.geerts@luc.ac.be (F. Geerts), bart.kuijpers@luc.ac.be (B. Kuijpers).

[1] Part of this work was done while the author was at the Basic Research Unit, Helsinki Institute for Information Technology, Finland.

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2004.10.034

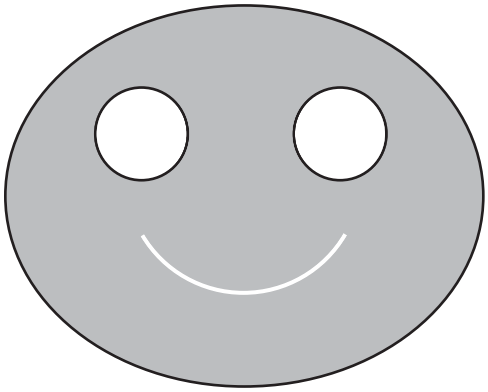

Fig. 1. An example of a constraint data base in $\mathbf{R}^2$.

1. Introduction and summary

The framework of *constraint databases*, introduced in 1990 by Kanellakis et al. [13] and by now well-studied [21,26], provides an elegant and powerful model for applications that deal with infinite sets of points in some real space $\mathbf{R}^n$, for instance spatial databases. In the setting of the constraint model, these infinite sets are finitely represented as Boolean combinations of polynomial equalities and inequalities over the reals. A wide range of geometric figures can be modeled in this way. The smiling face, shown in Fig. 1, is an example of a two-dimensional set that can be described as $\{(x, y) \in \mathbf{R}^2 \mid x^2/25 + y^2/16 \leqslant 1 \wedge x^2 + 4x + y^2 - 2y \geqslant -4 \wedge x^2 - 4x + y^2 - 2y \geqslant -4 \wedge (x^2 + y^2 - 2y \neq 8 \vee y > -1)\}$. An example in a higher dimension is the spatial data base consisting of the set of points on the northern hemisphere together with the points on the equator of the unit sphere in the three-dimensional space $\mathbf{R}^3$. It can be represented by the formula $x^2 + y^2 + z^2 = 1 \wedge z \geqslant 0$.

The relational calculus augmented with polynomial constraints, or first-order logic over the reals augmented with predicates to address the data base, denoted FO for short, is the standard first-order query language for constraint databases. The FO -sentence $(\exists r)(\forall x)(\forall y)(\forall z)(S(x, y, z) \rightarrow x^2 + y^2 + z^2 < r^2)$ expresses that the three-dimensional spatial relation S is bounded. Although variables in such expressions range over the real numbers, queries expressed in this calculus can still be effectively computed, and we have the closure property that says that an FO -query, when evaluated on a constraint data base yields again databases in the constraint model. These properties are direct consequences of a quantifier-elimination procedure for the first-order theory of real closed fields that was first given by Tarski [27].

Although many interesting properties can be expressed in FO, its most important deficiency is that its expressive power is rather limited. For instance, several practically relevant topological properties of spatial data, such as connectivity and reachability, are not expressible in FO [19] and various people have proposed and studied extensions of FO with tractable recursion mechanisms to obtain more expressive languages. For example, datalog versions with constraints have been proposed [12,18,20] (for an overview see [21, Chapter 7]); a programming language extending FO with assignments and a while-loop has been shown to be a computationally complete language for constraint databases [21, Chapter 2]; and extensions of FO with topological predicates have been proposed and studied [2,11]. In analogy with the classical graph connectivity query, which cannot be expressed in the standard relational calculus but which can be expressed in the relational calculus augmented with a transitive-closure operator, also extensions of FO with various *transitive-closure operators* have been proposed. These extensions are more expressive, in particular, they allow the expression of connectivity and reachability queries and some are even computationally complete [10,12,15–17]. Recently, the present authors introduced FO+TC and FO+TCS, two languages in which an operator is added to FO that allows the computation of the transitive closure of unparameterized sets in some $\mathbf{R}^{2k}$ [10]. In the latter language also FO-definable stop conditions are allowed to control the evaluation of the transitive-closure. Later on, Kreutzer has studied the language that we refer to as FO + KTC [16], which is an extension of FO with a transitive-closure operator that may be applied to parameterized sets and in which the computation of the transitive closure can be restricted to certain paths (after specifying certain starting points). The fragments of FO+TCS and FO + KTC, that does not use multiplication, are shown to be computationally complete on databases definable by linear constraints [10,16].

In all of these transitive-closure languages, we face the well-known fact that recursion involving arithmetic over an infinite domain, such as the reals with addition and multiplication in this setting, is not guaranteed to terminate. In this paper, we are interested in termination of query evaluation in these different transitive-closure logics and in particular in *deciding termination*. We show that the termination of the evaluation of a given query, expressed in any of these languages, on a given input data base is undecidable as soon as the transitive closure of 4-ary relations is allowed. In fact, a known undecidable problem in *dynamical systems theory*, namely deciding *nilpotency* of functions from $\mathbf{R}^2$ to $\mathbf{R}^2$ [3,4], can be reduced to our decision problem. When the transitive-closure operator is restricted to work on binary relations, the matter is more complicated. We show the undecidability of termination for FO+TCS restricted to binary relations. However, both for FO+TC and FO + KTC restricted to binary relations, finding an algorithm for deciding termination is related to some outstanding open problems in dynamical systems theory. Indeed, a decision procedure for FO + KTC restricted to binary relations would solve the *point-to-fixed-point problem*. If we can show that testing termination of the evaluation of expressions restricted to binary relations in FO+TC is decidable, we also have decidability of *nilpotency* for functions from $\mathbf{R}$ to $\mathbf{R}$. Both these decision problems from dynamical systems theory are already open for some time [3,14].

For FO+TC restricted to binary relations, we have obtained a positive decidability result, however. A basic problem in this context is deciding whether the transitive closure of a fixed subset of the two-dimensional plane, viewed as a binary relation over the reals, terminates. Even if these subsets are restricted to be the graphs of possibly discontinuous functions from $\mathbf{R}$ to $\mathbf{R}$, this problem is already puzzling dynamical system theorists for a number of years (it relates to the above-mentioned point-to-fixed-point problem). However, when we restrict our attention to the transitive closure of *continuous function graphs*, we can show that the termination of the transitive closure of these figures is decidable. As an illustration of possible inputs for this decision problem, two continuous function graphs

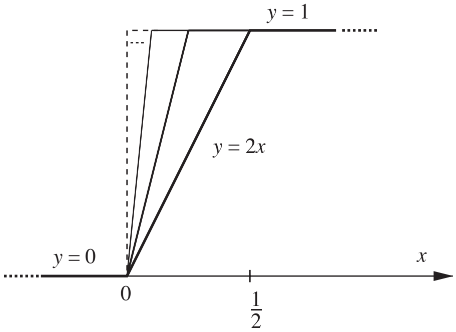

Fig. 2. A function graph (thick) with non-terminating transitive closure (thin).

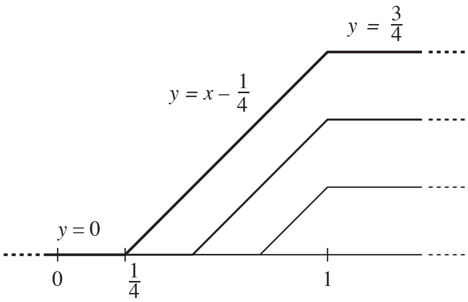

Fig. 3. A function graph (thick) with terminating transitive closure (thin).

are given in Figs. 2 and 3. The one in Fig. 2 has a non-terminating transitive closure, but the one in Fig. 3 terminates after four iterations. Furthermore, we show that this decision procedure is expressible in FO. In the course of our proof, we also give a stronger version of Sharkovskiĭ's theorem [1] from dynamical systems theory for terminating continuous functions. We also extend another result in this area, namely, we show that nilpotency of continuous semi-algebraic functions is decidable and that this decision procedure is even expressible in FO. Previously, this result was only stated, without proof, for continuous piecewise affine functions [4].

Based on this decision result, we define a fragment of FO+TC in which the transitive-closure operator is restricted to work on graphs of continuous functions from **R** to **R**. Termination of queries in this language is shown to be decidable. Furthermore, we define a *guarded* fragment of this transitive-closure logic in which only, and all, terminating queries can be formulated. We also show that this very restricted form of transitive closure yields a language that is strictly more expressive than FO.

This paper is organized as follows. In Section 2, we define constraint databases, the query language FO and several extensions with transitive-closure operators. In Section 3, we give general undecidability results. In Section 4, we give a procedure to decide termination of the transitive closure of continuous function graphs in the plane. In Section 5, we study the extension of FO with a transitive closure operator that is restricted to work on continuous function graphs. In this section, we also describe a guarded fragment of this language and give expressiveness results. The paper concludes with some remarks on generalizations to arbitrary real closed fields.

2. Definitions and preliminaries

In this section, we define constraint databases and their standard first-order query language FO. We also define existing extensions of this logic with different transitive-closure operators: FO+TC, FO+TCS and FO + KTC.

2.1. Constraint databases and first-order logic over the reals

Let **R** denote the set of the real numbers, and $\mathbf{R}^n$ the n-dimensional real space (for $n \geqslant 1$).

Definition 1. An n-dimensional *constraint data base* is a geometrical figure in $\mathbf{R}^n$ that can be defined as a Boolean combination (union, intersection and complement) of sets of the form $\{(x_1, \ldots, x_n) \in \mathbf{R}^n \mid p(x_1, \ldots, x_n) > 0\}$, where $p(x_1, \ldots, x_n)$ is a polynomial with integer coefficients in the real variables $x_1, \ldots, x_n$.

Spatial databases in the constraint model are usually defined as finite collections of such geometrical figures (see [21, Chapter 2]). We have chosen the simpler definition of a data base as a single geometrical figure, but all results carry over to the more general setting.

We remark that in mathematical terminology, constraint databases are called *semi-algebraic sets* [5]. If a constraint data base can be described by linear polynomials only, we refer to it as a *linear constraint data base*.

Example 1. The constraint model allows to describe a wide range of geometrical figures. In the Introduction some examples were given. Fig. 4 shows another example of a constraint data base in $\mathbf{R}^2$ which can be defined by the formula $(x = 0 \wedge 0 \leqslant y \leqslant 2) \vee (-1 \leqslant x \leqslant 1 \wedge y = 2) \vee ((x - \frac{5}{2})^2 + (y - 1)^2 = 1 \wedge x \leqslant \frac{5}{2})$.

We observe that $p(x_1, \ldots, x_n) = 0$ is equivalent to $\neg(p(x_1, \ldots, x_n) > 0) \wedge \neg(-p(x_1, \ldots, x_n) > 0)$, so polynomial equations can be used as well as polynomial inequalities.

In this paper, we will use FO, the relational calculus augmented with polynomial inequalities as a basic query language.

Definition 2. A formula in FO, over an n-dimensional input data base, is a first-order logic formula, $\varphi(y_1, \ldots, y_m, S)$, built, using the logical connectives and quantification over real variables, from two kinds of atomic formulas, namely $S(x_1, \ldots, x_n)$ and $p(x_1, \ldots, x_k) > 0$, where S is a n-ary relation name representing the input data base and $p(x_1, \ldots, x_k)$ is a polynomial with integer coefficients in the real variables $x_1, \ldots, x_k$.

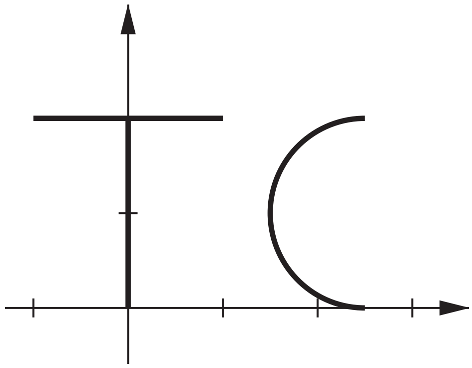

Fig. 4. An example of a constraint data base in $\mathbf{R}^2$.

In the expression $\varphi(y_1, \ldots, y_m, S)$, $y_1, \ldots, y_m$ denote the free variables. Variables in such formulas are assumed to range over $\mathbf{R}$. Tarski's quantifier-elimination procedure for first-order logic over the reals guarantees that FO expressions can be evaluated effectively on constraint data base inputs and their result is a constraint data base (in $\mathbf{R}^m$) that also can be described by means of polynomial constraints over the reals [6,27].

If $\varphi(y_1, \ldots, y_m, S)$ is an FO formula, $a_1, \ldots, a_m$ are reals, and A is an n-dimensional constraint data base, then we denote by $(a_1, \ldots, a_m, A) \models \varphi(y_1, \ldots, y_m, S)$ that $(a_1, \ldots, a_m, A)$ satisfies φ. We denote by $\varphi(A)$ the set $\{(a_1, \ldots, a_m) \in \mathbf{R}^m \mid (a_1, \ldots, a_m, A) \models \varphi(y_1, \ldots, y_m, S)\}$.

The fragment of FO in which multiplication is disallowed is called $\mathrm{FO_{Lin}}$. This fragment is closed on the class of linear constraint databases [21].

Example 2. The FO formula $S(x, y) \wedge (\forall \varepsilon)(\varepsilon > 0 \Rightarrow (\exists v)(\exists w)(\neg S(v, w) \wedge (x - v)^2 + (y - w)^2 < \varepsilon))$ has x and y as free variables. For a 2-dimensional constraint data base S, it expresses the set of points with coordinates (x, y) that belong to the intersection of S and its topological border.

The sentence $(\exists r)(\forall x)(\forall y)(S(x, y, z) \rightarrow x^2 + y^2 + z^2 \leqslant r^2)$ expresses that a given 3-dimensional constraint data base S is bounded.

2.2. Transitive-closure logics

We now define a number of extensions of FO (and of $\mathrm{FO_{Lin}}$) with different types of transitive-closure operators. Recently, the present authors introduced and studied the first two extensions, FO+TC and FO+TCS [9,10]. The latter extension, FO + KTC, is due to Kreutzer [16].

Definition 3. A formula in FO+TC is a formula built in the same way as an FO formula, but with the following extra formation rule: if $\psi(\vec{x}, \vec{y})$ is a formula with $\vec{x}$ and $\vec{y}$ k-tuples of real variables, and with all free variables of ψ among $\vec{x}$ and $\vec{y}$ and if $\vec{s}, \vec{t}$ are k-tuples of real variables, then

$$[\mathrm{TC}_{\vec{x};\vec{y}}\, \psi(\vec{x}, \vec{y})](\vec{s}, \vec{t}) \tag{1}$$

is also a formula which has as free variables those in $\vec{s}$ and $\vec{t}$.

The semantics of a subformula of the above form (1) evaluated on a data base A is defined in the following operational manner: start computing the following iterative sequence of $2k$-ary relations: $X_0 := \psi(A)$ and $X_{i+1} := X_i \cup \{(\vec{x}, \vec{y}) \in \mathbf{R}^{2k} \mid (\exists \vec{z})\, (X_i(\vec{x}, \vec{z}) \wedge X_0(\vec{z}, \vec{y}))\}$ and stop as soon as $X_i = X_{i+1}$. The semantics of $[\mathrm{TC}_{\vec{x};\vec{y}}\, \psi(\vec{x}, \vec{y})](\vec{s}, \vec{t})$ is then defined as $(\vec{s}, \vec{t})$ belonging to the $2k$-ary relation X_i.

Since every step in the above algorithm, including the test for $X_i = X_{i+1}$, is expressible in FO, every step is effective and the only reason why the evaluation may not be effective is that the computation does *not terminate*. In that case the semantics of the formula (1) (and any other formula in which it occurs as subformula) is undefined.

In general, the semantics of a formula φ in FO+TC is evaluated in the standard bottom-up fashion. The result of the evaluation of subformulas is passed on to formulas that are higher up in the parsing tree of φ. Also for the languages FO+TCS and FO + KTC, that we discuss below, this bottom-up evaluation method is used.

Example 3. As an example of an FO+TC formula over a 2-dimensional input data base S, we take

$$[\mathrm{TC}_{x;y}\, S(x, y)](s, t).$$

This expression, when applied to a 2-dimensional figure, returns the transitive closure of this figure, viewed as a binary relation over $\mathbf{R}$.

For illustrations of the evaluation of this formula, we return to the examples in Figs. 2 and 3 in the Introduction. When applied to the graph of the function shown in Fig. 2 (thick lines), we get a non-terminating evaluation. Indeed, in each iteration, line segments of the line $y = 1$ and of a line $y = 2^n x$ for ever larger $n \geqslant 1$ are added. But on input the graph of the function shown in Fig. 3 (thick lines), it terminates after four iterations (since $X_5 = X_4$) and returns the depicted figure (thick plus thin lines).

The language $\mathrm{FO}_{\mathrm{Lin}}$+TC consists of all FO+TC formulas that do not use multiplication.

The following language, FO+TCS, is a modification of FO+TC that incorporates a construction to specify explicit termination conditions on transitive closure computations.

Definition 4. A formula in FO+TCS is built in the same way as an FO formula, but with the following extra formation rule: if $\psi(\vec{x}, \vec{y})$ is a formula with $\vec{x}$ and $\vec{y}$ k-tuples of real variables; σ is an FO sentence over the input data base and a special $2k$-ary relation name X; and $\vec{s}, \vec{t}$ are k-tuples of real variables, then

$$[\mathrm{TC}_{\vec{x};\vec{y}}\, \psi(\vec{x}, \vec{y}) \mid \sigma](\vec{s}, \vec{t}) \tag{2}$$

is also a formula which has as free variables those in $\vec{s}$ and $\vec{t}$. We call σ the *stop condition* of this formula.

The semantics of a subformula of the above form (2) evaluated on databases A is defined in the same manner as in the case without stop condition, but now we stop not only in case an i is found such that $X_i = X_{i+1}$, but also when an i is found such that $(A, X_{i+1}) \models \sigma$, whichever case occurs first. As above, we also consider the restriction $\mathrm{FO}_{\mathrm{Lin}}$+TCS. It was shown that $\mathrm{FO}_{\mathrm{Lin}}$+TCS is computationally complete, in the sense of Turing-complete on the polynomial constraint representation of databases (see [21, Chapter 2]) on linear constraint databases [10].

Example 4. As an example of an FO+TCS formula over a 2-dimensional input data base S, we take

$$[\mathrm{TC}_{x;y}\, S(x, y) \mid (\exists x)(\exists y)(X(x, y) \wedge y = 1 \wedge 10x \leqslant 1)](s, t).$$

When applied to the graph of the function shown in Fig. 2, we see that X_3 satisfies the sentence in the stop condition since for instance $(\frac{1}{16}, 1)$ belongs to it. The evaluation has become terminating (as opposed to the expression without stop condition in Example 3). On input the graph of the function shown in Fig. 3, this expression still terminates after four iterations (since $X_5 = X_4$, not because the stop condition is satisfied) and returns the same result as in the case without stop condition.

Finally, we define FO + KTC. In finite model theory [8], transitive-closure logics, in general, allow the use of parameters. Also the language FO+KTC allows parameters in the transitive closure. Moreover, the computation of the transitive closure can be restricted to certain paths, after specifying certain starting points.

Definition 5. A formula in FO+KTC is a formula built in the same way as an FO formula, but with the following extra formation rule: if $\psi(\vec{x}, \vec{y}, \vec{u})$ is a formula with $\vec{x}$ and $\vec{y}$ k-tuples of real variables, $\vec{u}$ some further ℓ-tuple of free variables, and where $\vec{s}, \vec{t}$ are k-tuples of real terms, then

$$[\mathrm{TC}_{\vec{x};\vec{y}}\, \psi(\vec{x}, \vec{y}, \vec{u})](\vec{s}, \vec{t}) \tag{3}$$

is also a formula which has as free variables those in $\vec{s}, \vec{t}$ and $\vec{u}$.

Since the free variables in $\psi(\vec{x}, \vec{y}, \vec{u})$ are those in $\vec{x}$, $\vec{y}$ and $\vec{u}$, here parameters are allowed in applications of the TC-operator. The semantics of a subformula of the form (3), with $\vec{s} = (s_1, \ldots, s_k)$, evaluated on a data base A is defined in the following operational manner: let I be the set of indices i for which s_i is a constant. Then, we start computing the following iterative sequence of $(2k + \ell)$-ary relations: $X_0 := \psi(A) \wedge \bigwedge_{i \in I}(s_i = x_i)$ and $X_{i+1} := X_i \cup \{(\vec{x}, \vec{y}, \vec{u}) \in \mathbf{R}^{2k+\ell} \mid (\exists \vec{z}) \big(X_i(\vec{x}, \vec{z}, \vec{u}) \wedge \psi(\vec{z}, \vec{y}, \vec{u})\big)\}$ and stop as soon as $X_i = X_{i+1}$. The semantics of $[\mathrm{TC}_{\vec{x};\vec{y}}\, \psi(\vec{x}, \vec{y}, \vec{u})](\vec{s}, \vec{t})$ is then defined as $(\vec{s}, \vec{t}, \vec{u})$ belonging to the $(2k + \ell)$-ary relation X_i.

We again also consider the fragment $\mathrm{FO}_{\mathrm{Lin}}$ + KTC of this language. It was shown that $\mathrm{FO}_{\mathrm{Lin}}$ + KTC is computationally complete on linear constraint databases [16].

Example 5. As an example of an FO+KTC formula over a 2-dimensional input data base S, we take

$$[\mathrm{TC}_{x;y}\, S(x, y)](\tfrac{1}{4}, t).$$

When applied to the graph A of the function, shown in Fig. 2, we see that $X_0 = A \cap \{(x, y) \mid x = \frac{1}{4}\}$ and this set is just $\{(\frac{1}{4}, \frac{1}{2})\}$. Next, X_1 is computed to be $\{(\frac{1}{4}, \frac{1}{2}), (\frac{1}{4}, 1)\}$. In subsequent iterations, no further tuples are added (i.e., $X_2 = X_1$). This example shows that in FO+KTC, the evaluation can be restricted to the computation of certain paths in the transitive closure and this gives control over the termination.

We next make the following remark.

Proposition 1. *All* FO+TC *formulas are expressible in* FO + KTC.

Proof. It is clear that it suffices to show that FO+TC-expressions of the form $[\mathrm{TC}_{\vec{x};\vec{y}}\, \psi(\vec{x}, \vec{y})]$ $(\vec{s}, \vec{t})$ are expressible in FO+KTC. It is readily verified that this formula is equivalently expressed by $(\exists \vec{v})([\mathrm{TC}_{\vec{x};\vec{y}}\, \psi(\vec{x}, \vec{y})](\vec{v}, \vec{t}) \wedge \vec{v} = \vec{s})$, where $\vec{v}$ is a vector of previously unused variables. $\square$

For all of the transitive-closure logics that we have introduced in this section, we consider fragments in which the transitive-closure operator is restricted to work on relations of arity at most $2k$ and we denote this by adding $2k$ as a superscript to the name of the language. For example, in the language FO+TCS^4, the transitive closure is restricted to binary and 4-ary relations.

3. Undecidability of the termination of the evaluation of transitive-closure formulas

The decision problems that we consider in this section and the next take couples (φ, A) as input, where φ is an expression in the transitive-closure logic under consideration and A is an input data base, and the answer to the decision problem is *yes* if the computation of the semantics of φ on A (as defined for that logic) terminates. We then say, for short, that *φ terminates on A*.

Now, we give a general undecidability result concerning termination. In its proof and further on, the notion of nilpotency of a function will be used: a function $f : \mathbf{R}^n \to \mathbf{R}^n$ is called *nilpotent* if there exists a natural number $k \geqslant 1$ such that for all $\vec{x} \in \mathbf{R}^n$, $f^k(\vec{x}) = (0, \ldots, 0)$.

In the proof of the following theorem and further on, we also use the notion of a piecewise affine function. A function $f : \mathbf{R}^n \to \mathbf{R}^n$ is called *piecewise affine* if its graph is a linear semi-algebraic subset of $\mathbf{R}^n \times \mathbf{R}^n$.

Theorem 1. *It is undecidable whether a given formula in* FO+TC^4 *terminates on a given input data base.*

Proof. We reduce deciding whether a piecewise affine function $f : \mathbf{R}^2 \to \mathbf{R}^2$ is nilpotent to deciding whether the evaluation of a formula in FO+TC^4 terminates. For the sake of contradiction, assume that termination of formulas in FO+TC^4 is decidable. For a given piecewise affine function $f : \mathbf{R}^2 \to \mathbf{R}^2$, graph$(f)$, the graph of f, is a semi-algebraic subset of $\mathbf{R}^4$. We give a (hypothetical) procedure to decide whether f is nilpotent:

Algorithm NILPOTENT(input f):

Step 1: Decide (using the decision procedure that exists by assumption) whether the FO+TC^4-query

$$[\mathrm{TC}_{x_1,x_2;y_1,y_2}\, S(x_1, x_2, y_1, y_2)](s_1, s_2, t_1, t_2)$$

terminates on the input graph(f); if the answer is *no*, then return *no*, else continue with Step 2.

Step 2: Compute $f^1(\mathbf{R}^2)$, $f^2(\mathbf{R}^2)$, $f^3(\mathbf{R}^2)$, ... and return *yes* if this ends with $\{(0, 0)\}$, else return *no*.

This algorithm decides correctly whether f is nilpotent. Indeed, suppose that the function f is nilpotent. Then there exists a natural number k such that for all (x, y) in $\mathbf{R}^2$, $f^k(x, y) = (0, 0)$. Therefore, the evaluation of the transitive closure of graph(f) will terminate after at most $2k$ iterations. Therefore, for nilpotent f, also the process in Step 2 is guaranteed to terminate, and the correct answer is produced. Also for functions f that are not nilpotent, it is clear that in both cases (output in Step 1 or in Step 2) the correct answer is returned.

Since nilpotency of piecewise affine functions from $\mathbf{R}^2$ to $\mathbf{R}^2$ is known to be undecidable [4], this completes the proof. □

The following corollary follows immediately from the previous theorem and the fact that FO+TC^4-formulas are in FO+KTC^4 (as shown in Proposition 1).

Corollary 1. *It is undecidable whether a given formula in* FO+KTC^4 *terminates on a given input data base.*

For transitive-closure logics with stop-condition, we even have undecidability for transitive closure restricted to binary relations.

Theorem 2. *It is undecidable whether a given formula in* FO+TCS^2 *terminates on a given input data base.*

Proof. We prove this result by reducing the undecidability of a variant of Hilbert's 10th problem to this decision problem. This variant of Hilbert's 10th problem is deciding whether a polynomial $p(x_1, \ldots, x_{13})$ in 13 real variables and with integer coefficients has a solution in the natural numbers [7,23]. For any such polynomial $p(x_1, \ldots, x_{13})$, let σ_p be the

FO-expressible stop-condition:

$$(\exists x_1)\cdots(\exists x_{13})\left(\bigwedge_{i=1}^{13} X(-1,x_i) \wedge p(x_1,\ldots,x_{13})=0\right).$$

Since, in consecutive iterations of the computation of the transitive closure of the graph of $y = x + 1$, -1 is mapped to $0, 1, 2, \ldots$, it is easy to see that $p(x_1,\ldots,x_{13})$ has an integer solution if and only if $[\mathrm{TC}_{x;y}\ y = x + 1 \mid \sigma_p](s,t)$ terminates. Since the above mentioned Diophantine decision problem is undecidable [7,23], this completes the proof. □

The results, given in this section, are complete for the languages FO+TC, FO+TCS and FO + KTC, apart from the cases FO+TC^2 and FO+KTC^2. The former case will be studied in the next sections. For the latter case, we remark that an open problem in dynamical systems theory, namely, the *point-to-fixed-point problem* reduces to it. This open problem is the decision problem that asks whether for a given algebraic number x_0 and a given piecewise affine function $f : \mathbf{R} \to \mathbf{R}$, the sequence $x_0, f(x_0), f^2(x_0), f^3(x_0), \ldots$, reaches a fixed point. Even for piecewise linear functions with two non-constant pieces this problem is open [3,14]. It is clear that this point-to-fixed-point problem can be expressed in FO+KTC^2. So, we are left with the following unsolved problem.

Open problem 1. *Is it decidable whether a given formula in* FO+KTC^2 *terminates on a given input data base*?

4. Deciding termination for continuous function graphs in the plane

In this section, we study the termination of the transitive closure of a fixed semi-algebraic subset of the plane, viewed as a binary relation over $\mathbf{R}$. We say that a subset A of $\mathbf{R}^2$ has a *terminating transitive closure*, if the query expressed by $[\mathrm{TC}_{x;y}\ S(x,y)](s,t)$ terminates on input A using the semantics of FO+TC. In the previous section, we have shown that deciding nilpotency of functions can be reduced to deciding termination of the transitive closure of their function graphs. However, since it is not known whether nilpotency of (possibly discontinuous) functions from $\mathbf{R}$ to $\mathbf{R}$ is undecidable, we cannot use this reduction to obtain the undecidability in case of binary function graphs. We therefore have another unsolved problem:

Open problem 2. *Is it decidable whether a given formula in* FO+TC^2 *terminates on a given input data base*?

Here, we study the termination of the transitive closure of fixed semi-algebraic function [2] graphs in the plane. Function graphs are easier to deal with than arbitrary sets in $\mathbf{R}^2$. They have the nice property that they have a terminating transitive closure if and only if this transitive closure is also semi-algebraic. For arbitrary sets in $\mathbf{R}^2$ this is not true. Take, for example, the filled triangle with corner points $(0,0)$, $(\frac{1}{4},1)$ and $(\frac{1}{2},1)$ in the plane. This set

[2] A function is called *semi-algebraic* if its graph is semi-algebraic.

has a non-terminating transitive closure. But its transitive closure, which is reached after a countably infinite number of steps, is the filled semi-algebraic triangle with corner points $(0, 0)$, $(0, 1)$ and $(\frac{1}{2}, 1)$. The mentioned property of function-graphs is the following.

Proposition 2. *Let $f : \mathbf{R} \to \mathbf{R}$ be a semi-algebraic function. The graph of f has a terminating transitive closure if and only if the transitive closure*[3] *of the graph of f is semi-algebraic.*

Proof. The only-if direction is trivial, so we focus on the if-direction. So, assume that $\mathrm{TC}(f)$, the transitive closure of the graph of f, is semi-algebraic. The transitive closure of $\mathrm{graph}(f)$ is the set

$$\bigcup_{k\geqslant 1} \{(x, f^k(x)) \mid x \in \mathbf{R}\}.$$

Indeed, it is easily verified that the latter set contains $\mathrm{graph}(f)$ and is transitively closed and therefore contains $\mathrm{TC}(f)$. The other inclusion is trivial.

By the Uniform Bounds Theorem[4] [24] there exists an integer $N_{\mathrm{TC}(f)}$ such that for each $x \in \mathbf{R}$, the cardinality of $\cup_{k\geqslant 1}\{f^k(x)\}$ is less than $N_{\mathrm{TC}(f)}$. Hence, the evaluation of the query expressed by $[\mathrm{TC}_{x;y}\, S(x, y)](s, t)$ will terminate, on input $\mathrm{graph}(f)$, after at most $N_{\mathrm{TC}(f)}$ stages. □

There are obviously classes of functions for which deciding termination of their function graphs is trivial. An example is the class of the piecewise constant functions. In this section, we concentrate on a class that is non-trivial, namely the class of the *continuous semi-algebraic functions from* **R** *to* **R**. The main purpose of this section is to prove the following theorem.

Theorem 3. *There is a decision procedure that on input a continuous semi-algebraic function $f : \mathbf{R} \to \mathbf{R}$ decides whether the transitive closure of* $\mathrm{graph}(f)$ *terminates. Furthermore, this decision procedure can be expressed by a formula in* FO (*over a* 2*-dimensional data base that represents the graph of the input function*).

Before we arrive at the proof of Theorem 3, we give a series of six technical lemma's. First, we introduce some terminology.

Let $f : \mathbf{R} \to \mathbf{R}$ be a continuous function and let x be a real number. We call the set $\{f^k(x) \mid k \geqslant 0\}$ the *orbit of x* (with respect to f). A real number x is said to be a *periodic point of f* if $f^d(x) = x$ for some natural number $d \geqslant 1$. And we call the smallest such d the *period of x* (with respect to f). Let $\mathrm{Per}(f)$ be the set of periodic points of f. If a real number x is not a periodic point of f, but if $f^k(x)$ is periodic for some natural number $k \geqslant 1$, we call x an *eventually periodic point of f* and we call the smallest such number k the

[3] Here, we mean transitive closure in the mathematical sense, i.e., the smallest transitively closed subset of $\mathbf{R}^2$ that contains the graph of f.

[4] The Uniform Bounds (or Uniform Finiteness [28]) theorem, applied to $\mathbf{R}^2$, states that if $A \subseteq R^2$ is a semi-algebraic set, then there exists an integer N_A such that for each $x \in \mathbf{R}$, the set $\{y \in \mathbf{R} \mid A(x, y)\}$ is composed of fewer than N_A intervals and isolated points.

run-up of x (with respect to f). Finally, we call f *terminating* if graph(f) has a terminating transitive closure.

We remark that Lemmas 1–4 hold for arbitrary functions, not only for semi-algebraic ones.

Lemma 1. *The function* $f : \mathbf{R} \to \mathbf{R}$ *is terminating if and only if there exist natural numbers* k *and* d *such that for each* $x \in \mathbf{R}$, $f^k(x)$ *is a periodic point of* f *of period at most* d.

Proof. For the if-direction, if there exist natural numbers k and d such that for each $x \in \mathbf{R}$, $f^k(x)$ is a periodic point of f of period at most d, then clearly each path in the transitive closure of graph(f) is of length at most $k + d$.

For the only-if direction, if the computation of the transitive closure of graph(f) terminates after n iterations, then for each $x \in \mathbf{R}$, $f^n(x)$ is a periodic point of f of period at most n. □

Lemma 2. *Let* $f : \mathbf{R} \to \mathbf{R}$ *be a continuous function. If* f *is terminating, then* Per(f) *is a non-empty, closed and connected part of* $\mathbf{R}$. *In particular,* Per$(f) = f^k(\mathbf{R})$ *for some* $k \geqslant 1$.

Proof. It follows from Lemma 1 that, for a terminating f, there is a bound d on the periods with respect to f and a bound k on the run-ups with respect to f.

Denote by C_i the set of fixed points of f^i, i.e., the set of $x \in \mathbf{R}$ for which $f^i(x) = x$. We first show that Per(f) is closed. Since, Per(f) equals $C_1 \cup \cdots \cup C_d$, it suffices to show that each C_i is closed. Hereto, let x be a point in the closure of C_i and consider a sequence $(x_k)_{k\geqslant 1}$ in C_i converging to x. From the continuity of f it follows that $f^i(x) = \lim_{k\to\infty} f^i(x_k) = \lim_{k\to\infty} x_k = x$. Hence x is in C_i. This implies that C_i is closed.

Now, we show that Per$(f) = f^k(\mathbf{R})$ for some $k \geqslant 1$. The non-emptyness of Per(f) follows immediately from this. It also implies the connectedness of Per(f). Indeed, since f is continuous and $\mathbf{R}$ is connected, also $f^k(\mathbf{R})$ is connected.

Since all the run-ups are smaller than k, it is clear that $f^k(\mathbf{R}) \subseteq$ Per(f). On the other hand, let x be a periodic point of f with period d', with $d' \leqslant d$. Let $y = f^a(x)$ where a is $-k \bmod d'$. Then $f^k(y) = f^{k+a}(x) = f^{qd'}(x)$ for some integer $q \geqslant 1$, since $(k + a) \bmod d' = 0$. Since $f^{qd'}(x) = x$, x belongs to $f^k(\mathbf{R})$ and therefore Per$(f) \subseteq f^k(\mathbf{R})$. □

Lemma 3. *Let* C *be a non-empty, closed and connected part of* $\mathbf{R}$. *If* $f : C \to C$ *is a continuous function and if every* $x \in C$ *is a periodic point of* f, *then* f *or* f^2 *is the identity mapping on* C.

Proof. We remark that C can either be the complete line $\mathbf{R}$ or be of the form $[a, +\infty)$, $(-\infty, b]$ or $[a, b]$ with $a \leqslant b$. We will cover all these cases by taking C to be $[a, b]$, with the understanding that a can be $-\infty$ and/or b can be $+\infty$.

First of all, we observe that f must be a bijection of C. Indeed, let $y \in C$ a periodic point of period d, then $y = f^d(y) = f(f^{d-1}(y)) = f(x)$ with $x = f^{d-1}(y)$. Hence f

is surjective. Next suppose that $f(x) = f(y)$. This implies that $f(x)$ and $f(y)$ are in the same orbit of f, say of period d. Therefore, $x = f^{d-1}(f(x)) = f^{d-1}(f(y)) = y$ and f is also injective.

Since a continuous bijection is either strictly increasing or decreasing, we must have that either $f(a) = a$ and $f(b) = b$, or $f(a) = b$ and $f(b) = a$. To prove the lemma, it suffices to show that $f(a) = a$ and $f(b) = b$ implies that f is the identity mapping. Indeed, the second case reduces to the first when applied to f^2.

So, we assume that $f(a) = a$ and $f(b) = b$. Suppose that there exists an $x_0 \in C$ such that $f(x_0) \neq x_0$. By continuity, this means that there exists an open interval (c, d) containing x_0 such that $f(x) \neq x$ in (c, d). Let (c, d) be maximal with these properties. From the maximality of (c, d) it follows that $f(c) = c$ and $f(d) = d$ and hence $f((c, d)) = (c, d)$ (for the unbounded cases, c and/or d may be $-\infty$ and $+\infty$, or just one of them). Moreover, we have that either $f(x) > x$ for all $x \in (c, d)$, or $f(x) < x$ for all $x \in (c, d)$. Take a point $y \in (c, d)$, then $y, f(y), f^2(y), \ldots$ is a strictly increasing (if $f(x) > x$) or a strictly decreasing (if $f(x) < x$) sequence of points. Hence, (c, d) does not contain any periodic points, which contradicts the premises. Hence, f is the identity mapping on C. □

Lemma 4. *For a continuous and terminating* $f : \mathbf{R} \to \mathbf{R}$, $\mathrm{Per}(f) = \{x \in \mathbf{R} \mid f^2(x) = x\}$.

Proof. If f is terminating, then, by Lemma 2, $\mathrm{Per}(f)$ is a closed and connected. Therefore, Lemma 3 can be applied to f restricted to $\mathrm{Per}(f)$. This shows that $\mathrm{Per}(f) \subseteq \{x \in \mathbf{R} \mid f^2(x) = x\}$. The other inclusion follows from the fact that any x which satisfies $f^2(x) = x$ has period 1 or 2. □

Denote by C_i, as in the proof of Lemma 2, the set of fixed points of f^i, i.e., the set of $x \in \mathbf{R}$ for which $f^i(x) = x$. From the previous lemmas it follows that for continuous and terminating f,

$$\mathrm{Per}(f) = C_1 \cup C_2,$$

and that either $C_2 \setminus C_1$ is empty and C_1 is non-empty or $C_2 \setminus C_1$ is non-empty and C_1 is a singleton with the points of $C_2 \setminus C_1$ appearing around C_1 (remark that $C_1 \subseteq C_2$).

Sharkovskiĭ's theorem [1] from 1964, one of the most fundamental result in dynamical system theory, tells us that for a continuous and terminating $f : \mathbf{R} \to \mathbf{R}$ only periods $1, 2, 4, \ldots, 2^d$ can appear for some integer value d. The previous lemma has the following corollary which strengthens the result of Sharkovskiĭ's for terminating functions.

Corollary 2. *If* $f : \mathbf{R} \to \mathbf{R}$ *is continuous and terminating, then f can only have periodic points with periods* 1 *and* 2.

Further on, in the proof of Theorem 3, we distinguish between functions f for which $C_1 \cup C_2$ is $\mathbf{R}$, and other functions. For the former case, no further tests are needed. For the latter case, however, if $C = C_1 \cup C_2$ is closed and connected, we construct a continuous function $\tilde{f}$ from the given continuous function f, and further investigate $\tilde{f}$.

Let $C = C_1 \cup C_2$ be closed and connected and different from $\mathbf{R}$. Hence, C is of the form $[a, b]$, $[a, +\infty)$ or $(-\infty, b]$.

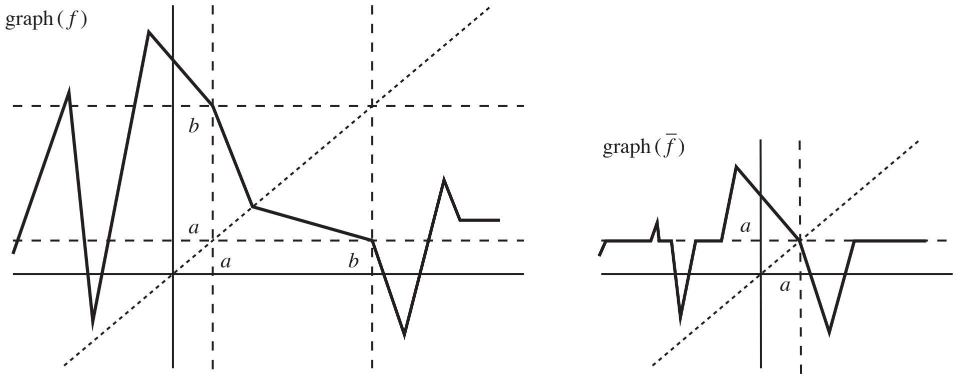

Fig. 5. Illustration of the construction of $\bar{f}$ (right) from f (left).

First, we collapse C to $\{a\}$ if C is bounded or of the form $[a, +\infty)$; and to $\{b\}$ if C is of the form $(-\infty, b]$. Let us first consider the case $C = [a, b]$. Let $f_{\in C} = \{x \in \mathbf{R} \mid f(x) \in C\}$, $f_{<C} = \{x \in \mathbf{R} \mid f(x) < a\}$, and $f_{>C} = \{x \in \mathbf{R} \mid f(x) > b\}$.

We define the continuous function $\bar{f}$ on $\mathbf{R}$ as $\bar{f}(x) :=$

$$\begin{cases} f(x) & \text{if } x \in f_{<C} \text{ and } x < a, \\ f(x) - (b-a) & \text{if } x \in f_{>C} \text{ and } x < a, \\ f(x + (b-a)) & \text{if } x + (b-a) \in f_{<C} \text{ and } x > a, \\ f(x + (b-a)) - (b-a) & \text{if } x + (b-a) \in f_{>C} \text{ and } x > a, \\ a & \text{if } x \in f_{\in C}. \end{cases}$$

This construction is illustrated in Fig. 5.

Let us next consider the case $C = [a, +\infty)$. First, remark that here f is the identity on $[a, +\infty)$, i.e., $C_2 \setminus C_1$ is empty. Here, the function $\bar{f}$ on $\mathbf{R}$ is defined as

$$\begin{cases} f(x) & \text{if } x \in f_{<C} \text{ and } x < a, \\ a & \text{if } x < a \text{ and } x \in f_{\in C} \text{ or if } x \geqslant a. \end{cases}$$

In the case $C = (-\infty, b]$, $\bar{f}$ is defined as

$$\begin{cases} f(x) & \text{if } x \in f_{>C} \text{ and } b < x, \\ b & \text{if } b < x \text{ and } x \in f_{\in C} \text{ or } x \leqslant b. \end{cases}$$

Remark that here f is the identity on $(-\infty, b]$, i.e., $C_2 \setminus C_1$ is also empty in this case.

Finally, we define

$$\tilde{f}(x) := \bar{f}(x + c) - c,$$

where c is a or b, depending on the case.

The following lemma describes the relation between f and $\tilde{f}$. Although, when looking at the graphics this result is intuitively clear, its proof is somehow tedious.

Lemma 5. *Let* $f : \mathbf{R} \to \mathbf{R}$ *be a function with non-empty, closed and connected* $C_1 \cup C_2$ *that is not* $\mathbf{R}$ *and such that* $f(C_1 \cup C_2) = C_1 \cup C_2$. *Then* $f^k(\mathbf{R}) = C_1 \cup C_2$ *if and only if* $\tilde{f}^k(\mathbf{R}) = \{0\}$.

Proof. Let f be as in the statement of the lemma. Abbreviate $C_1 \cup C_2$ by C. Let c denote a or b, depending on the case, as in the above definition of $\tilde{f}$.

From $\tilde{f}(x) := \bar{f}(x+c)-c$ it is easy to show that for $k \geqslant 1$ we have $\tilde{f}^k(x) := \bar{f}^k(x+c)-c$, for example, by straightforward induction on k. From this observation, it immediately follows that $\tilde{f}^k(\mathbf{R}) = \{0\}$ if and only if $\bar{f}^k(\mathbf{R}) = \{c\}$.

It therefore suffices to show that $f^k(\mathbf{R}) = C$ if and only if $\bar{f}^k(\mathbf{R}) = \{c\}$. We first do this for the easier cases where C is unbounded and next prove this equivalence for a bounded interval C.

Let C be $[a, +\infty)$ and thus $c = a$. Here, we show, by induction on $k \geqslant 1$, that for all $x \in \mathbf{R}$ that

$$\bar{f}^k(x) = \min\{a, f^k(x)\}.$$

For $k = 1$, this follows from the definition of $\bar{f}$. Assume, it holds for k. Because $\bar{f}^{k+1}(x) = \bar{f}(\bar{f}^k(x))$, we know that $\bar{f}^{k+1}(x) \leqslant a$. So, if $\bar{f}^{k+1}(x)$ is strictly smaller than a we have to show that it equals $f^{k+1}(x)$. Indeed, from $\bar{f}(\bar{f}^k(x)) < a$ it follows that $f(\bar{f}^k(x)) < a$ and therefore also $f^k(x) < a$ (here we use that $f([a, +\infty)) = [a, +\infty)$). By the induction hypothesis, therefore $\bar{f}^k(x) = f^k(x)$ and $\bar{f}^{k+1}(x) = \bar{f}(\bar{f}^k(x)) = \bar{f}(f^k(x)) = f(f^k(x)) = f^{k+1}(x)$. From $\bar{f}^k(x) = \min\{a, f^k(x)\}$, it follows that for all $x \in \mathbf{R}$, $f^k(x) \geqslant a$ if and only if for all $x \in \mathbf{R}$, $\bar{f}^k(x) = a$. This proves this case.

In the case where C is $(-\infty, b]$, we show in a similar way that for $k \geqslant 1$ and for all $x \in \mathbf{R}$ that $\bar{f}^k(x) = \max\{b, f^k(x)\}$, and this proves this case.

Finally, we have the case where C is a bounded interval $[a, b]$. Here we have $c = a$. To facilitate the notation, we introduce two functions from $\mathbf{R}$ to $\mathbf{R}$: α and β. We define $\alpha(x)$ as

$$\begin{cases} x & \text{if } x < a, \\ a & \text{if } a \leqslant x \leqslant b, \text{ and} \\ x - (b-a) & \text{if } x > b, \end{cases}$$

and $\beta(x)$ as

$$\begin{cases} x & \text{if } x \leqslant a, \\ x + (b-a) & \text{if } x > a. \end{cases}$$

Intuitively, we could say that α maps the domain of f to that of $\bar{f}$ and β does the inverse. Indeed, the composed function $\beta \circ \alpha$ is the identity on $(-\infty, a] \cup (b, +\infty)$ and constant a on the interval $(a, b]$.

It is easily verified that $\bar{f}^k = (\alpha \circ f \circ \beta)^k$, for $k \geqslant 1$. Finally, we define the function g to be $\beta \circ \alpha \circ f$. This function is constant a where f maps numbers in $[a, b]$ and is equal to f on all other numbers. From the above it follows that $\bar{f}^k = \alpha \circ f \circ g^{k-1} \circ \beta$.

Claim. *For all* $k \geqslant 1$ *and all* $x \in \mathbf{R}$, $g^k(x) = a$ *if* $f^k(x) \in [a, b]$ *and* $g^k(x) = f^k(x)$ *if* $f^k(x) \notin [a, b]$.

Proof of the claim. We proceed by induction on $k \geqslant 1$. For $k = 1$, the claim follows from the definition of g. Assume that the claim holds for k. For $k + 1$ there are two cases. Firstly, assume that $f^{k+1}(x) \in [a, b]$. We have to show that $g^{k+1}(x) = a$. There are two subcases. If $f^k(x) \in [a, b]$, then $g^k(x) = a$ by the induction hypothesis and therefore $g^{k+1}(x) = a$. If $f^k(x) \notin [a, b]$, then $g^k(x) \neq a$ by the induction hypothesis and therefore $g^{k+1}(x) = g(g^k(x)) = g(f^k(x)) = a$ since $f(f^k(x)) \in [a, b]$. Secondly, assume that $f^{k+1}(x) \notin [a, b]$. Then $f^j(x) \notin [a, b]$ for all j, $1 \leqslant j \leqslant k + 1$, since $f([a, b]) = [a, b]$. Therefore, $g^k(x) = f^k(x)$. So, $f^{k+1}(x) = f(f^k(x)) = g(f^k(x)) = g(g^k(x)) = g^{k+1}(x)$. The second equality holds since $f^k(x) \notin [a, b]$. This concludes the proof of the claim. □

We are now ready to show that $f^k(\mathbf{R}) = [a, b]$ if and only if $\bar{f}^k(\mathbf{R}) = \{a\}$.

For the if-direction, we assume that $\bar{f}^k(\mathbf{R}) = \{a\}$. Suppose that there exists an $x_0 \in \mathbf{R}$ such that $f^k(x_0) \notin [a, b]$. We claim that $\bar{f}^k(\alpha(x_0)) \neq a$, contradicting the assumption. Indeed, assume that $\bar{f}^k(\alpha(x_0)) = a$. Since $\bar{f}^k(\alpha(x_0)) = (\alpha \circ (f \circ \beta \circ \alpha)^{k-1} \circ f \circ \beta \circ \alpha)(x_0) = (\alpha \circ (f \circ \beta \circ \alpha)^k)(x_0)$, we get that $(f \circ \beta \circ \alpha)^k(x_0) \in [a, b]$. From this follows that $f^k(x_0) \in [a, b]$, contradicting the assumption made about x_0. To prove the latter implication, assume that $f^k(x_0) \notin [a, b]$. Then $f^j(x_0) \notin [a, b]$ for all j with $0 \leqslant j \leqslant k$. From this, and the definition of $\beta \circ \alpha$, it follows that $(f \circ \beta \circ \alpha)^j(x_0) = f^j(x_0)$ for all j with $0 \leqslant j \leqslant k$. This concludes the proof of the if-direction.

For the only-if direction, assume that for all $x \in \mathbf{R}$, $f^k(x) \in [a, b]$. Assume that there exists an $x_0 \in \mathbf{R}$ such that $\bar{f}^k(x_0) \neq a$. Using an above made remark, we therefore have that $(\alpha \circ f \circ g^{k-1} \circ \beta)(x_0) \neq a$ and therefore also $(f \circ g^{k-1} \circ \beta)(x_0) \notin [a, b]$. So, $(g^{k-1} \circ \beta)(x_0) \notin f_{\in[a,b]}$ and therefore certainly $(g^{k-1} \circ \beta)(x_0) \neq a$. Because of the above proven claim we have that therefore $g^{k-1}(\beta(x_0)) = f^{k-1}(\beta(x_0))$. Hence, $\bar{f}^k(x_0) = (\alpha \circ f \circ g^{k-1} \circ \beta)(x_0) = (\alpha \circ f \circ f^{k-1} \circ \beta)(x_0) = (\alpha \circ f^k \circ \beta)(x_0)$. Since this latter value is not equal to a, we have that $f^k(\beta(x_0)) \notin [a, b]$. We conclude that there exists a number y_0, namely $y_0 = \beta(x_0)$, such that $f^k(y_0) \notin [a, b]$. This contradicts the above made assumption and concludes the proof. □

As mentioned in the previous section, in the area of dynamical systems, a function $\tilde{f}$ is called *nilpotent* if $\tilde{f}^k(\mathbf{R}) = \{0\}$ for some integer k. The following lemmas show that this is a decidable property in our setting. For continuous piecewise affine functions this result was already stated (without proof) [4]. So, we extend this result to continuous semi-algebraic functions and furthermore show that the decision procedure is expressible in FO.

Lemma 6. *There is an* FO *sentence that expresses whether a continuous semi-algebraic function* $f : \mathbf{R} \to \mathbf{R}$ *is nilpotent.*

Proof. We describe the algorithm NILPOTENT(input f) to decide nilpotency of continuous semi-algebraic functions $f : \mathbf{R} \to \mathbf{R}$ and later on argue its correctness.

Algorithm NILPOTENT(input f):

Step 1: Compute the set $\{x \in \mathbf{R} \mid f^2(x) = x\}$. If this set differs from $\{0\}$, then answer *no*, else continue with Step 2.

Step 2: Compute the set $B = \{r \mid \gamma_{BB}(r)\}$, where $\gamma_{BB}(r)$ is the formula that defines positive real numbers r that satisfy one of the following three conditions:

(1) $\lim_{x\to-\infty} f(x)$ and $\lim_{x\to+\infty} f(x)$ are constants and $f((-\infty, r]) \subset (-r, +r)$ and $f([r, +\infty)) \subset (-r, +r)$;

(2) $\lim_{x\to-\infty} f(x) = +\infty$ and $\lim_{x\to+\infty} f(x)$ is a constant and $f([r, +\infty)) \subset (-r, +r)$;

(3) $\lim_{x\to-\infty} f(x)$ is a constant and $\lim_{x\to+\infty} f(x) = -\infty$ and $f((-\infty, r]) \subset (-r, +r)$;

If B is empty, answer *no*, else compute the infimum r_0 of B and continue with Step 3.

Step 3: Let g be the function defined as $g(x) := f(x)$ if $-r_0 < x < r_0$ and $g(x) := f(-r_0)$ if $x \leqslant -r_0$ and $g(x) := f(r_0)$ if $x \geqslant r_0$.

If for g there exists a positive real number ε such that

(1) g is constant 0 on $(-\varepsilon, +\varepsilon)$, or

(2) g is constant 0 on $(0, +\varepsilon)$ and has a left tangent with strictly negative slope in 0, or

(3) g is constant 0 on $(-\varepsilon, 0)$ and has a right tangent with strictly negative slope in 0,

then continue with Step 4, else answer *no*.

Step 4: If for all $x > 0$, $g(x) < x$ and $g^2(x) < x$ and for every $x < 0$, $g(x) > x$ and $g^2(x) > x$ holds, then answer *yes*, else answer *no*.

We now prove the correctness of the algorithm NILPOTENT. Clearly, if f has periodic points other than 0, then f cannot be nilpotent. Furthermore, for a nilpotent f, $f(0)$ must be 0. From Sharkovskiĭ's theorem [1], it follows that if f has periodic points of some period d ($d > 1$), then f also has periodic points of period 2. Therefore, the test in Step 1, makes sure that 0 is the only periodic point of f.

In Step 2, the consistency of nilpotency with the behavior of f towards $-\infty$ and $+\infty$ is tested. We first remark that if the limit conditions in either of the three cases are satisfied, also values of r satisfying the inclusion conditions are guaranteed to exist. This follows from the fact that f is semi-algebraic. We show this for Case 2. The other cases are similar. So, assume $\lim_{x\to-\infty} f(x) = +\infty$ and $\lim_{x\to+\infty} f(x) = c$ with c a constant. We have to show that there exists an r such that $f([r, +\infty)) \subset (-r, +r)$. Consider the set $\{x \in \mathbf{R} \mid f(x) < c + 1\}$. This is a semi-algebraic subset of $\mathbf{R}$ that is not bounded towards $+\infty$. Therefore there exists a number d such that $[d, +\infty)$ is completely in $\{x \in \mathbf{R} \mid f(x) < c + 1\}$. It is clear that $r = \max\{c + 1, d\}$ satisfies $f([r, +\infty)) \subset (-r, +r)$.

From the fact that f has a semi-algebraic graph it follows that the set B, computed in Step 2, is empty if (1) $\lim_{x\to-\infty} f(x) = -\infty$ or (2) $\lim_{x\to+\infty} f(x) = +\infty$ or (3) $\lim_{x\to-\infty} f(x) = +\infty$ and $\lim_{x\to+\infty} f(x) = -\infty$.

In Case (1), for all $x < 0$ we have (1a) $f(x) < x < 0$ or (1b) $x < f(x) < 0$. Indeed, because of the test in Step 1, the case $f(x) = x$ cannot occur any more outside the origin. In Case (1a), there exists an infinite orbit $\cdots < f^2(x) < f(x) < x < 0$, hence f is not nilpotent. In Case (1b), there exist arbitrary long orbits converging to x, namely from any point in the sequence $\cdots < f^{-2}(x) < f^{-1}(x) < x < 0$. Hence f is not nilpotent.

For Case (2), a similar analysis can be made, again depending on the graph of f being situated below or above the diagonal.

Also in Case (3), we have this phenomena, this time depending on the graph of f^2 being situated below or above the diagonal. Here, for all $x > 0$, we have (3a) $x < f^2(x)$ or (3b) $x > f^2(x)$. Because of the test in Step 1, there is no third case. In Case (3a), there exists an infinite orbit because $x < f^2(x) < f^4(x) < \cdots$, hence f is not nilpotent. In Case (3b), there exist arbitrary long orbits starting from any point in the sequence $0 < x < f^{-2}(x) < f^{-4}(x) < \cdots$. Hence f is not nilpotent.

Hence, if B is empty, then f is not nilpotent.

If B is non-empty, on the other hand, then $f((-\infty, r_0]) \subset (-r_0, r_0)$ and/or $f([r_0, +\infty)) \subset (-r_0, r_0)$ (depending on which case occurred in Step 2). For the function g, defined in Step 3, this also holds if you replace r_0 by some r_1, with r_1 larger than r_0 and max $\{|g(x)| \mid x \in \mathbf{R}\}$. Furthermore $g([-r_1, +r_1]) \subseteq [-r_1, +r_1]$ holds for such r_1. By the choice of r_1, it follows that f is nilpotent if and only if g is nilpotent. The only-if direction is immediately clear. For the if-direction, we observe that f or f^2 (again depending on the case that occurred in Step 2) maps numbers outside $[-r_1, +r_1]$ into $[-r_1, +r_1]$, and (the behaviour of) f and g are the same within $[-r_1, +r_1]$.

In Steps 3 and 4, the consistency of the behavior of g in a neighborhood of 0 with nilpotency is tested. In the cases (1)–(3), $g^2(x) = 0$ holds for a small ε-environment of 0. Every different behavior of g in the neighborhood of 0, leads to infinitely long or arbitrarily long orbits of g (and hence of f). Since this analysis is completely analogous to the one made in Step 2, we omit the details.

The condition in Step 4, expresses what is known as the global convergence of g [3], which is equivalent to nilpotency of g because g^2 maps a neighborhood of 0 to 0 [4]. That g^2 maps a neighborhood of 0 to 0 follows from Step 3.

Finally, we remark that all computations and tests performed in the algorithm NILPOTENT, are expressible by a FO formula over the binary relation representing the graph of the input f. Limits, for instance, can be implemented in FO using the classical ε-δ definition. □

We are now ready for the proof of Theorem 3.

Proof of Theorem 3. We describe a decision procedure TERMINATE(input f) that on input a function $f : \mathbf{R} \to \mathbf{R}$, decides whether the transitive closure of graph(f) terminates after a finite number of iterations.

Algorithm TERMINATE(input f):

Step 1: Compute the sets $C_1 = \{x \mid f(x) = x\}$ and $C_2 = \{x \mid f^2(x) = x\}$. If C_2 is a closed and connected part of $\mathbf{R}$ and if C_1 is a point with $C_2 \setminus C_1$ around it or if $C_2 \setminus C_1$ is empty, then continue with Step 2, else answer *no*.

Step 2: If C_2 is $\mathbf{R}$, answer *yes*, else compute the function $\tilde{f}$ (as described before Lemma 5) and apply the algorithm NILPOTENT in the proof of Lemma 6 to $\tilde{f}$ and return the answer.

The correctness of this procedure follows from Lemmas 4–6. From the remark at the end of the proof of Lemma 6 and the construction of C_1, C_2 and $\tilde{f}$, it is clear that all ingredients can be expressed in FO. □

Example 6. We use the function f_1, given in Fig. 2 in the Introduction, and the function f_2, given in Fig. 3, to illustrate the decision procedure TERMINATE(input f).

For f_1, $C_1 \cup C_2$ is $\{0, 1\}$, and therefore f_1 does not survive Step 1 and TERMINATE (input f_1) immediately returns *no*.

For f_2, $C_1 \cup C_2$ is $\{0\}$, and therefore we have $\tilde{f_2} = f_2$. Next, the algorithm NILPOTENT is called with input f_2. For f_2, the set B, computed in Step 2 of the algorithm NILPOTENT, is non-empty and r_0 is 2. So, the function g in Step 3 will be f_2 again and r_1 is strictly larger than 2. Since g is identical zero around the origin, finally the test in Step 4 decides. Here, we have that for $x > 0$, $g(x) < x$ and also $g^2(x) < x$ since $x - \frac{1}{4} < x$ and $x - \frac{1}{2} < x$. For $x < 0$, we have that both $g(x)$ and $g^2(x)$ are identical zero and thus the test succeeds also here. The output of NILPOTENT on input f_2 and therefore also the output of TERMINATE on input f_2 is *yes*.

For a continuous and terminating function, the periods that can appear are 1 and 2 (see Lemma 3). In dynamical systems theory, finding an upper bound on the length of the run-ups in terms of some characteristics of the function, is considered to be, even for piecewise affine functions, a difficult problem [22,25]. Take, for instance, the terminating continuous piecewise affine function that is constant towards $-\infty$ and $+\infty$ and that has turning points $(0, \frac{1}{3})$, $(\frac{1}{3}, \frac{2}{3} - \varepsilon)$, $(\frac{4}{9}, \frac{4}{9})$, $(\frac{5}{9}, \frac{5}{9})$, $(\frac{2}{3}, \frac{1}{3})$, and $(1, \frac{2}{3})$, with $\varepsilon > 0$ small. Here, it seems extremely difficult to find an upper bound on the length of the run-ups in terms of the number of line segments or of their endpoints. The best we can say is that also the maximal run-up can be computed.

Corollary 3. *Let $f : \mathbf{R} \to \mathbf{R}$ be a continuous, terminating semi-algebraic function. The maximal run-up of a point in* **R** *with respect to f can be computed.*

We end this section with a remark concerning termination of continuous functions that are defined on a connected part I of $\mathbf{R}$. Let $f : I \to I$ be such a function. We define the function $\bar{f}$ to be the continuous extension of f to $\mathbf{R}$ that is constant on $\mathbf{R} \setminus I$. It is readily verified that the transitive closure of graph(f) terminates if and only if $\bar{f}$ is terminating. We therefore have the following corollary of Theorem 3.

Corollary 4. *Let I be a connected part of* **R**. *There is an* FO *expressible decision procedure that decides whether the transitive closure of the graph of a continuous semi-algebraic function $f : I \to I$ terminates.*

5. Logics with transitive closure restricted to function graphs

In this section, we study fragments of FO+TC and FO+TCS where the transitive-closure operator is restricted to work only on the graphs of continuous semi-algebraic functions from $\mathbf{R}^k$ to $\mathbf{R}^k$. These languages bear some similarity with *deterministic* transitive-closure logics in finite model theory [8].

If $\vec{x}$ and $\vec{y}$ are k-dimensional real vectors and if $\psi(\vec{x}, \vec{y})$ is an FO+TC-formula (respectively FO+TCS-formula), let γ_ψ be the FO+TC-sentence (respectively, FO+TCS-sentence)

$\gamma^1_\psi \wedge \gamma^2_\psi$, where γ^1_ψ expresses that $\psi(\vec{x}, \vec{y})$ defines the graph of a function from $\mathbf{R}^k$ to $\mathbf{R}^k$ and where γ^2_ψ expresses that $\psi(\vec{x}, \vec{y})$ defines a continuous function graph. We can express γ^2_ψ using the classical ε-δ definition of continuity.

More specifically, γ^1_ψ can be written as

$$(\forall \vec{x})(\exists \vec{y})\psi(\vec{x}, \vec{y}) \wedge (\forall \vec{x})(\forall \vec{y})(\forall \vec{z})(\psi(\vec{x}, \vec{y}) \wedge \psi(\vec{x}, \vec{z}) \Rightarrow \vec{y} = \vec{z})$$

and γ^2_ψ can be written as

$$\begin{aligned}&(\forall \vec{x}_1)(\forall \varepsilon > 0)(\exists \delta > 0)(\forall \vec{x}_2)(\|\vec{x}_1 - \vec{x}_2\| < \delta \Rightarrow \\ &\quad (\forall \vec{y}_1)(\forall \vec{y}_2)(\psi(\vec{x}_1, \vec{y}_1) \wedge \psi(\vec{x}_2, \vec{y}_2) \Rightarrow \|\vec{y}_1 - \vec{y}_2\| < \varepsilon)).\end{aligned}$$

Proposition 3. *Let $\psi(\vec{x}, \vec{y})$ be an* FO+TC-*formula* (*respectively*, *an* FO+TCS-*formula*). *The evaluation of $\psi(\vec{x}, \vec{y})$ on an input data base A terminates if and only if the evaluation of γ_ψ on A terminates.*

Proof. It should be clear that the above expressions for γ^1_ψ and γ^2_ψ make direct calls to $\psi(\vec{x}, \vec{y})$ and no new calls to a TC-formula are introduced. Using the bottom-up evaluation method described in Section 2.2, it is clear that evaluation of both γ^1_ψ and γ^2_ψ terminates on A if and only the evaluation of ψ terminates on A. □

Definition 6. We define FO+cTC (respectively, FO+cTCS) to be the fragment of FO+TC (respectively, FO+TCS) in which only TC-expressions of the form $[\mathrm{TC}_{\vec{x};\vec{y}}\, \psi(\vec{x}, \vec{y}) \wedge \gamma_\psi](\vec{s}, \vec{t})$ (respectively, $[\mathrm{TC}_{\vec{x};\vec{y}}\, \psi(\vec{x}, \vec{y}) \wedge \gamma_\psi \mid \sigma](\vec{s}, \vec{t})$) are allowed to occur.

We again use superscript numbers to denote restrictions on the arities of the relations of which transitive closure can be taken.

5.1. *Deciding termination of the evaluation of* FO+cTC^2 *queries*

Since, when $\psi(x, y)$ is $y = x + 1$, γ_ψ is *true*, from the proof of Theorem 2 the following negative result follows.

Corollary 5. *It is undecidable whether a given formula in* FO+cTCS^2 *terminates on a given input data base.*

We remark that for this undecidability it is not needed that the transitive closure of continuous functions on an *unbounded* domain is allowed ($f(x) = x + 1$ in the proof of Theorem 2). Even when, for example, only functions from [0, 1] to [0, 1] are allowed, we have undecidability. We can see this by modifying the proof of Theorem 2 as follows. For any polynomial $p(x_1, \ldots, x_{13})$, let σ_p be the FO-expressible stop-condition:

$$\begin{aligned}&(\exists x_1) \cdots (\exists x_{13})(\bigwedge_{i=1}^{13} ((\exists y_i)(x_i y_i = 1 \wedge X(1, y_i)) \\ &\quad \vee x_i = 0 \vee x_i = 1) \wedge p(x_1, \ldots, x_{13}) = 0).\end{aligned}$$

Since, in consecutive iterations, the continuous extension $\bar{f}$ of $f : [0, 1] \to [0, 1] : x \mapsto \frac{x}{x+1}$, maps 1 to $\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots$, it is then easy to see that $p(x_1, \ldots, x_{13})$ having an integer solution is equivalent to

$$[\mathrm{TC}_{x;y}\, \psi(x, y) \mid \sigma_p](s, t)$$

terminating, where $\psi(x, y)$ defines graph$(\bar{f})$. Remark again that the $\gamma_{\mathrm{graph}(\bar{f})}$ is *true*.

The main result of this section is the following.

Theorem 4. *It is decidable whether a given formula in* FO+cTC^2 *terminates on a given input data base. Moreover, this decision procedure is expressible in* FO+cTC^2.

Proof. Given a formula φ in FO+cTC^2 and an input data base A, we can decide whether the evaluation of φ on A terminates by first evaluating the deepest FO-formulas on which a TC-operator works on A and then using Theorem 3 to decide whether the computation of transitive closure halts on this set. If it does not terminate, we answer *no*, else we compute the result and continue recursively to less deep occurrences of TC-operators in φ. We continue this until the complete formula φ is processed. Only if we reach the end and all intermediate termination tests returned *yes*, we output *yes*.

The expressibility of the decision procedure in FO+cTC^2 can straightforwardly be proven by induction on the structure of the formula. □

5.2. *A guarded fragment of* FO+cTC^2

The fact that termination of FO+cTC^2-formulas is expressible in FO+cTC^2, allows us to define a *guarded* fragment, FO+cTC$^2_{\mathrm{G}}$, of this language. Indeed, if ψ is a formula in FO+cTC^2 of the form $[\mathrm{TC}_{\vec{x};\vec{y}}\, \psi(\vec{x}, \vec{y})](\vec{s}, \vec{t})$, let τ_ψ be the FO+cTC^2-sentence that expresses that this TC-expression terminates (obviously, τ_ψ also depends on the input data base). We can now define the guarded fragment of FO+cTC^2, in which every TC-expression is accompanied by a *termination guard*.

Definition 7. We define FO+cTC$^2_{\mathrm{G}}$ to be the fragment of FO+cTC^2 in which only TC-expressions of the form $[\mathrm{TC}_{\vec{x};\vec{y}}\, \psi(\vec{x}, \vec{y}) \wedge \tau_\psi](\vec{s}, \vec{t})$ are allowed.

The following property follows from the above remarks.

Proposition 4. *In the language* FO+cTC$^2_{\mathrm{G}}$, *every query terminates on all possible input databases. Furthermore, all terminating queries of* FO+cTC^2 *are expressible in* FO+cTC$^2_{\mathrm{G}}$.

Proof. Since, for each expression φ in FO+cTC$^2_{\mathrm{G}}$, every subformula of φ that is a TC-expression includes a termination guard, these subexpressions are guaranteed to terminate on all inputs. Therefore, the evaluation of φ is guaranteed to terminate on every input.

For the second part of this proposition, let φ be a formula in FO+cTC^2 that is terminating on all inputs. By adding termination guards in φ, starting at TC-subformulas that appear deepest and continuing outwards, we obtain a formula $\bar{\varphi}$ in FO+cTC$^2_{\mathrm{G}}$ that equivalently expresses the query expressed by φ. □

5.3. Expressiveness results

Even the least expressive of the discussed transitive-closure logics is still more expressive than first-order logic.

Theorem 5. *The language* FO+cTC$^2_{\mathrm{G}}$ *is more expressive than* FO *on finite constraint databases.*

Proof. Consider the following query Q_{int} on 1-dimensional databases S: "Is S a singleton that contains a natural number?". The query Q_{int} is not expressible in FO (if it would be expressible, then also the predicate $\mathrm{nat}(x)$, expressing that x is a natural number, would be in FO). The query Q_{int} is expressible in FO+cTC$^2_{\mathrm{G}}$ by the sentence that says that S is a singleton that contains 0, 1 or an element $r > 1$ such that $(\exists s)(\exists t)([\mathrm{TC}_{x;y}\, \psi(x, y) \wedge \gamma_\psi \wedge \tau_{\psi(x,y)\wedge\gamma_\psi}](s, t) \wedge s = 1 \wedge t = \frac{1}{r})$, where $\psi(x, y)$ is the formula $(\exists r)(S(r) \wedge \varphi(r, x, y))$. Here, $\varphi(r, x, y)$ defines the graph of the continuous piecewise affine function that maps x to

$$y = \begin{cases} 0 & \text{if } x \leqslant \frac{1}{r}, \\ x - \frac{1}{r} & \text{if } \frac{1}{r} < x < 1, \\ 1 - \frac{1}{r} & \text{if } x \geqslant 1. \end{cases}$$

Remark that γ_ψ is always *true*. The sentence $\tau_{\psi(x,y)\wedge\gamma_\psi}$ is *true* when the data base is a singleton containing a number larger than one. The function given by $\varphi(r, x, y)$ is illustrated in Fig. 3 for $r = 4$. The evaluation of this transitive closure is guaranteed to end after at most $\lceil r \rceil$ iterations and this sentence indeed expresses Q_{int} since $(1, \frac{1}{r})$ belongs to the result of the transitive closure if and only if $r > 1$ is a natural number. □

We remark that the fact that we can express in FO+cTC$^2_{\mathrm{G}}$ that a 1-dimensional singleton databases S contains a natural number does not imply that we can define the natural numbers in FO+cTC$^2_{\mathrm{G}}$. This follows immediately from the guaranteed termination of FO+cTC$^2_{\mathrm{G}}$-expressible queries. On input a constraint data base the evaluation of a FO+cTC$^2_{\mathrm{G}}$-expression is guaranteed to terminate and to return an output that can be described by means of polynomial constraints, i.e., that is semi-algebraic. The set of natural numbers is non-semi-algebraic subset of $\mathbf{R}$ and can therefore not be defined in FO+cTC$^2_{\mathrm{G}}$. Looking at the formula in the above proof that expresses that a 1-dimensional singleton databases S contains a natural number, we see that the therein used TC-expression works on the formula $\psi(x, y)$, which is $(\exists r)(S(r) \wedge \varphi(r, x, y))$. We see that the number r of which naturalness is expressed is bound by a quantifier in the formula $\psi(x, y)$. Therefore, if we would want to define the natural numbers by modifying the formula in the proof this would lead to applying the transitive-closure operator to a formula $\psi'(x, y, r)$ with an extra parameter. This would lead us outside FO+cTC and inside FO + KTC.

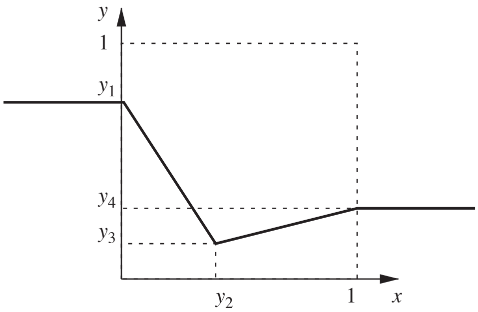

Fig. 6. The graph of the function $f(x, y_1, \ldots, y_4)$ in the (x, y)-plane.

6. Concluding remarks

We conclude with a number of remarks. One of our initial motivations to look at termination of query evaluation in transitive closure logics was to study the expressive power of FO+TC compared to that of FO+TCS. As mentioned in the Introduction and Section 2, the latter language is computationally complete on linear constraint databases. It is not clear whether FO+TC is also complete. In general, we have no way to separate these languages. But if we restrict ourselves to their fragments FO+cTC^2 and FO+cTCS^2, the fact that for the former termination is decidable, whereas it is not for the latter, might give the impression that at least these fragments can be separated. But this is not the case, since equivalence of formulas in these languages is undecidable. In fact, the expressions used in the proof of Theorem 2, are expressible in FO+TC (they do not even use an input data base).

A last remark concerns the validity of the results in Section 4 for more general settings. Lemmas 1–5 are also valid for arbitrary real closed fields R. One could ask whether the same is true for Lemma 6. However, the proof of the correctness of the FO-sentence which decides global convergence in Step 4 [3], relies on the Bolzano–Weierstrass theorem, which is known not to be valid for arbitrary real closed fields [5]. Furthermore, we can even prove the following.

Theorem 6. *Termination of continuous semi-algebraic functions* $f : R \to R$ *for arbitrary real closed fields* R *is not expressible in* FO.

Proof. Let F_R be the family of continuous piecewise affine functions from R to R parameterized by $[0, 1]^4 \subset R^4$ and defined by

$$F_R : R \times [0, 1]^4 \to R : (x, \vec{y}) \mapsto f(x, \vec{y}),$$

where $\vec{y} = (y_1, \ldots, y_4)$ and f is the continuous piecewise affine function that is constant outside $[0, 1]$ and that in the unit interval connects $(0, y_1)$ with (y_2, y_3) and (y_2, y_3) with $(1, y_4)$ (see Fig. 6).

For each $k > 0$, it is clear that there exists an FO-formula $\varphi_k(\vec{y})$ which expresses that the evaluation of the transitive closure of graph$(F_R(\vec{y}))$ terminates after k iterations.

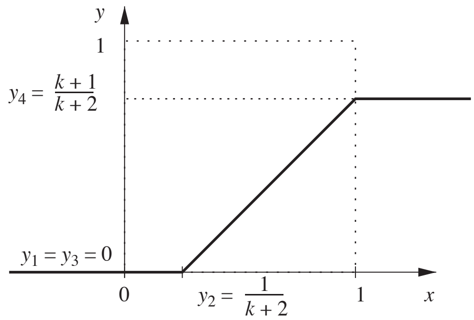

Fig. 7. The graph of the function $f\left(x, 0, \frac{1}{k+2}, 0, \frac{k+1}{k+2}\right)$.

We prove the proposition by contradiction. Suppose that there exists an FO-sentence ψ which expresses the termination of the transitive closure of function graphs for semi-algebraic functions on an arbitrary real closed field R. This implies that there also exists a FO-formula $\psi(\vec{y})$ which expresses that the evaluation of the transitive closure of graph$(F_R(\vec{y}))$ is terminating.

Let ψ_{rcf} be an FO-sentence expressing the axioms of real closed fields. Then, for each $k > 0$, the formula

$$(\exists \vec{y})(\psi(\vec{y}) \wedge \neg\varphi_1(\vec{y}) \wedge \cdots \wedge \neg\varphi_k(\vec{y})) \wedge \psi_{rcf}$$

is satisfied when we consider $R = \mathbf{R}$ and we take $y_1 = 0$, $y_2 = 1/(k+2)$, $y_3 = 0$, and $y_4 = (k+1)/(k+2)$ as parameters. Indeed, the evaluation of the transitive closure of the graph of $f(x, 0, 1/(k+2), 0, (k+1)/(k+2))$ is terminating but only after $k+1$ iterations (see Fig. 7).

Hence, by the compactness theorem, the countable set of formulas $\{\psi_{rcf}, \psi(\vec{y}), \neg\varphi_1(\vec{y}), \neg\varphi_2(\vec{y}), \ldots\}$ is consistent. Hence, there exists a real closed field $\widetilde{R}$ and a $\vec{y} \in \widetilde{R}^4$ such that $\psi(\vec{y})$ expresses that the evaluation of the transitive transitive closure of graph$(F_{\widetilde{R}}(\vec{y}))$ terminates, or equivalently, that $f_{\vec{y}} : \widetilde{R} \to \widetilde{R} : x \mapsto f(x, \vec{y})$ is terminating. However, there exists no k such that $f(x, \vec{y})$ terminates after k iterations. This is clearly a contradiction. Hence, the assumption that φ expresses the termination of functions $f : R \to R$ for arbitrary real closed fields R must be false. □

Acknowledgements

The authors would like to thank Michael Benedikt for suggesting a significant simplification to the proof of Theorem 6.

References

[1] Ll. Alsedà, J. Llibre, M. Misiurewicz, Combinatorial Dynamics and Entropy in Dimension One, Advances Series in Nonlinear Dynamics, Vol. 5, World Scientific, Singapore, 1993.

[2] M. Benedikt, M. Grohe, L. Libkin, L. Segoufin, Reachability and connectivity queries in constraint databases, in: Proc. 19th ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems (PODS'00), ACM, New York, 2000, pp. 104–115.
[3] V.D. Blondel, O. Bournez, P. Koiran, C.H. Papadimitriou, J.N. Tsitsiklis, Deciding stability and mortality of piecewise affine dynamical systems, Theoret. Comput. Sci. 255 (1–2) (2001) 687–696.
[4] V.D. Blondel, O. Bournez, P. Koiran, J.N. Tsitsiklis, The stability of saturated linear dynamical systems is undecidable, J. Comput. System Sci. 62 (3) (2001) 442–462.
[5] J. Bochnak, M. Coste, M.-F. Roy, Real Algebraic Geometry, Ergebenisse der Mathematik und ihrer Grenzgebiete, Vol. 36, Folge 3, Springer, Berlin, 1998.
[6] G.E. Collins, Quantifier elimination for real closed fields by cylindrical algebraic decomposition, in: Automata Theory and Formal Languages, Lecture Notes in Computer Science, Vol. 33, Springer, Berlin, 1975, pp. 134–183.
[7] M. Davis, Y. Matijasevič, J. Robinson, Hilbert's tenth problem, Diophantine equations: positive aspects of a negative solution, in: Mathematical Developments Arising from Hilbert Problems, Vol. 28, American Mathematical Society, Providence, RI, 1976, pp. 323–378.
[8] H.-D. Ebbinghaus, J. Flum, Finite Model Theory, Springer, Berlin, 1995.
[9] F. Geerts, Linear approximation of semi-algebraic spatial databases using transitive closure logic, in arbitrary dimension, in: G. Ghelli, G. Grahne (Eds.), Proc. 8th Internat. Workshop on Database Programming Languages (DBPL'01), Lecture Notes in Computer Science, Vol. 2397, Springer, Berlin, 2002, pp. 182–197.
[10] F. Geerts, B. Kuijpers, Linear approximation of planar spatial databases using transitive-closure logic, in: Proc. 19th ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems (PODS'00), ACM, New York, 2000, pp. 126–135.
[11] Ch. Giannella, D. Van Gucht, Adding a path connectedness operator to FO+poly (linear), Acta Inform. 38 (9) (2002) 621–648.
[12] S. Grumbach, G. Kuper, Tractable recursion over geometric data, in: G. Smolka (Ed.), Proc. Principles and Practice of Constraint Programming (CP'97), Lecture Notes in Computer Science, Vol. 1330, Springer, Berlin, 1997, pp. 450–462.
[13] P.C. Kanellakis, G.M. Kuper, P.Z. Revesz, Constraint query languages, J. Comput. System Sci. 51 (1) (1995) 26–52 A preliminary report appeared in the Proc. 9th ACM Symp. on Principles of Database Systems (PODS'90).
[14] P. Koiran, M. Cosnard, M. Garzon, Computability with low-dimensional dynamical systems, Theoret. Comput. Sci. 132 (1994) 113–128.
[15] S. Kreutzer, Fixed-point query languages for linear constraint databases, in: Proc. 19th ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems (PODS'00), ACM, New York, 2000, pp. 116–125.
[16] S. Kreutzer, Operational semantics for fixed-point logics on constraint databases, in: R. Nieuwenhuis, A. Voronkov (Eds.), Proc. 8th Internat. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'01), Lecture Notes in Computer Science, Vol. 2250, Springer, Berlin, 2001, pp. 470–484.
[17] S. Kreutzer, Query languages for constraint databases: first-order logic, fixed-points, and convex hulls, in: J. Van den Bussche, V. Vianu (Eds.), Proc. 8th Internat. Conf. on Database Theory (ICDT'01), Lecture Notes in Computer Science, Vol. 1973, Springer, Berlin, 2001, pp. 248–262.
[18] B. Kuijpers, J. Paredaens, M. Smits, J. Van den Bussche, Termination properties of spatial Datalog programs, in: D. Pedreschi, C. Zaniolo (Eds.), Internat. Workshop on Logic in Databases (LID'96), Lecture Notes in Computer Science, Vol. 1154, Springer, Berlin, 1996, pp. 101–116.
[19] B. Kuijpers, J. Paredaens, J. Van den Bussche, Topological elementary equivalence of closed semi-algebraic sets in the real plane, J. Symbolic Logic 65 (4) (2000) 1530–1555.
[20] B. Kuijpers, M. Smits, On expressing topological connectivity in spatial datalog, in: V. Gaede, A. Brodsky, O. Gunter, D. Srivastava, V. Vianu, M. Wallace (Eds.), Proc. Workshop "Constraint Databases and their Applications" (CDB'97), Lecture Notes in Computer Sciences, Vol. 1191, Springer, Berlin, 1997, pp. 116–133.
[21] G.M. Kuper, J. Paredaens, L. Libkin, Constraint Databases, Springer, Berlin, 2000.
[22] J. Llibre, C. Preston, Personal Communication. 2002.
[23] Y. Matiyasevich, Hilbert's Tenth Problem, The MIT Press, Cambridge, MA, 1993.
[24] A. Pillay, C. Steinhorn, Definable sets in ordered structures, III, Trans. Amer. Math. Soc. 309 (1988) 469–476.

[25] C. Preston, Iterates of Maps on an Interval, Lecture Notes in Mathematics, Vol. 999, Springer, Berlin, 1983.
[26] P. Revesz, Introduction to Constraint Databases, Springer, Berlin, 2002.
[27] A. Tarski, A Decision Method for Elementary Algebra and Geometry, University of California Press, California, 1951.
[28] L. van den Dries, Tame Topology and O-minimal Structures, Cambridge University Press, Cambridge, 1998.