THE UNIVERSITY
OF BIRMINGHAM

Real Functions Computable by Finite Transducers using Affine IFS Representations

Michal Konečný

CSR-00-10
July 2000

School of Computer Science
The University of Birmingham
Birmingham B15 2TT
United Kingdom

*School of Computer Science
Research Reports*

Abstract

This report tries to classify the functions of type $I^n \to I$ (for some interval $I \subseteq \mathbb{R}$) that can be exactly realized by a finite transducer. Such a class of functions strongly depends on the choice of real number representation. This report considers so-called IFS representations where numbers are represented by infinite compositions of contracting functions on I. IFS representations include the radix (e.g. decimal, signed binary) and LFT representations. The results in this report apply to those IFS representations that use affine contractions, like the radix representations, for example.

The first result is that all piecewise affine functions of n variables with rational coefficients are computable by a finite transducer which uses the signed binary representation. The second and main result is that any function computable by a finite transducer using an affine IFS representation is affine on any open connected subset of I^n on which it is continuously differentiable. This limitation theorem shows that the set of finitely computable functions is very restricted.

Contents

1 Introduction

In many computer applications, there is the need to represent and compute with real numbers. Nevertheless, real numbers, being infinite objects in substance, cannot generally be stored in computer memory nor supplied to a digital computer process as an argument in finite time. The traditional solution to this problem is to approximate real numbers by a fairly large finite set of rational numbers called the floating point numbers. In many cases, this approximation significantly deviates from what it should represent due to accumulated round-off errors (cf. [Ménissier-Morain 1996] for a thorough investigation). One solution to this problem (as used by numerical analysts) is to develop algorithms that are stable—despite round-off errors, they calculate a result with a guaranteed degree of accuracy. Another solution is to abandon the floating point and compute without round-off errors by means of some representation of real numbers that allows arbitrary accuracy—so-called exact real number computation. We restrict ourselves to this kind of computation in this report.

There has been extensive research on which real functions are computable exactly using various real number representations and computational models (see for example [Weihrauch 1997, Escardó 1997, Edalat & Sünderhauf 1997] among the more recent studies). There has also been research on complexity theory of real number computation by Ko and others [Ko & Friedman 1982, Ko 1991]. Their research has been mainly focused on higher complexity classes like P, NP and EXP.

We study one of the lowest complexity classes, the class of functions computable with constant memory usage. This is a proper subclass of the linear time complexity class. Such a study is motivated mainly by a desire to get algorithms for very fast and cheap computation of some functions and to show that others cannot be computed so cheaply, no matter what representation or technique is used. As a byproduct, this study might contribute to our understanding of exact real number representation and computation in general.

1.1 Computational models

Computation with exact real numbers can be organized as an incremental stream computation or as a query-and-answer dialogue. In the incremental stream-based approach (e.g. used in [Escardó 1997, Potts 1998]), a machine reads infinite streams that represent the arguments and produces another infinite stream that represents the result. Each finite prefix of an infinite representing stream is an approximation to the represented value. In the query-and-answer computational model, a query usually states the precision and the answer is an approximation to the value of the given precision. Examples of the non-incremental approach are exact real number algorithms by Ménissier-Morain [1996] and the oracle Turing machines studied by Ko [1991].

The latter approach has the advantage that no effort is wasted in producing intermediate results and in storing information which would be used only for processing a precision higher than the one actually needed. The former one can also have an advantage: its incrementality allows the computation to proceed without either (a) having to calculate the desired precision for each argument before knowing anything about them or (b) risking redundant work caused by multiple overlapping queries about the same argument or (c) by querying for an unnecessarily high precision about an argument.

1.2 Finite computability

Since we are studying computability with a limited memory, we can consider only input of a limited amount of data at a time. Therefore we choose the stream processing incremental computation model, embodied by finite transducers.

The difficulty in studying a low level complexity class is that it very much depends on the representation[1]. For example, nullary functions, i.e. constants, that can be computed with a fixed amount of memory are exactly those whose representation is periodic. The set of numbers that have a periodic representation is different in different representations.

If we are successful in exploring how the set of first order finitely computable functions depends on the representation, we might get a tool for comparing the efficiency of representations. If a representation allows more of the important basic functions to be computable by finite transducers, it can be considered more efficient in a certain sense.

There has been some work in the area of real number computability by finite machines. There are exact real algorithms (e.g. for calculating the binary arithmetical mean) that can be implemented as finite machines using the signed binary representation. Heckmann [1998] has shown which unary and binary Möbius transformations are computable in a representation by Potts & Edalat [1997][2]. This report extends this result for a wider class of functions and a wider class of representations. We consider total and single-valued n-ary functions for $n \in \mathbb{N}$ as well as unary functions. The only restriction which we put on a function before we can say something strong about its finite computability is that it is continuously differentiable within some region.

Similar results to ours, but restricted to a particular variant of the binary representation, have been published very recently by Lisovik, Shkaravskaya and others. In [Lisovik & Shkaravskaya 1995], it is stated and, in [Lisovik & Shkaravskaya 1998], fully proved that whenever a push-down unary transducer computes a function f differentiable on an interval (a, b), then f is affine on (a, b). In the latter paper, the same technique was

[1] First order computability is independent of the choice of representation as long as a set of basic functions is computable [Brattka 1999]. The class of polynomial time computable real functions is the same for the most common representations [Ko 1991].

[2] This representation is a translation of the signed binary representation to another interval.

used to prove an even stronger theorem stating that a function computed by a finite transducer cannot be properly convex on any interval (a, b).

We will prove a result similar to the former one but holding for transducers that use *any affine representations* and for functions of *multiple arity*. On the other hand, our technique requires a stronger assumption of *continuous* differentiability.

Shkaravskaya showed in [1998] that any affine function $f\colon I_1 \times \cdots \times I_n \to I_1 \times \cdots \times I_n$ with rational coefficients defined on a compact[3] hyper-rectangle with rational coordinates is computable by a finite transducer. In [Lisovik et al. 1999], it is shown that any unary piecewise affine function with rational coefficients is computable by a finite transducer.

We will extend also these results by showing that any piecewise affine *multiple argument* function with rational coefficients is computable by a finite transducer using the signed binary representation.

1.3 Structure of the article

In Section 2, the class of IFS representations is defined as well as its most common computationally feasible example, the signed binary representation.

Formal definition of a finite transducer with n input and one output channels follows in Section 3. The definition is tailored to our exact real arithmetic purposes so that all input and output symbols are digits of some IFS representations. A general transducer computes a stream function on the level of symbols. This function can be factored by the (to some extent redundant) representations to become a partial multi-valued function on the real numbers.

In Section 4, we make a crucial observation that in transducers which use IFS representations, we can associate a function with each reachable state. This function is (a) the function which would be computed if this state would be initial, (b) derived from the function of the initial state. This observation leads to two propositions. One is useful for proving the main limitation theorem in Section 8 and the other one for constructing finite transducers and for proving them correct.

The latter proposition is fully utilized in Section 5 to show that all piecewise affine n-ary functions with rational coefficients are finitely computable using the signed binary representation.

The next three sections (i.e. Sections 6–8) gradually proceed to the main limitation theorem for affine representations. Section 6 proves it for strongly connected unary transducers, Section 7 for arbitrary unary transducers and finally Section 8 generalizes it to transducers with multiple input channels.

[3] The original assumption is a bit weaker but not substantially for our context.

2 IFS Representations

The abbreviation IFS stands for 'iterated function system' as it is defined and used in fractal theory [Hutchinson 1981, Edalat 1995]. An IFS is a finite collection of contracting functions on some space with a distance. The largest set which is fixed on the union of images of all of the functions is called its attractor. We will not define IFS formally because we need only a modified special case of IFS. The reason for that is that we use IFS for a different purpose than fractal theory. The IFS which we need operates only on real number intervals and its attractor is the whole space.

Let us denote by $\mathcal{I}$ the set of all non-singleton compact real intervals and for each $I \in \mathcal{I}$, let $\mathcal{I}(I)$ be the set of all compact subintervals of I.

The following definition of a contraction to be used in our version of IFS is specific for real numbers representation only. Unlike the traditional definition, it does not explicitly mention any distance measure on the domain.

Definition 2.1 (D-contraction).
An endo-function $f\colon I \to I$ on some interval $I \in \mathcal{I}$ is called a *d-contraction* if

(1) f has a unique fixpoint

(2) f is continuously differentiable

(3) for each $x \in I$ it holds $0 < f'(x) < \infty$ [4]

We will denote the unique fixpoint of f by Fix_f. The derivative in the fixpoint $f'(\mathrm{Fix}_f)$ of a d-contraction f is called its *strength*[5] and is denoted by Strg_f.

The maximal extension of a d-contraction to intervals is denoted by $\overline{f}$:

$$\overline{f}\colon \mathcal{I}(I) \to \mathcal{I}(I) : [a, b] \mapsto [f(a), f(b)]$$

The interval $\overline{f}(I)$ is denoted by $[\![f]\!]$.
The set of all d-contractions on I is denoted by dContr_I.

The notion of d-contraction differs from the standard notion of a contraction on a real number interval which fulfills (2) and (3) above. A contraction is defined as an endo-function which strictly decreases a distance measure of any two distinct points in the domain of the function. This means that a d-contraction does not need to be a contraction relative to the standard metrics distance on $\mathbb{R}$ (see the illustration in Figure 1). The fact that there is no single distance measure with respect to which all d-contractions

[4]The range of the derivative is often being extended to the values $+\infty$ and $-\infty$.
[5]The closer its strength is to zero, the stronger the contraction contracts if iterated.

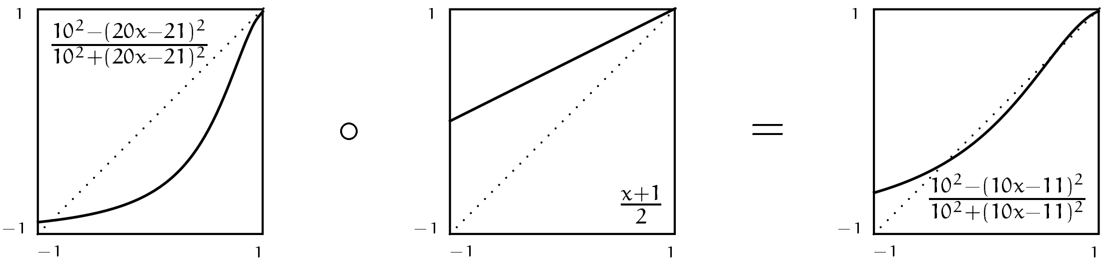

Figure 1: Graphs of two d-contractions and their composition

would be contractions is demonstrated by the failure of d-contractions to be closed on composition.

Despite the difference between contractions and d-contractions, one can show that for every d-contraction on I, there is a metrics on I equivalent to the standard one relative to which it is a contraction (see [Konečný 2000]).

Definition 2.2 (D-IFS).
A tuple $\mathcal{S} = (I, D, \rho)$ where

1. $I \in \mathcal{I}$ is the *base* interval
2. D is a finite set of symbols called *digits*
3. ρ: $D \to \mathrm{dContr}_I$ is an interpretation of the digits

is called a *d-Iterated Function System* (d-IFS) over the interval I.

The components of an d-IFS $\mathcal{S}$ are also denoted by $I_{\mathcal{S}}$, $D_{\mathcal{S}}$ and $\rho_{\mathcal{S}}$.

A d-IFS $\mathcal{S}$ is *affine* if $\rho(\partial)$ is affine for every $\partial \in D_{\mathcal{S}}$.

Example 2.3 (Signed binary representation).
The d-IFS $\mathcal{S}_{\text{sigbin}} = ([-1, 1], \{\bar{1}, 0, 1\}, \rho_{\text{sigbin}})$ where

$$\rho_{\text{sigbin}}(\bar{1})(x) = \frac{x-1}{2}, \qquad \rho_{\text{sigbin}}(0)(x) = \frac{x}{2}, \qquad \rho_{\text{sigbin}}(1)(x) = \frac{x+1}{2}$$

is called the *signed binary representation* (see Figure 2).

Definition 2.4 (Semantics of d-IFS).
For an d-IFS (I, D, ρ), we can define the extension $[\![\cdot]\!]\colon D^\infty \to \mathcal{I}[I]$ of $\partial \mapsto [\![\rho(\partial)]\!]$ for all arbitrary sequences of digits:

$$\begin{aligned}
[\![\epsilon]\!] &= I \\
[\![\partial_1 \cdots \partial_k]\!] &= \left(\overline{\rho(\partial_1)} \circ \cdots \circ \overline{\rho(\partial_k)}\right)(I) \\
[\![\partial_1 \partial_2 \partial_3 \cdots]\!] &= \bigcap_{k \in \mathbb{N}} [\![\partial_1 \cdots \partial_k]\!]
\end{aligned}$$

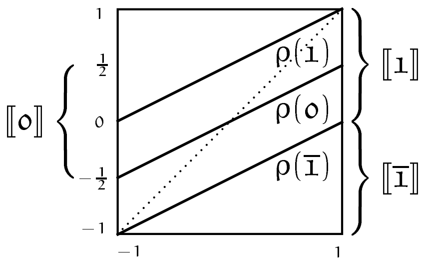

Figure 2: The d-IFS interpretation of the signed binary representation digits.

The semantic function $[\![\cdot]\!]$, in the finite case, returns the range of the composition of digit functions. This range is thought of as the information captured by the sequence of digits. By this set of real numbers it is meant 'the number which is being represented is within this set'. The previous definition expresses the idea that further digits in a sequence *refine* the information given by the preceding ones. Each successive digit restricts the domain of the function given by its preceding digits. Thus it also restricts the range of the new composite function to a smaller range (see Figure 3 on page 8 for an illustration).

Lemma 2.5. *In the signed binary representation, it holds*

$$[\![\partial_1 \cdots \partial_n]\!] = \Big[\sum_{1 \leqslant i \leqslant n} (2^{-i} \cdot \partial_i) - 2^{-n} \,,\ \sum_{1 \leqslant i \leqslant n} (2^{-i} \cdot \partial_i) + 2^{-n} \Big]$$

$$[\![\partial_1 \partial_2 \partial_3 \ldots]\!] = \big\{ 2^{-1} \cdot \partial_1 + 2^{-2} \cdot \partial_1 + 2^{-3} \cdot \partial_1 + \cdots \big\}$$

where the $\partial_i \in \{\bar{1}, 0, 1\}$ *on the right hand side of the above equations are interpreted as the numbers* $-1, 0, 1$*, respectively.*

Proof. Let us proceed by induction on n. For $n = 0$, it holds

$$[\![\epsilon]\!] = I = [-1, 1] = [-2^0, +2^0].$$

Suppose the lemma holds for $n - 1$.

$$\begin{aligned}
[\![\partial_1 \cdots \partial_n]\!] &= \left(\overline{\rho(\partial_1)} \circ \cdots \circ \overline{\rho(\partial_n)}\right)(I) = \overline{\rho(\partial_1)}\left([\![\partial_2 \cdots \partial_n]\!]\right) = \\
&= \overline{\rho(\partial_1)}\Big(\Big[\sum_{1 \leqslant i \leqslant n-1} (2^{-i} \cdot \partial_{i+1}) - 2^{-n+1} \,,\ \sum_{1 \leqslant i \leqslant n-1} (2^{-i} \cdot \partial_{i+1}) + 2^{-n+1} \Big]\Big) = \\
&= \left[\frac{\sum_{i=1}^{n-1} (2^{-i} \cdot \partial_{i+1}) - 2^{-n+1} + \partial_1}{2} \,,\ \frac{\sum_{i=1}^{n-1} (2^{-i} \cdot \partial_{i+1}) + 2^{-n+1} + \partial_1}{2} \right] = \\
&= \Big[\sum_{1 \leqslant i \leqslant n} (2^{-i} \cdot \partial_i) - 2^{-n} \,,\ \sum_{1 \leqslant i \leqslant n} (2^{-i} \cdot \partial_i) + 2^{-n} \Big].
\end{aligned}$$

The second equation in the lemma follows from the first one and from the definition of $[\![\cdot]\!]$ and the fact that a shrinking sequence of intervals converges to a point when its sizes converge to zero. □

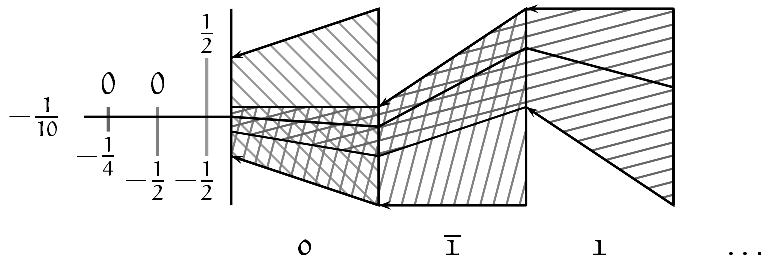

Figure 3: Representing $-1/10$ in the signed binary representation.

Although we called the d-IFS $\mathcal{S}_{\text{sigbin}}$ a 'representation', we still do not have a general notion of when a d-IFS is a representation and when it isn't. With the semantic function $[\![\cdot]\!]$, it can be easily defined as follows.

Definition 2.6 (IFS representation).
A d-IFS (I, D, ρ) for which it holds:

1. $(\forall \alpha \in D^\omega)(\exists r \in I)([\![\alpha]\!] = \{r\})$ (digits contract strongly enough)
2. $(\forall r \in I)(\exists \alpha \in D^\omega)([\![\alpha]\!] = \{r\})$ (every number can be represented)

is called an IFS representation.

Lemma 2.7 (Alternative definition).
The previous definition is equivalent to its variant in which point (2) is replaced by the following condition:

(2') $(\forall r \in I)(\exists \mathfrak{d} \in D)(r \in [\![\mathfrak{d}]\!])$

Proof. Let (I, D, ρ) be an IFS-representation and $r \in I$. By point (1), there is $\mathfrak{d}_1\mathfrak{d}_2 \cdots \in D^\omega$ with $[\![\mathfrak{d}_1\mathfrak{d}_2 \cdots]\!] = \{r\}$ which implies $r \in \overline{\rho(\mathfrak{d}_1)}(I) = [\![\mathfrak{d}_1]\!]$.

On the other hand, suppose (I, D, ρ) is a d-IFS for which points (1) and (2') hold and pick $r \in I$. Define two infinite sequences $r_1, r_2, \cdots \in I$ and $\mathfrak{d}_1, \mathfrak{d}_2, \cdots \in D$ inductively as follows: $r_1 = r$, $\mathfrak{d}_i$ be a digit with $r_i \in [\![\mathfrak{d}_i]\!]\,(I)$ (existence of which is guaranteed by (2')) and r_{i+1} be the number for which $[\![\mathfrak{d}_i]\!]\,(r_{i+1}) = r_i$. Now it holds $r \in [\![\mathfrak{d}_1 \cdots \mathfrak{d}_i]\!]$ for every $i > 0$ and thus $r \in [\![\mathfrak{d}_1\mathfrak{d}_2 \cdots]\!]$. By point (1) we get that $\{r\} = [\![\mathfrak{d}_1\mathfrak{d}_2 \cdots]\!]$. $\square$

Figure 3 shows how several initial signed binary digits produce interval approximations to the represented number by means of composition of their d-contractions.

The following is an obvious property. We point it out because it is useful in the proofs to come.

Lemma 2.8. *Let* (I, D, ρ) *be an IFS representation and* $J \subseteq I$ *an open interval. There exists a sequence* $u \in D^*$ *such that* $[\![u]\!] \subset J$.

Proof. Take any point $r \in J$ and some of its representations $\alpha \in D^\omega$: $[\![\alpha]\!] = r$. Because J is an open neighbourhood of r and $\bigcap_{k\to\infty} [\![\alpha[1,k]]\!] = \{r\}$, for some k it has to hold $[\![\alpha[1,k]]\!] \subset J$. Put $u = \alpha[1,k]$. □

Although d-contractions are not closed on composition in general, those that constitute an IFS representation have to be closed on mutual composition:

Lemma 2.9 (Superposition of digits).
In any IFS representation (I, D, ρ)*, any composition* $\rho(\mathfrak{d}_1) \circ \cdots \circ \rho(\mathfrak{d}_n)$ *of finitely many digits* $\mathfrak{d}_1, \ldots, \mathfrak{d}_n \in D$ *is a d-contraction.*

Proof. Denote $u = \mathfrak{d}_1 \cdots \mathfrak{d}_n$ and $f = \rho(\mathfrak{d}_1) \circ \cdots \circ \rho(\mathfrak{d}_n)$. We need to prove that f is a d-contraction. Since it is a composition of d-contractions, it is continuously differentiable and its derivative is strictly positive. All that remains to prove is that it has a unique fixpoint.

By point (1) of Definition 2.6, there is a number $r \in I$ such that $[\![uuu\cdots]\!] = \bigcap_{i\in\mathbb{N}} f^i(I) = \{r\}$. It follows that r is a fixpoint of f. On the other hand, if x is a fixpoint of f, then $x \in f^i(I)$ for any $i \in \mathbb{N}$. This means that $x \in [\![uuu\cdots]\!]$, i.e. $x = r$. □

Definition 2.10. For a sequence of digits $\mathfrak{d}_1 \cdots \mathfrak{d}_n \in D_{\mathcal{S}}^*$ of an IFS representation $\mathcal{S}$, we denote the d-contraction $\rho_{\mathcal{S}}(\mathfrak{d}_1) \circ \cdots \circ \rho_{\mathcal{S}}(\mathfrak{d}_n)$ by $\rho_{\mathcal{S}}(\mathfrak{d}_1 \cdots \mathfrak{d}_n)$.

It is a standard observation that it makes a significant difference whether the ranges of the digit d-contractions of an IFS representation overlap by their interiors or not. This is closely related to the *redundancy* of the representation—how many different representing sequences for various numbers there exist. The more the ranges overlap the more frequent it is that there is a choice of which digit might be the next one to represent a certain number.

The decimal representation is virtually not redundant and due to this fact, it does not allow to compute even such functions like $x \mapsto 3x$. Mazenc [1993] and also Brattka & Hertling [2000], for a different context, have shown that such a deficiency is inevitable in a very general case of representations that are not redundant to a considerable extent.

Definition 2.11 (Open representation).
An IFS representation (I, D, ρ) is called *open* if for each $r \in I^o$ (i.e. in the interior of I), there exists a digit $\mathfrak{d} \in D$ such that $r \in [\![\mathfrak{d}]\!]^o$.

Lemma 2.12 (Open approximation).
In an open IFS representation (I, D, ρ) *for every number* $r \in I^o$*, there is a sequence* $\alpha \in D^\omega$ *such that every finite prefix* $\alpha[1,n]$ *gives an* open *approximation to* r*:* $(\forall n \in \mathbb{N})(r \in [\![\alpha[1,n]]\!]^o)$.

Proof. In a similar fashion like in the proof of Lemma 2.7, define $\alpha = \mathfrak{d}_1\mathfrak{d}_2\cdots$ inductively together with a sequence $r_1, r_2, \ldots \in I^o$. First, set $r_1 = r$ and let every $\mathfrak{d}_i$ be a digit for which $r_i \in [\![\mathfrak{d}_i]\!]^o$ and $r_{i+1} = \rho(\mathfrak{d}_i)^{-1}(r_i)$. Since $r_i \in [\![\mathfrak{d}_i]\!]^o$, it holds $r_{i+1} \in I^o$ because $\rho(\mathfrak{d}_i)$ is an increasing function. Now we can prove that α has the desired property because it holds

$$r = (\rho(\mathfrak{d}_1) \circ \cdots \circ \rho(\mathfrak{d}_i))\,(r_{i+1}) = \rho(\alpha[1,n])(r_{i+1}).$$

Since $r_{i+1} \in I^o$ and $\rho(\alpha[1,n])$ is an increasing function, also the image of r_{i+1} by $\rho(\alpha[1,n])$ is in the interior of the range: $r \in [\![\alpha[1,n]]\!]^o$. □

Lemma 2.13. *The signed binary representation is an open IFS representation.*

Proof. It follows from Lemma 2.5 that the digits of the signed binary representation contract strongly enough (cf. Definition 2.6 point (1)). We need to observe that every number $r \in [-1,1]^o$ is in the interior of the domain of some of the digits. Since these interiors are the open intervals $(-1,0)$, $(-\frac{1}{2},\frac{1}{2})$ and $(0,1)$, this condition is true. The rest follows from Lemma 2.7. □

3 Finite Transducers for IFS Representations

We consider only deterministic machines because we will restrict ourselves to the computation of single-valued functions. We also do not consider machines that can observe whether there is input available on a channel or not. Therefore, whenever a machine wants to read an input, it waits until it gets it.

Definition 3.1 (Finite transducer for IFS).
Let $\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_n$ be IFS representations. A *finite transducer* $\mathcal{T}$ of type

$$(\mathcal{S}_1, \ldots, \mathcal{S}_n) \to \mathcal{S}_0$$

is a tuple $(Q, \gamma, \zeta, q_i, Q_F)$ where

1. Q is a finite set of *states* including the *initial state* q_i and some (possibly zero) number of *looping states* $Q_F \subseteq Q$,

2. $\gamma\colon Q \to \wp(\{1,\ldots,n\})$ assigns a set of input *channels* to each state and satisfies $q \in Q_F \implies \gamma(q) = \emptyset$ and $q \in Q \setminus (\{q_i\} \cup Q_F) \implies \gamma(q) \neq \emptyset$, i.e. looping states are *without input* and, apart from the initial state, no other states can be without input.

3. $\zeta\colon \left(\bigcup_{q\in Q}(\{q\} \times \prod_{i\in\gamma(q)} D_{\mathcal{S}_i})\right) \to Q \times D^*_{\mathcal{S}_0}$ is the *transition and output function* of $\mathcal{T}$ which to each state and a set of digits read from the adequate channels assigns another state and a string of output digits. The two projections of ζ are denoted by ζ^{st} and ζ^{out}, respectively. The transition function has to satisfy

(a) $q \in Q_F \Longrightarrow \zeta^{st}(q, ()) = q$, i.e. the looping states loop,

(b) $\gamma(q_i) = \emptyset \Longrightarrow \zeta^{st}(\_, \_) \neq q_i$, i.e. the next state is always a state with input or a looping state.

Definition 3.2 (Underlying graph).
A finite transducer $\mathcal{T}$ naturally corresponds to the (oriented, pointed, edge and node labelled) graph $\mathcal{G}_{\mathcal{T}}$ whose nodes are the states Q, its root is the initial state, the nodes are labelled by the values of γ and there is an edge for each q and for each set of symbols $(x_i)_{i\in\gamma(q)}$ leading from q to $\zeta^{st}(q, (x_i)_{i\in\gamma(q)})$ and is labelled by the ":"-separated pair $(x_i)_{i\in\gamma(q)} : \zeta^{out}(q, (x_i)_{i\in\gamma(q)})$. This graph is called the *underlying graph* of $\mathcal{T}$.

The underlying graph can be used to illustrate finite transducers of smaller sizes. For some examples in this report, see Figure 4 below and Figure 5 on page 16.

Definition 3.3 (Reachable states).
A state of a finite transducer $\mathcal{T}$ is called *reachable* if there is a path to it from the initial state q_i in $\mathcal{G}_{\mathcal{T}}$.

The low level semantics of a transducer operates on streams of digits.

Definition 3.4 (Stream semantics).
A finite transducer $\mathcal{T} = (Q, \gamma, \zeta, q_i, Q_F)$ of type $(\mathcal{S}_1, \ldots, \mathcal{S}_n) \to \mathcal{S}_0$ *computes the stream function*

$$\sigma^{out}_{\mathcal{T}} \colon D^{\infty}_{\mathcal{S}_1} \times \cdots \times D^{\infty}_{\mathcal{S}_n} \to D^{\infty}_{\mathcal{S}_0}$$

which is defined (together with a *partial* state function $\sigma^{st}_{\mathcal{T}} \colon D^{*}_{\mathcal{S}_1} \times \cdots \times D^{*}_{\mathcal{S}_n} \to Q$) as follows:

1. $\sigma^{st}_{\mathcal{T}}(\epsilon, \ldots, \epsilon) = q_i, \quad \sigma^{out}_{\mathcal{T}}(\epsilon, \ldots, \epsilon) = \epsilon$

2. for each $u = (u_1, \ldots, u_n) \in D^{*}_{\mathcal{S}_1} \times \cdots \times D^{*}_{\mathcal{S}_n}$, such that $q = \sigma^{st}(u)$ is defined and $q \notin Q_F$, and for each $(x_1, \ldots, x_n)$ where every x_i is either a digit from $D_{\mathcal{S}_i}$ if $i \in \gamma(q)$ or $x_i = \epsilon$ otherwise, it holds

$$\sigma^{st}_{\mathcal{T}}(u_1x_1, \ldots, u_nx_n) = \zeta^{st}(q, (x_i)_{i\in\gamma(q)})$$

and (denoting $q' = \sigma^{st}_{\mathcal{T}}(u_1x_1, \ldots, u_nx_n)$)

$$\sigma^{out}_{\mathcal{T}}(u_1x_1, \ldots, u_nx_n) = \begin{cases} \sigma^{out}_{\mathcal{T}}(u) \cdot \zeta^{out}(q, (x_i)_{i\in\gamma(q)}) & \text{if } q' \notin Q_F \\ \sigma^{out}_{\mathcal{T}}(u) \cdot \zeta^{out}(q, (x_i)_{i\in\gamma(q)}) \cdot \zeta^{out}(q', ())^{\omega} & \text{if } q' \in Q_F \end{cases}$$

Now we will start viewing finite transducers via the meaning of the streams that they process.

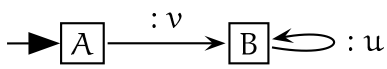

Figure 4: A transducer computing $[\![vuuu\ldots]\!]$.

Definition 3.5 (Semantics modulo representation).
A finite transducer $\mathcal{T} = (Q, \gamma, \zeta, q_i, Q_F)$ of type $(\mathcal{S}_1, \cdots, \mathcal{S}_n) \to \mathcal{S}_0$ *computes the partial multi-valued function*

$$f_{\mathcal{T}} \colon I_{\mathcal{S}_1} \times \cdots \times I_{\mathcal{S}_n} \rightrightarrows I_{\mathcal{S}_0}$$

which is defined by

$$f_{\mathcal{T}}(x_1, \ldots, x_n) = \left[\!\!\left[D^{\omega}_{\mathcal{S}_0} \cap \sigma^{\mathrm{out}}_{\mathcal{T}}([\![\{x_1\}]\!]^{-1}, \ldots, [\![\{x_n\}]\!]^{-1}) \right]\!\!\right] \tag{1}$$

where $\sigma^{\mathrm{out}}_{\mathcal{T}}$ and $[\![\cdot]\!]$ are extended to handle sets of values and $[\![\cdot]\!]^{-1}$ denotes the inverse of the set-valued $[\![\cdot]\!]$.

The transducer $\mathcal{T}$ is:

- *extensionally partial* if for each set of values x_i
  either $\sigma^{\mathrm{out}}_{\mathcal{T}}([\![\{x_1\}]\!]^{-1}, \ldots, [\![\{x_n\}]\!]^{-1}) \subseteq D^{\omega}_{\mathcal{S}_0}$ ($f_{\mathcal{T}}(x_1, \ldots, x_n)$ is defined)
  or $\sigma^{\mathrm{out}}_{\mathcal{T}}([\![\{x_1\}]\!]^{-1}, \ldots, [\![\{x_n\}]\!]^{-1}) \cap D^{\omega}_{\mathcal{S}_0} = \emptyset$ (it is undefined).
- *(extensionally) total* if it is extensionally partial and the function $f_{\mathcal{T}}$ is total.
- *extensional* if it is extensionally partial and the function $f_{\mathcal{T}}$ is single-valued.

The results of this report concern only with total and extensional transducers. That means those transducers which for every input that represents a certain given vector of values, return a representation of one and the same value.

Finitely computable real numbers (shortly finite numbers) are very easy to characterize. It has already been mentioned in the introduction that they are the numbers that have a periodic representation. We define them formally because we need some of their properties in Section 8.

Definition 3.6 (Finite numbers).
Let $\mathcal{S} = (I, D, \rho)$ be an IFS representation. The set of all constants $c \in I$ for which there is a finite transducer $\mathcal{T}$ of type $() \to \mathcal{S}$ with $f_{\mathcal{T}} = c$, is denoted *Fin*$_{\mathcal{S}}$.

Lemma 3.7 (Finite numbers are periodically representable).
For any $x \in \mathit{Fin}_{\mathcal{S}}$ *(for some IFS representation* $\mathcal{S}$*), there are* $u, v \in D^*_{\mathcal{S}}$ *such that* $x = [\![vuuu\ldots]\!]$*. On the other hand, for any* $v, u \in D^*_{\mathcal{S}}$*, it holds* $[\![vuuu\ldots]\!] \in \mathit{Fin}_{\mathcal{S}}$*.*

Proof. The first part follows from a well known fact that finite transducers without input can generate only periodical sequences. See, for example, [Trakhtenbrot & Barzdin 1973]. The number $[\![vuuu\dots]\!]$ can be computed by the two state transducer shown in Figure 4. ☐

Lemma 3.8 (Finite numbers are dense).
For any IFS representation $\mathcal{S}$, the set $Fin_{\mathcal{S}}$ is dense in $I_{\mathcal{S}}$.

Proof. Choose any open set $O \subset I_{\mathcal{S}}$ and by Lemma 2.8 get $u \in D^*_{\mathcal{S}}$ with $[\![u]\!] \subset O$. The finitely computable number $[\![uuu\dots]\!]$ is in O. ☐

The following lemma is a special case of the following observation: For any two finite transducers, there is a finite transducer that computes the composition of the functions computed by the first two (see [Konečný 2000]).

Lemma 3.9 (Instantiating an argument).
For any extensional and total finite transducer $\mathcal{T}$ of type $(\mathcal{S}_1, \dots, \mathcal{S}_n) \to \mathcal{S}_0$ any index $1 \leqslant k \leqslant n$ and a finitely computable number $x \in Fin_{\mathcal{S}_k}$, there is a finite transducer $\mathcal{T}'$ of type $(\mathcal{S}_1, \dots, \mathcal{S}_{k-1}, \mathcal{S}_{k+1}, \dots, \mathcal{S}_n) \to \mathcal{S}_0$, computing the function

$$f_{\mathcal{T}'}(x_1, \dots, x_{n-1}) = f_{\mathcal{T}}(x_1, \dots, x_{k-1}, x, x_k, \dots, x_{n-1}).$$

Proof. Let $\mathcal{T} = (Q, \gamma, \zeta, q_i, Q_F)$. The idea is that $\mathcal{T}'$ is a composition of $\mathcal{T}$ and a transducer which computes $[\![vuuu\cdots]\!]$ and outputs one digit in each transition. Formally, $\mathcal{T}' = (Q', \gamma', \zeta', q_i', Q_F')$ is defined as follows (without loss of generality, assume that $k = 1$ for technical reasons):

- $Q' = (Q \setminus Q_F) \times \{1, \dots, |vu|\} \cup Q_F \times \{1\}$
- $\gamma'(q, n) = \gamma(q) \setminus \{k\}$
- $\zeta^{\mathrm{out}\prime}((q, n), (\partial_i)_{i\in\gamma'(q,n)}) = \begin{cases} \zeta^{\mathrm{out}}(q, (\partial_i)_{i\in\gamma(q)}) & \text{for } k \notin \gamma(q) \\ \zeta^{\mathrm{out}}(q, (vu)[n], (\partial_i)_{i\in\gamma'(q,n)}) & \text{for } k \in \gamma(q) \end{cases}$
- $\zeta^{\mathrm{st}\prime}((q, n), (\partial_i)_{i\in\gamma'(q,n)}) = \begin{cases} (q', n) & \text{for } k \notin \gamma(q), q' \notin Q_F \\ (q', n+1) & \text{for } k \in \gamma(q), q' \notin Q_F, n < |vu| \\ (q', |v|+1) & \text{for } k \in \gamma(q), q' \notin Q_F, n = |vu| \\ (q', 1) & \text{for } q' \in Q_F \end{cases}$

  where $q' = \begin{cases} \zeta^{\mathrm{st}}(q, (\partial_i)_{i\in\gamma(q)}) & \text{for } k \notin \gamma(q) \\ \zeta^{\mathrm{st}}(q, (vu)[n], (\partial_i)_{i\in\gamma'(q,n)}) & \text{for } k \in \gamma(q) \end{cases}$
- $q_i' = (q_i, 1)$
- $Q_F' = Q_F \times \{1\}$

It is a matter of routine to check that the above is indeed a finite transducer, especially that $\gamma(q, n)$ is not empty except for the looping states and possibly the initial state, and that the looping states are really looping.

The stream function $\sigma^{\mathrm{out}}_{\mathcal{T}'}$ is identical to $\sigma^{\mathrm{out}}_{\mathcal{T}}$ with a fixed k-th argument equal to $vuuu\ldots$ From this all of the claims about $f_{T'}$ follow directly. $\square$

4 States are Functions

The main idea of this section is the following. During a computation of a function, a transducer moves from state to state while accepting input and producing output. In each state, one can abstract from the input and output made so far and imagine that the machine is starting afresh from this state. From this point of view, the transducer computes another function, which is derived from the original function and the input and output made so far.

Definition 4.1 (Changing the initial state).
Let $\mathcal{T} = (Q, \gamma, \zeta, q_i, Q_F)$ be a finite transducer of type $(\mathcal{S}_1, \ldots, \mathcal{S}_n) \to \mathcal{S}_0$. For every state $q \in Q$, define

$$\mathcal{T}^q = (Q^q, \gamma|_{Q^q}, \zeta|_{Q^q}, q, Q_F \cap Q^q)$$

where Q^q is the set of states reachable from q. The tuple $\mathcal{T}^q$ is a finite transducer and the functions $f_{\mathcal{T}^q}$, $\sigma^{\mathrm{out}}_{\mathcal{T}^q}$ and $\sigma^{\mathrm{st}}_{\mathcal{T}^q}$ are said to be *computed by* $\mathcal{T}$ *from state* q. Let us put $f^q_{\mathcal{T}} = f_{\mathcal{T}^q}$ for typographical reasons.

Notice that if $\mathcal{T}$ is extensionally partial, total or extensional, so is $\mathcal{T}^q$ for any *reachable* state $q \in Q_{\mathcal{T}}$.

Proposition 4.2 (States represent functions).
Let $\mathcal{T}$ be an extensional and total finite transducer of type $(\mathcal{S}_1, \ldots, \mathcal{S}_n) \to \mathcal{S}_0$. For every tuple of inputs $u = (u_1, \ldots, u_n) \in \mathrm{Dom}(\sigma^{\mathrm{st}}_{\mathcal{T}})$ and the associated state $q = \sigma^{\mathrm{st}}_{\mathcal{T}}(u)$ such that the output $v = \sigma^{\mathrm{out}}_{\mathcal{T}}(u)$ is a finite sequence, it holds:

$$f^q_{\mathcal{T}} = \rho_{\mathcal{S}_0}(v)^{-1} \circ f_{\mathcal{T}} \circ (\rho_{\mathcal{S}_1}(u_1), \ldots, \rho_{\mathcal{S}_n}(u_n)). \tag{2}$$

(Although the inverse $\rho_{\mathcal{S}_0}(v)^{-1}$ is partial, the composition is total.)

Proof. Suppose $f^q_{\mathcal{T}}(x_1, \ldots, x_n) = y$ and take some representations $[\![\alpha_i]\!] = \{x_i\}$ for $i = 1, \ldots, n$. It holds $[\![u_i\alpha_i]\!] = \rho_{\mathcal{S}_i}(u_i)([\![\alpha_i]\!]) = \rho_{\mathcal{S}_i}(u_i)(x_i)$. Using the representations $u_i\alpha_i$, we get that the value of $f_{\mathcal{T}}(\rho_{\mathcal{S}_1}(u_1)(x_1), \ldots, \rho_{\mathcal{S}_n}(u_n)(x_n))$ is $\rho_{\mathcal{S}_0}(v)(y)$, i.e.

$$\left(\rho_{\mathcal{S}_0}(v)^{-1} \circ f_{\mathcal{T}} \circ (\rho_{\mathcal{S}_1}(u_1), \ldots, \rho_{\mathcal{S}_n}(u_n))\right)(x_1, \ldots, x_n) = y.$$

$\square$

On the other hand, we will show that a transducer in which we can associate a function with each of its states, really computes those functions. The only extra condition for this to be true is that the transducer cannot loop in a cycle without producing any output.

Proposition 4.3 (Constructing a transducer from functions).
Let $\mathcal{T} = (Q, \gamma, \zeta, q_i, Q_F)$ be a finite transducer of type $(\mathcal{S}_1, \dots, \mathcal{S}_n) \to \mathcal{S}_0$ such that for each $q \in Q$ and each $(u_1, \dots, u_n) \in D^*_{\mathcal{S}_1} \times \dots \times D^*_{\mathcal{S}_n}$

$$\left(\sigma^{\mathrm{st}}_{\mathcal{T}^q}(u_1, \dots, u_n) = q, (u_1, \dots, u_n) \neq (\epsilon, \dots, \epsilon)\right) \Longrightarrow \sigma^{\mathrm{out}}_{\mathcal{T}^q}(u_1, \dots, u_n) \neq \epsilon \tag{3}$$

and $h\colon Q \to (I_{\mathcal{S}_1} \times \dots \times I_{\mathcal{S}_n} \to I_{\mathcal{S}_0})$ an assignment of (single-valued total) functions to states such that for each $q \in Q$ and $\eth_i \in D_{\mathcal{S}_i}$, for $i \in \gamma(q)$, it holds

$$h\left(\zeta^{\mathrm{st}}(q, (\eth_i)_{i\in\gamma(q)})\right) = \rho\left(\zeta^{\mathrm{out}}(q, (\eth_i)_{i\in\gamma(q)})\right)^{-1} \circ h(q) \circ (g_1, \dots, g_n) \tag{4}$$

where $g_i = \begin{cases} \rho(\eth_i) & \textit{for } i \in \gamma(q), \\ \mathrm{Id}_{I_{\mathcal{S}_i}} & \textit{otherwise.} \end{cases}$

Then $\mathcal{T}^q$ is extensionally partial and $f^q_{\mathcal{T}} = h(q)$ for every $q \in Q$. Especially, $f_{\mathcal{T}} = h(q_i)$ and $\mathcal{T}$ is total and extensional.

Proof. Fix a state $q \in Q$. It is sufficient to prove that for every tuple

$$(\alpha_1, \dots, \alpha_n) \in D^\omega_{\mathcal{S}_1} \times \dots \times D^\omega_{\mathcal{S}_n},$$

the output $\beta = \sigma^{\mathrm{out}}_{\mathcal{T}^q}(\alpha_1, \dots, \alpha_n)$ is infinite and it holds

$$h(q)(\llbracket\alpha_1\rrbracket, \dots, \llbracket\alpha_n\rrbracket) = \llbracket\beta\rrbracket\,. \tag{5}$$

For natural numbers $k_1, \dots, k_n > 0$, put $\alpha_{k_1,\dots,k_n} \equiv (\alpha_1[1, k_1], \dots, \alpha_n[1, k_n])$.

For any $k_1, \dots, k_n > 0$ such that $\alpha_{k_1,\dots,k_n} \in \mathrm{Dom}(\sigma^{\mathrm{st}}_{\mathcal{T}^q})$, by successive application of (4), we obtain

$$h\left(\sigma^{\mathrm{st}}_{\mathcal{T}^q}(\alpha_{k_1,\dots,k_n})\right) = \rho\left(\sigma^{\mathrm{out}}_{\mathcal{T}^q}(\alpha_{k_1,\dots,k_n})\right)^{-1} \circ h(q) \circ \left(\rho(\alpha_1[1, k_1]), \dots, \rho(\alpha_n[1, k_n])\right) \tag{6}$$

For the sake of contradiction, suppose that β is finite. If there would exist $k_1, \dots, k_n > 0$ with $\sigma^{\mathrm{st}}_{\mathcal{T}^q}(\alpha_{k_1,\dots,k_n}) \in Q_F$, then $\sigma^{\mathrm{out}}_{\mathcal{T}^q}(\alpha_{k_1,\dots,k_n})$ would be infinite because of the condition that output is not empty in any cycle. This would mean that also $\sigma^{\mathrm{out}}_{\mathcal{T}^q}(\alpha_1, \dots, \alpha_n) = \beta$ would be infinite, a contradiction.

Suppose then that such $k_1, \dots, k_n$ do not exist, i.e. with the input $(\alpha_1, \dots, \alpha_n)$, transducer $\mathcal{T}$ avoids all looping states. Now we get that, because $\mathcal{T}$ is finite, there have to exist two sets of indices $k_1, \dots, k_n > 0$ and $k'_1 \geqslant k_1, \dots, k'_n \geqslant k_n$ with $k'_i > k_i$ for at least one i, such that

$$\sigma^{\mathrm{st}}_{\mathcal{T}^q}(\alpha_{k_1,\dots,k_n}) = \sigma^{\mathrm{st}}_{\mathcal{T}^q}(\alpha_{k'_1,\dots,k'_n}) \quad \text{and} \quad \sigma^{\mathrm{out}}_{\mathcal{T}^q}(\alpha_{k_1,\dots,k_n}) = \sigma^{\mathrm{out}}_{\mathcal{T}^q}(\alpha_{k'_1,\dots,k'_n}).$$

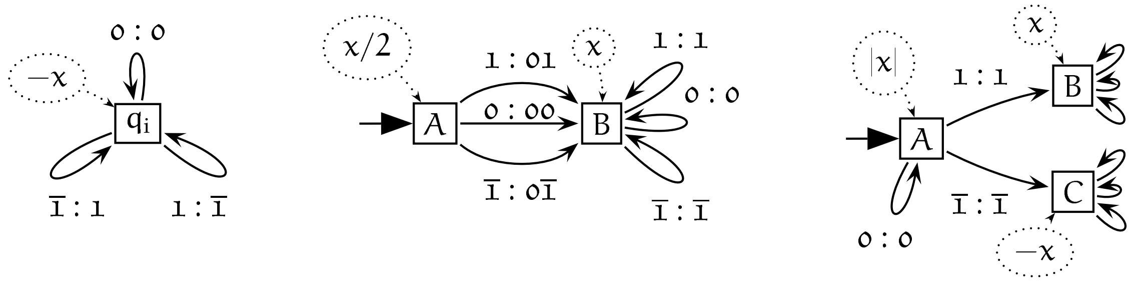

Figure 5: Examples of simple unary finite transducers with signed binary

Again, this means that there is a cycle without output, this time from the state $q' = \sigma^{st}_{\mathcal{T}^q}(\alpha_{k_1,\ldots,k_n})$:

$$\sigma^{st}_{\mathcal{T}^{q'}}(\alpha_1[k_1+1, k_1'], \ldots, \alpha_n[k_n+1, k_n']) = q'$$
$$\text{and } \sigma^{out}_{\mathcal{T}^{q'}}(\alpha_1[k_1+1, k_1'], \ldots, \alpha_n[k_n+1, k_n']) = \epsilon$$

which is a contradiction.

There is still equation (5) left to be proved. Equation (6) implies that the range of the function $h(q)$, when restricted to the domain $(\llbracket \alpha_1[1,k_1] \rrbracket, \ldots, \llbracket \alpha_n[1,k_n] \rrbracket)$, is within $\llbracket \sigma^{out}_{\mathcal{T}^q}(\alpha_{k_1,\ldots,k_n}) \rrbracket$. When $\mathcal{T}$ computes with the input $(\alpha_1, \ldots, \alpha_n)$, the output β is infinite and thus $\llbracket \sigma^{out}_{\mathcal{T}^q}(\alpha_{k_1,\ldots,k_n}) \rrbracket$ tends to the singleton set $\llbracket \beta \rrbracket$. Thus, in the limit case, equation (6) results in equation (5). $\square$

This proposition can be applied for constructing and checking finite transducers that compute certain real number functions. For example, in Figure 5, there are three simple unary finite transducers using signed binary for both input and output. They are drawn via their underlying graphs. The initial state is marked by a bold arrow. Each state is associated with a function in such a way that the conditions of Proposition 4.3 are satisfied. This fact and Proposition 4.3 prove that the transducers really compute the functions that they are assigned with, i.e. the first one computes negation, the second one $x/2$ and the third one the absolute value. We have omitted the labels on the looping arrows on states B and C in the last transducer as they had already been described in the previous two transducers.

5 Piecewise Affine Functions

These are continuous functions whose graph is composed of a finite number of n-dimensional polygons (see Figure 6). The aim of this section is to prove that whenever such a function f is of type $[-1,1]^n \to [-1,1]$ and the vertices of the polygons have rational coordinates (equivalently: the coefficients of the affine functions are rational), then there is a finite transducer computing f.

Figure 6: The graphs of two piecewise affine functions.

Definition 5.1 (Piecewise affine function).
A continuous function $f\colon I_1 \times \cdots \times I_n \to I_0$ is called *piecewise affine* if there exists a finite set K of n-dimensional polygons such that

$$(\forall A, B \in K)(A^o \cap B^o = \emptyset) \text{ and } \bigcup_{A \in K} A = I_1 \times \cdots \times I_n$$

and f is affine on each $A \in K$.

Any piecewise affine function f *with rational coefficients* can be also denoted more specifically as follows:

$$f = \mathit{Aff}\left(K, \kappa, (a_i^{(j)})_{i \leqslant n}^{j \leqslant m}, (b^{(j)})^{j \leqslant m}, c\right) \tag{7}$$

where $m > 0$, $\kappa\colon K \to \{1, \ldots, m\}$, $a_i^{(j)}, b^{(j)} \in \mathbb{Z}$ for $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m$, and $c \in \mathbb{N}$, $c > 0$, such that for every $C \in K$ it holds

$$f(x_1, \ldots, x_n) = \frac{a_1^{(\kappa(C))} x_1 + \cdots + a_n^{(\kappa(C))} x_n + b^{(\kappa(C))}}{c} \tag{8}$$

for all $(x_1, \ldots, x_n) \in C$.

Sometimes by specifying the affine functions within a piecewise-affine function f, the function f is not fully determined. Nevertheless, there are only finitely many functions satisfying such a description.

Lemma 5.2 (Coefficients almost determine piecewise affine function).
Let $I_1, \ldots, I_n \in \mathcal{I}$. For every set of numbers

- $m \in \mathbb{N}$, $m > 0$,
- $a_i^{(j)}, b^{(j)} \in \mathbb{Z}$ *for each* $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m$, *and*
- $c \in \mathbb{N}$, $c > 0$,

there are finitely many piecewise affine functions of the form (7) for some K and κ.

Proof. There are altogether m possible affine functions that can be assigned to the polygons in K, let us denote them as follows for $j = 1, \dots, m$:

$$A_j(x_1, \dots, x_n) = \frac{a_1^{(j)} x_1 + \dots + a_n^{(j)} x_n + b^{(j)}}{c}.$$

Let us assume, without loss of generality, that these m functions are pairwise different. Let H_{j_1, j_2} be the set of solutions of the equation $A_{j_1} = A_{j_2}$. It is either a hyperplane in $\mathbb{R}^n$ or empty.

Any hyperplane H in $\mathbb{R}^n$ which forms the border of two adjacent polygons in $C_1, C_2 \in K$ has the property that the affine functions $f|_{C_1}$ and $f|_{C_2}$ agree on H. Therefore H has to be identical to $H_{\kappa(C_1), \kappa(C_2)}$.

It follows that the boundary of each $C \in K$ is formed solely by the hyperplanes among H_{j_1, j_2} and by the boundary hyperplanes of the interval $I = I_1 \times \dots \times I_n$. Therefore C is a union of some of the sections of $\mathbb{R}^n$ cut out by all of the hyperplanes H_{j_1, j_2} within the interval I. Since there are not more than 2^m of these sections by Lemma 5.3 below and there are m affine functions to choose from in each section, there are not more than $m^{(2^m)}$ piecewise affine functions composed of pieces of the affine functions $A_1, \dots, A_m$. $\square$

Lemma 5.3. *In $\mathbb{R}^n$, any m hyperplanes divide the space to at most 2^m sections.*

Proof. Let us prove the claim by induction on m. With $m = 0$ there is exactly one section, i.e. the whole space $\mathbb{R}^n$. Thus the claim holds for $m = 0$. For $m > 0$, pick one of the hyperplanes H. By the induction hypothesis, the remaining $m - 1$ hyperplanes divide $\mathbb{R}^n$ to at most 2^{m-1} sections. The hyperplane H can divide each of these convex sections into at most two. Thus altogether the m hyperplanes divide $\mathbb{R}^n$ into at most $2 \cdot 2^{m-1} = 2^m$ sections. $\square$

Another technical lemma gives a limit on the b coefficients and on the total difference of the function, provided that there is a limit on the a coefficients.

Lemma 5.4. *For any piecewise affine function*

$$f = \mathit{Aff}\left(K, \kappa, (a_i^{(j)})_{i \leqslant n}^{j \leqslant m}, (b^{(j)})^{j \leqslant m}, c\right) : [-1, 1]^n \to [-1, 1]$$

with $\left|a_i^{(j)}\right| \leqslant c/4n$ *for all* i, j, *it holds*

1. *for every* $C \in K$, $\left|b^{(\kappa(C))}\right| \leqslant c \cdot \frac{5}{4}$
2. *for any* $\mathfrak{d}_1, \dots, \mathfrak{d}_n \in \{\bar{\mathfrak{1}}, \mathfrak{o}, \mathfrak{1}\}$ *and* $x, y \in ([\![\mathfrak{d}_1]\!], \dots, [\![\mathfrak{d}_n]\!])$ $|f(x) - f(y)| \leqslant \frac{1}{2}$.

Proof. Take any point $(x_1, \dots, x_n) \in C \in K$ and put $j = \kappa(C)$. It holds

$$\left|a_1^{(j)} x_1 + \cdots + a_n^{(j)} x_n + b^{(j)}\right| \leqslant c$$

because $f(x_1, \dots, x_n) \in [-1, 1]$. Applying $\left|a_i^{(j)}\right| \leqslant c/4n$, $|x_i| \leqslant 1$ and the triangular inequality, we get

$$c \;\geqslant\; \left|a_1^{(j)} x_1 + \cdots + a_n^{(j)} x_n + b^{(j)}\right| \;\geqslant\; \left|b^{(j)}\right| - \left|-a_1^{(j)}\right| - \cdots - \left|-a_n^{(j)}\right| \;\geqslant\; \left|b^{(j)}\right| - c/4$$

which yields $\left|b^{(j)}\right| \leqslant c + c/4 = c \cdot \frac{5}{4}$.

In order to prove the second statement, consider the points

$$z^{(i)} = (x_1, \dots, x_i, y_{i+1}, \dots, y_n)$$

for all $i = 0, \dots, n$. It holds $x = z^{(0)}$ and $y = z^{(n)}$. We will show that $\left|f(z^{(i-1)}) - f(z^{(i)})\right| \leqslant 1/2n$ for all $i = 1, \dots, n$, from which the desired inequality will follow by the triangular inequality:

$$|f(x) - f(y)| = \left|\sum_{1 \leqslant i \leqslant n} f(z^{(i-1)}) - f(z^{(i)})\right| \leqslant \sum_{1 \leqslant i \leqslant n} \left|f(z^{(i-1)}) - f(z^{(i)})\right| \leqslant \frac{1}{2}.$$

Consider the line connecting $z^{(i-1)}$ and $z^{(i)}$:

$$\{(x_1, \dots, x_{i-1}, t, y_{i+1}, \dots, y_n) \mid x_i \leqslant t \leqslant y_i\}$$

where, without loss of generality, we assume $x_i \leqslant y_i$. Denote all of the points where this line leaves (going from x_i to y_i) some polygon in K by $p_1, \dots, p_{s-1}$ and let $p_0 = z^{(i-1)}$, $p_s = z^{(i)}$. Also let $\beta\colon \{0, \dots, s\} \to \{1, \dots, m\}$ be the function assigning to each segment of this line an affine function with which f coincides in this segment. Denote by t_ℓ the i-th component of each p_ℓ for all $\ell = 0, \dots, n$. Now we can get what we need by triangular inequality and $|t_{\ell-1} - t_\ell| \leqslant 2$:

$$\left|f(z^{(i-1)}) - f(z^{(i)})\right| \leqslant \sum_{1 \leqslant \ell \leqslant s} |f(p_{\ell-1}) - f(p_\ell)| =$$

$$\sum_{1 \leqslant \ell \leqslant s} \left(\frac{\left|a_i^{(\beta(\ell))}\right|}{c} \cdot |t_{\ell-1} - t_\ell|\right) \leqslant \frac{1}{4n} \cdot \sum_{1 \leqslant \ell \leqslant s} |t_{\ell-1} - t_\ell| \leqslant \frac{1}{2n}.$$

$\square$

Theorem 5.5 (Signed binary computes piecewise affine functions).
For every piecewise affine function $\widehat{f}$ with rational coefficients, it holds $\widehat{f} = f_{\mathcal{T}}$ for some finite transducer $\mathcal{T}\colon (\mathcal{S}_{\text{sigbin}}, \dots, \mathcal{S}_{\text{sigbin}}) \to \mathcal{S}_{\text{sigbin}}$.

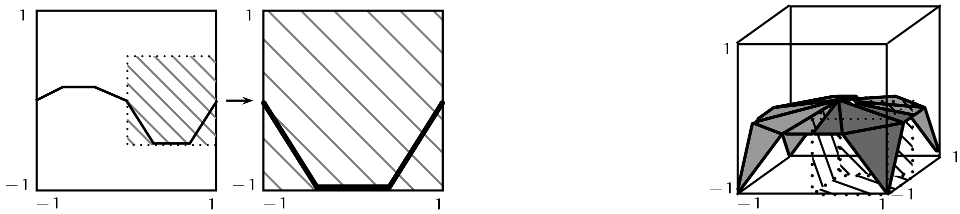

Figure 7: Clipping and enlarging piecewise affine functions during a transition.

Proof. Let $\mathit{Aff}\left(\widehat{K}, \widehat{\kappa}, (\widehat{a}_i^{(j)})_{i\leqslant n}^{j\leqslant m}, (\widehat{b}^{(j)})^{j\leqslant m}, \widehat{c}\right)$ be a more specific name for $\widehat{f}$. Define and choose $k \in \mathbb{N}$, $k > 0$ such that $2^{-k} \cdot |\widehat{a}_i^{(j)}/\widehat{c}| < \frac{1}{4n}$ for every i, j.

Now we can define $\mathcal{T} = (Q, \gamma, \zeta, q_i, Q_F)$ as follows:

- $Q = Q_1 \cup Q_2$ where

$$Q_1 = \left\{\widehat{f} \circ (\rho(u_1), \dots, \rho(u_n)) \,\middle|\, u_1, \dots, u_n \in D_{\text{sigbin}}^{k'},\ 0 \leqslant k' < k\right\}$$
$$Q_2 = \left\{\mathit{Aff}\left(K, \kappa, (\widehat{a}_i^{(j)})_{i\leqslant n}^{j\leqslant m}, (b^{(j)})_{j\leqslant m}, 2^k \cdot \widehat{c}\right) \,\middle|\, |b^{(j)}| \leqslant \tfrac{5}{4} \cdot 2^k \cdot \widehat{c}\right\}$$

- $q_i = \widehat{f}$,
- $Q_F = \emptyset$,
- $\gamma(q) = \{1, \dots, n\}$ for all $q \in Q$,
- $\zeta^{\text{out}}(f, (\mathfrak{d}_1, \dots, \mathfrak{d}_n)) = \begin{cases} \epsilon & \text{if } f \notin Q_2 \\ \mathfrak{d} \text{ with } [\![\mathfrak{d}]\!] \supseteq f([\![\mathfrak{d}_1]\!], \dots, [\![\mathfrak{d}_n]\!]) & \text{if } f \in Q_2 \end{cases}$,
- $\zeta^{\text{st}}(f, (\mathfrak{d}_1, \dots, \mathfrak{d}_n)) = \rho(\zeta^{\text{out}}(f, (\mathfrak{d}_1, \dots, \mathfrak{d}_n)))^{-1} \circ f \circ (\rho(\mathfrak{d}_1), \dots, \rho(\mathfrak{d}_n))$

We need to prove that the above tuple is a finite transducer and that it fulfills the premises of Proposition 4.3 to conclude that it computes $\widehat{f}$.

First of all, Q_2 is finite by Lemma 5.2 applied on each set of $b^{(j)}$'s. The initial state is defined correctly because $\widehat{f} \in Q_1$ with $k' = 0 < k$. Also ζ^{out} is defined correctly even for $f \notin Q_1$ because by Lemma 5.4 for any $x, y \in ([\![\mathfrak{d}_1]\!], \dots, [\![\mathfrak{d}_n]\!])$, it holds $|f(x) - f(y)| \leqslant \frac{1}{2}$. From this we can deduce that $|f([\![\mathfrak{d}_1]\!], \dots, [\![\mathfrak{d}_n]\!])| \leqslant \frac{1}{2}$ and therefore there is a digit $\mathfrak{d} \in \{\bar{1}, 0, 1\}$ which covers the whole range $f([\![\mathfrak{d}_1]\!], \dots, [\![\mathfrak{d}_n]\!])$.

For the same reason, the composition in the definition of $\zeta^{\text{st}}(f, \vec{\mathfrak{d}})$ is a total function, the graph of which is an enlarged and clipped version of the graph of f (see Figure 7 for an illustration).

Next we will show by induction that any reachable $f \in Q$ is of the form:

$$f = \mathit{Aff}\left(K, \kappa, (\widehat{a}_i^{(j)})_{i \leqslant n}^{j \leqslant m}, b^{(j)}, 2^{\ell} \cdot \widehat{c}\right) \text{ where } 0 \leqslant \ell \leqslant k.$$

This obviously holds for $\widehat{f}$ with $K = \widehat{K}$, $\kappa = \widehat{\kappa}$, $b^{(j)} = \widehat{b}^{(j)}$ and $\ell = 0$. If this is true for f, we will show that it is also true for any successor $f' = \zeta^{st}(f, \vec{\mathfrak{d}})$ (f' is not the derivative of f here). That means that for some K', κ', $(b')^{(j)}$ and ℓ'

$$f' = \mathit{Aff}\left(K', \kappa', (a_i^{(j)})_{i \leqslant n}^{j \leqslant m}, ({b'}^{(j)})^{j \leqslant m}, 2^{\ell'} \cdot c\right).$$

We need to observe that an intersection of an n-dimensional rectangle (in this case $[\![\mathfrak{d}_1]\!] \times \cdots \times [\![\mathfrak{d}_n]\!]$) and a polygon is a (possibly empty) set of polygons. Therefore let K' be the version of K clipped and enlarged by $(\rho(\mathfrak{d}_1), \ldots, \rho(\mathfrak{d}_n))$.

On each polygon $C' \in K'$ which is an enlarged section of an old polygon $C \in K$, we can put $\kappa'(C') = \kappa(C)$ because either

$$f'(x_1, \ldots, x_n) = \frac{\widehat{a}_1^{(\kappa(C))}\left(\frac{x_1 + \mathfrak{d}_1}{2}\right) + \cdots + \widehat{a}_n^{(\kappa(C))}\left(\frac{x_n + \mathfrak{d}_n}{2}\right) + b^{(\kappa(C))}}{\widehat{c}}$$

if $\zeta^{out}(f, \vec{\mathfrak{d}}) = \epsilon$, i.e. $f \in Q_1$, or

$$f'(x_1, \ldots, x_n) = 2 \cdot \frac{\widehat{a}_1^{(\kappa(C))}\left(\frac{x_1 + \mathfrak{d}_1}{2}\right) + \cdots + \widehat{a}_n^{(\kappa(C))}\left(\frac{x_n + \mathfrak{d}_n}{2}\right) + b^{(\kappa(C))}}{\widehat{c}} - \mathfrak{d}$$

if $\mathfrak{d} = \zeta^{out}(f, \vec{\mathfrak{d}}) \in \{\bar{1}, 0, 1\}$ (the digits from $\{\bar{1}, 0, 1\}$ are interpreted as $-1, 0, 1$ respectively), i.e. the coefficients of $x_1, \ldots, x_n$ do not change. We could also derive the formula for $|{b'}^{(\kappa(C'))}|$ and check that it is not bigger than $\frac{5}{4} \cdot 2^k \cdot \widehat{c}$ in case $\ell' = k$. But this follows from Lemma 5.4.

We have finished the proof that $\mathcal{T}$ is a finite transducer. In order to be able to apply Proposition 4.3 with h assigning to each state itself as a function, we need to show that on any cycle in $\mathcal{T}$ there is some output. The only transitions that do not produce output are those leading from states in $Q_1 \setminus Q_2$. There cannot be a cycle using only states from Q_1 because from a state $q \in Q_1$, independently on what the input is, the transducer $\mathcal{T}$ proceeds to a state in Q_2 within at most k steps. The other condition of Proposition 4.3 for $\mathcal{T}$ follows directly from the definition of ζ^{st}. Thus we can conclude that $\mathcal{T}$ computes $\widehat{f}$. $\square$

6 Strongly Connected Unary Affine Transducers

An important special case of Proposition 4.2 is that which corresponds to a cycle in a transducer. Equation (2) is then of the following recursive form.

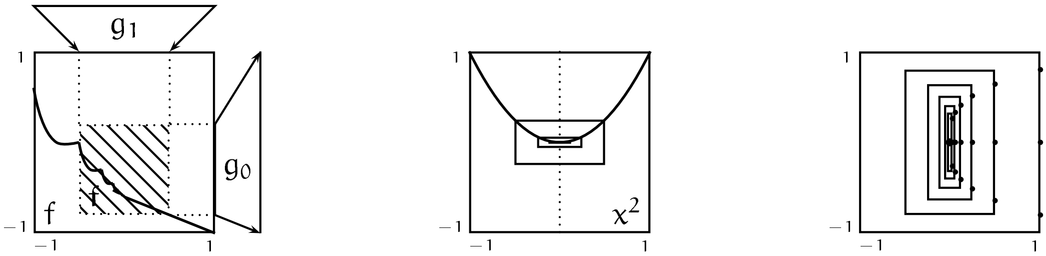

Figure 8: Affine sub-self-similarity of a function & strength mismatch

Definition 6.1 (Sub-self-similarity).
Let $I_0, I_1 \in \mathcal{I}$, $g_i \in \mathrm{dContr}_{I_i}$ for $i = 0, 1$ and $f\colon I_1 \to I_0$ such that

$$g_0 \circ f = f \circ g_1. \tag{9}$$

This equality is called a *sub-self-similarity*[6] of f.

When both d-contractions g_1 and g_0 are affine, the sub-self-similarity (9) can be visualized like in Figure 8 on the left.

Sub-self-similarity has several important consequences. The first one is concerned with the ratio of convergence speeds. At this point we start to restrict ourselves to affine d-contractions only. Although a similar lemma for arbitrary d-contractions holds, it requires a slightly more complicated proof. We do not include it here because we do not need it in this report. The main ideas of the following proof are visualized in Figure 8 on the right.

Lemma 6.2 (Strength mismatch in affine sub-self-similarity).
In the situation of the previous definition, it holds $f(\mathrm{Fix}_{g_1}) = \mathrm{Fix}_{g_0}$ and if both of the contractions g_i are affine, it holds:

1. *If $\mathrm{Strg}_{g_1} > \mathrm{Strg}_{g_0}$ (input is weaker than output), then the derivative $f'(\mathrm{Fix}_{g_1})$ exists and is equal to 0.*

2. *If $\mathrm{Strg}_{g_1} < \mathrm{Strg}_{g_0}$ (input is stronger than output), then the derivative $f'(\mathrm{Fix}_{g_1})$ either does not exist or exists and is equal to 0, $+\infty$ or $-\infty$.*

Proof. Denote $r_i = \mathrm{Fix}_{g_i}$. Applying the value r_1 to equation (9) of functions, we get $f(r_1) = g_0(f(r_1))$ which means $f(r_1) = \mathrm{Fix}_{g_0} = r_0$.

Suppose that g_0 and g_1 are affine.

From equation (9), we get $g_0^k \circ f = g_0^{k-1} \circ f \circ g_1 = \cdots = g_0 \circ f \circ g_1^{k-1} = f \circ g_1^k$.

Since g_0, g_1 are affine, it holds $\left|[\![g_i^k]\!]\right| = |I_i| \cdot w_i^k$ for every k, where $w_i = \mathrm{Strg}_{g_i}$. Similarly, $\max\{|x - r_i| \mid x \in [\![g_i^k]\!]\} = \max\{|x - r_i| \mid x \in I_i\} \cdot w_i^k$.

[6] We do not call it self-similarity to avoid confusion with self-similarity of fractals.

1. Assume $w_1 > w_0$. For any number $x \in I_1$, $x \neq r_1$ define k_x to be the index for which $x \in [\![g_1^{k_x}]\!]$ and at the same time $x \notin [\![g_1^{k_x+1}]\!]$. Take an arbitrary sequence $x_1, x_2, \dots$ of points in $I_1 \setminus \{r_1\}$ which converges to r_1. Notice that $\lim_{j\to\infty} k_{x_j} = \infty$. Now it holds

$$\lim_{j\to\infty} \frac{|f(x_j) - r_0|}{|x_j - r_1|} \leqslant \lim_{j\to\infty} \frac{\left|[\![g_0^{k_{x_j}}]\!]\right|}{\max\left\{|x - r_1| \,\middle|\, x \in [\![g_1^{k_{x_j}+1}]\!]\right\}} =$$

$$\lim_{j\to\infty} \frac{|I_0| \cdot w_0^{k_{x_j}}}{\max\{|x - r_1| \mid x \in I_1\} \cdot w_1^{k_{x_j}+1}} \leqslant C \cdot \lim_{j\to\infty} \left(\frac{w_0}{w_1}\right)^{k_{x_j}} = 0$$

where C is a constant. This proves that the derivative $f'(r_1)$ exists and is equal to 0.

2. Assume that $w_1 < w_0$ and that derivative $f'(r_1)$ exists. Denote the right hand side endpoint of I_1 by e_0 and let $e_k = g^k(e_0)$. Consider the sequence $f(e_k)$ of points from I_0. Since g_0 and g_1 are affine and $g_0^k \circ f = f \circ g_1^k$, it holds $e_k - r_1 = g_1^k(e_0) - r_1 = (e_0 - r_1) \cdot w_1^k$ and $f(e_k) - r_0 = g_0^k(f(e_0)) - r_0 = (f(e_0) - r_0) \cdot w_0^k$. There are 3 cases: $f(e_0) = r_0$, $f(e_0) > r_0$ and $f(e_0) < r_0$. In the first one, the following limit tends to 0, in the second one it tends to $+\infty$ and in the third one to $-\infty$:

$$\lim_{k\to\infty} \frac{f(e_k) - r_0}{e_k - r_1} = \lim_{k\to\infty} \frac{(f(e_0) - r_0) \cdot w_0^k}{(e_0 - r_1) \cdot w_1^k}$$

This implies that the derivative $f'(r_1)$ (under the assumption of its existence) is equal to 0, $+\infty$ or $-\infty$.

□

The following lemma allows us to extend the local property of 'having a non-zero derivative' to a global property of 'being affine'.

Lemma 6.3 (Affine sub-self-similarity implies affinity).
In the situation of Definition 6.1, if both of the contractions g_i are affine, the sub-self-similarity equation (9) implies that

1. *either* f *is not differentiable in* Fix_{g_1}
2. *or* $f'(\mathrm{Fix}_{g_1}) \in \{0, +\infty, -\infty\}$
3. *or* f *is affine, i.e. there exist constants* $a, b \in \mathbb{R}$ *such that* $f(x) = ax + b$ *for all* $x \in I_1$.

Proof. If $\mathrm{Strg}_{g_1} \neq \mathrm{Strg}_{g_0}$, we can apply Lemma 6.2 to conclude (1) or (2). Assume that $\mathrm{Strg}_{g_1} = \mathrm{Strg}_{g_0}$ for the rest of the proof and also suppose that the derivative $f'(\mathrm{Fix}_{g_1})$ exists.

Denote $r_i = \mathrm{Fix}_{g_i}$ and $w_i = \mathrm{Strg}_{g_i}$ for $i = 0, 1$. Since g_i are affine, it holds $g_i^k(x) - r_i = (x - r_i) \cdot w_i^k$ for every $x \in I_i$ and $k \in \mathbb{N}$.

Take an arbitrary point $x \in I_1$, $x \neq r_1$ and define $x_k = g_1^k(x)$ for all $k \in \mathbb{N}$. Now the value of the derivative can be expressed by

$$f'(r_1) = \lim_{k\to\infty} \frac{f(x_k) - r_0}{x_k - r_1} = \lim_{k\to\infty} \frac{(f(x) - r_0) \cdot w_0^k}{(x - r_1) \cdot w_1^k} = \frac{f(x) - r_0}{x - r_1}$$

out of which a formula for f can be extracted:

$$f(x) = f'(r_1)(x - r_1) + r_0$$

which, together with $f(r_1) = r_0$, proves that f is affine. □

Definition 6.4 (Strongly connected transducer).
A finite transducer $\mathcal{T}$ is *strongly connected (SC)* if its underlying graph $\mathcal{G}_\mathcal{T}$ is strongly connected.

SC transducers have the property that, to whatever state an input may lead to, there is a continuation of this input which would lead us back to the initial state. This means that we can find a cycle resulting in a sub-self-similarity in any open interval and thus prove the first restricted version of the main limitation theorem.

Proposition 6.5 (SC unary version of the main theorem).
Let $\mathcal{T}$ be a total and extensional SC transducer of type $(\mathcal{S}_1) \to \mathcal{S}_0$ using affine representations $\mathcal{S}_1, \mathcal{S}_0$. Either the function $f_\mathcal{T}$ is affine or there is a dense set $E \subseteq I_{\mathcal{S}_1}$ such that for each $x \in E$, it holds $f'(x) \in \{0, +\infty, -\infty\}$ whenever $f'(x)$ exists.

Proof. Let $C = \{u \in D_{\mathcal{S}_1}^* | \sigma_\mathcal{T}^{st}(u) = q_i\}$ be the set of input sequences that will cause the machine $\mathcal{T}$ to return to the initial state. For each $u \in C$, by applying equation (2) from Proposition 4.2, we derive the sub-self-similarity

$$\rho(\sigma_\mathcal{T}^{out}(u)) \circ f_\mathcal{T} = f_\mathcal{T} \circ \rho(u).$$

By Lemma 6.3, we get that either $f_\mathcal{T}$ is affine or its derivative $f'(\mathrm{Fix}_{\rho_{\mathcal{S}_1}(u)})$ does not exist or is equal to 0, $+\infty$ or $-\infty$. Denote

$$E = \{\mathrm{Fix}_{\rho_{\mathcal{S}_1}(u)} | u \in C\}.$$

All we need to prove in order to show that E has all of the desired properties is that it is dense in $I_{\mathcal{S}_1}$. Take any open interval $J \subseteq I_{\mathcal{S}_1}$. By Lemma 2.8, there is a sequence $u_1 \in D_{\mathcal{S}_1}^*$ such that $[\![u_1]\!] \subseteq J$. Since $\mathcal{T}$ is SC, there is another input sequence u_2 such that $u_1 u_2 \in C$. Therefore there is a number in E, namely $\mathrm{Fix}_{\rho_{\mathcal{S}_1}(u_1 u_2)}$, which is in J. □

Remark 6.6. The previous proposition would not hold without the condition of continuous differentiability or a similar smoothness condition. In [Konečný 2000], there is a 10-state unary transducer over the signed binary representation which has the following properties. It is SC, total and extensional. The function f that it computes is strictly increasing. There are two sets E_1, E_2, each of which is dense in $[-1, 1]$, such that $f'(x)$ exists for each $x \in E_1$ as well as for each $x \in E_2$ and it holds $x \in E_1 \implies f'(x) = 0$ and $x \in E_2 \implies f'(x) = +\infty$.

7 Unary Affine Transducers

How to extend Proposition 6.5 to transducers that are not SC? We will see that each transducer $\mathcal{T}$ has states q such that $\mathcal{T}^q$ is SC. Thus in any transducer, there are states to which Proposition 6.5 applies. In the following, we will prepare a method for proving that a property holds for each vertex of an oriented graph (i.e. for every state of a transducer, in our case) by induction. This will allow us to extend an essential part of Proposition 6.5 to arbitrary states of finite transducers. These ideas are borrowed from basic graph theory.

Definition 7.1 (Strongly connected components).
Let $\mathcal{G}$ be an oriented graph and $V_\mathcal{G}$ the set of its vertices. Let $q \to_\mathcal{G} q'$, for vertices $q, q' \in V_\mathcal{G}$, mean that there is an edge leading from q to q' in $\mathcal{G}$. Let $\to_\mathcal{G}^+$ be the transitive closure of $\to_\mathcal{G}$ and $\leftrightarrows_\mathcal{G}^*$ the equivalence relation generated from $\to_\mathcal{G}$. The expression $q \leftrightarrows_\mathcal{G}^* q'$ is pronounced "q *is strongly connected with* q'". The equivalence classes of $\leftrightarrows_\mathcal{G}^*$ are called the *strongly connected components* of $\mathcal{G}$. The $\leftrightarrows_\mathcal{G}^*$ class that contains $q \in V_\mathcal{G}$ is denoted by $[q]_{\leftrightarrows_\mathcal{G}^*}$.

Whenever there is no confusion about which graph they relate to, the subscript $\mathcal{G}$ can be left out from the symbols $V_\mathcal{G}$, $\to_\mathcal{G}$, $\to_\mathcal{G}^+$, $\leftrightarrows_\mathcal{G}^*$ as well as from other symbols relative to $\mathcal{G}$, defined later.

When factored by the equivalence of being strongly connected, a graph becomes acyclic. Therefore it is possible to define the depth of a vertex which can be roughly thought of as the distance from the associated root, as follows.

Definition 7.2 (Depth of a vertex).
For a vertex q of a finite oriented graph $\mathcal{G}$, let $\mathit{depth}_\mathcal{G}(q)$ denote the maximal number of $\leftrightarrows^*$ equivalence classes that the vertices on a path leading from q may belong to:

$$\mathit{depth}_\mathcal{G}(q) = \max\left\{\operatorname{card}\{[q_i]_{\leftrightarrows^*}\}_{i=1}^n \,\middle|\, n > 1,\ q_1, \ldots, q_n \in V,\ q = q_1, q_1 \to \ldots \to q_n\right\}$$

where max over the empty set is 0. For each $n \in \mathbb{N}$, define

$$SC_\mathcal{G}^n = \{q \in V \,|\, \mathit{depth}(q) \leqslant n\}$$

.

Remark 7.3 (Basic properties of depth).
In any finite oriented graph

1. the set SC^0 is equal to the set of vertices without any outgoing edge,
2. it holds: $SC^0 \subseteq SC^1 \subseteq SC^2 \subseteq \ldots$,
3. for any pair of vertices q, q' it holds:
   $q \to^+ q' \Longrightarrow q \leftrightarrows^* q'$ or $depth(q) > depth(q')$.
4. For every vertex q of a finite oriented graph it holds:
   $depth(q) = n > 1 \Longrightarrow (\exists q')(q \to^+ q'$ and $depth(q') < n)$

Lemma 7.4. *Let $\mathcal{G}$ be a nonempty finite oriented graph in which each vertex has at least one outgoing edge. Then it holds $SC^0 = \emptyset$ and $SC^1 \neq \emptyset$.*

Proof. $SC^0 = \emptyset$ follows from the previous remark, point (1). $SC^1 \neq \emptyset$ then follows from point (4) of the previous remark as follows: Let m be the smallest natural number for which $SC^m \neq \emptyset$. Such an m is well defined because $\mathcal{G}$ is finite and non-empty and therefore there exists a non-empty SC^k for some k. By point (4), this m has to be equal to 1 because otherwise we would find a vertex with depth less than m. ☐

Now we have all ingredients ready for proving the unary version of the main limitation theorem by induction on the depth of vertices in the underlying graph.

Proposition 7.5 (Unary version of Main Theorem).
Let $\mathcal{T}$ be a total and extensional transducer of type $(\mathcal{S}_1) \to \mathcal{S}_0$ using affine representations $\mathcal{S}_1, \mathcal{S}_0$. Moreover, let $\mathcal{S}_1$ be open. *If the function $f_\mathcal{T}$ is continuously differentiable on some open interval $J \subset I_{\mathcal{S}_1}$, then it is affine on J.*

Proof. We prove the proposition for every $f^q_\mathcal{T}$ instead of $f_\mathcal{T}$ only. We do it by induction on the depth of q in the underlying graph $\mathcal{G}_\mathcal{T}$.

For $depth(q) = 1$, the sub-transducer consisting states reachable from q is SC. We can apply Proposition 6.5 to get that either $f^q_\mathcal{T}$ is affine on the whole of $I_{\mathcal{S}_1}$ or there is a dense set E where the derivative of $f^q_\mathcal{T}$ either does not exist or is equal to 0, $+\infty$ or $-\infty$. The same is true for its subset $J \cap E$ except that the derivative has to exist everywhere. Assume that $f^q_\mathcal{T}$ is not affine. Since the 3 values 0, $+\infty$, $-\infty$ are separated from each other and $J \cap E$ is dense in J and $(f^q_\mathcal{T})'$ is continuous on J, $(f^q_\mathcal{T})'$ has to be constant on all of J. A derivative cannot be constantly infinite, therefore it has to be constantly 0 on J. This proves that $f^q_\mathcal{T}$ is affine on J.

Assume as the induction hypothesis that the proposition holds for all $f^q_\mathcal{T}$ for vertices q with $depth(q) < k$, i.e. that such a $f^q_\mathcal{T}$ is affine on any interval where it is differentiable, and take a q with $depth(q) = k$.

Let $U = \{u \in D_{\mathcal{S}_1}^* \mid [\![u]\!] \subseteq J\}$ and $F = \{\mathrm{Fix}_{\rho(u)} \mid u \in U\}$ be the set of all inputs which represent an interval within J and the set of their fixpoints. By Lemma 2.8, F is dense in J. If for all $x \in F$ it holds $(f_{\mathcal{J}}^q)'(x) \in \{0, +\infty, -\infty\}$, then we can conclude (similarly like above in the case of $q \in SC^1$) that the derivative $(f_{\mathcal{J}}^q)'$ is constantly 0 and thus $f_{\mathcal{J}}^q$ is constant on J. Assume, therefore, that there is $u \in U$ for which $(f_{\mathcal{J}}^q)'(\mathrm{Fix}_{\rho(u)}) \notin \{0, +\infty, -\infty\}$.

It follows from Remark 7.3, point (4) that for u, as well as for any other input, it holds either $\mathit{depth}(\sigma_{\mathcal{J}^q}^{st}(u)) < k$ or there is a v such that $\sigma_{\mathcal{J}^q}^{st}(uv) = q$. In the latter case, we can apply Lemma 6.3 to conclude that $f_{\mathcal{J}}^q$ is affine on the whole of $I_{\mathcal{S}_1}$. In the former case, we use the induction hypothesis for $q' = \sigma_{\mathcal{J}^q}^{st}(u)$ whose function $f_{\mathcal{J}}^{q'}$ is continuously differentiable on the whole of $I_{\mathcal{S}_1}$ by Proposition 4.2 and thus affine. By the same proposition we get that $f_{\mathcal{J}}^q$ is affine on $[\![u]\!]$, i.e. on some non-singleton interval $[a, b] = [\![u]\!]$.

This situation permits us to define

$$z = \min\left\{x \in \overline{J} \,\middle|\, f_{\mathcal{J}}^q|_{[x,b]} \text{ is affine}\right\} \text{ and } Z = \max\left\{x \in \overline{J} \,\middle|\, f_{\mathcal{J}}^q|_{[a,x]} \text{ is affine}\right\}$$

where $\overline{J}$ is the closure of J. This means that $[z, Z]$ is a maximal and non-singleton interval on which $f_{\mathcal{J}}^q$ is affine and moreover not constant. We will finish the proof by showing that $[z, Z] = \overline{J}$. This will follow from $z, Z \notin J$.

In order to pursue a contradiction, suppose that $z \in J$. Take an α with $[\![\alpha]\!] = \{z\}$ such that $z \in [\![\alpha[1, j]]\!]^o$ for each j by Lemma 2.12. (At this point we are using the assumption that S_1 is open.) Since J is open, there is an index j_0 such that $[\![\alpha[1, j]]\!] \subseteq J$ for all $j \geqslant j_0$. For each such a $j \geqslant j_0$, either $f_{\mathcal{J}}^q$ is affine on $[\![\alpha[1, k]]\!]$ or there is a point $x_j \in [\![\alpha[1, j]]\!]$ with $(f_{\mathcal{J}}^q)'(x_j) \in \{0, +\infty, -\infty\}$. In the former case, $f_{\mathcal{J}}^q$ would be affine on $[z, Z] \cup [\![\alpha[1, j]]\!]$ which is larger than $[z, Z]$ and thus contradicts the choice of z. If it is the latter case for all $j > j_0$, the sequence x_j converges to z and by continuity of derivative of $f_{\mathcal{J}}^q$ on J, its derivative in z would have to be 0, $+\infty$ or $-\infty$, which contradicts the fact that $f_{\mathcal{J}}^q$ is affine and non-constant on $[z, Z]$.

The proof that $Z \notin J$ is symmetrical. □

8 Multiple Input Affine Transducers

Finally, we are going to extend Proposition 7.5 to functions with multiple arguments. In order to enable us to reason about functions of more variables easier, let us define the following notation for instantiated functions:

Definition 8.1. Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be a function and $P \subseteq \{1, \ldots, n\}$ a set of argument indices. For any tuple of numbers $X = (x_i)_{i \notin P}$ denote

$$f_{P,X}\colon \mathbb{R}^{|P|} \to \mathbb{R} : (x_i)_{i \in P} \mapsto f(x_1, \ldots, x_n).$$

If $P = \{k\}$ is a singleton set, the function can be also denoted by $f_{k,X}$.

Unfortunately, the multi-dimensional limitation theorem does not follow from Proposition 7.5 alone because there are functions that are not affine but all of its unary instantiations are affine. For example, the binary function $(x, y) \mapsto x \cdot y$ is not affine but for each $t \in \mathbb{R}$ the functions $x \mapsto x \cdot t$ and $y \mapsto t \cdot y$ are affine. For this reason, we need to go back to sub-self-similarity and extend it to multiple dimensions.

Definition 8.2 (MD sub-self-similarity).
Let $I_0, I_1, \ldots, I_n \in \mathcal{I}$, $g_i \in \mathrm{dContr}_{I_i}$ for $i = 0, \ldots, n$, and $f\colon I_1 \times \cdots \times I_n \to I_0$ such that the following equality holds:

$$g_0 \circ f = f \circ (g_1, \ldots, g_n) \tag{10}$$

This equality is called an n-*dimensional sub-self-similarity* of f. The fixpoints Fix_{g_i} for $i = 0, \ldots, n$, in the context of the above sub-self-similarity, will be denoted by r_i and the strengths Strg_{g_i} by w_i. Denote also $\vec{r} = (r_1, \ldots, r_n)$ and $\vec{g} = (g_1, \ldots, g_n)$.

Lemma 8.3 (MD affine sub-self-similarity implies affinity).
In the situation of Definition 8.2, if all of the contractions g_i are affine, and if for some $P \subseteq \{1, \ldots, n\}$, the function $h = f_{P,(r_i)_{i \notin P}}$ is differentiable in $(r_i)_{i \in P}$ and for every $j \in P$, the partial derivative $h'_j((r_i)_{i \in P})$ is not zero, then there exist indices $a_i \in \mathbb{R}$ for $i \in P$, and $b \in \mathbb{R}$ such that

$$h((x_i)_{i \in P}) = b + \sum_{i \in P} a_i x_i$$

for all $(x_i)_{i \in P} \in \mathbb{R}^{|P|}$.

Proof. From the equation (10) a sub-self-similarity for h can be derived: $g_0 \circ h = h \circ (g_i)_{i \in P}$. Therefore it is sufficient to prove the lemma for $P = \{1, \ldots, n\}$ and $h = f$ since for a smaller P we may completely ignore the arguments which are not in P. Let us then assume that f is differentiable in $\vec{r}$ and all partial derivatives $p_k = f'_k(\vec{r})$ for $k = 1, \ldots, n$ are not zero.

First we will prove that $w_0 = w_1 = \cdots = w_n$. Suppose, on the contrary, that $w_0 \neq w_k$ for some $k \in \{1, \ldots, n\}$. Applying Lemma 6.2 to $f_k = f_{k,(r_i)_{i \neq k}}$ for which it holds $g_0 \circ f_k = f_k \circ g_k$, we get that $f'_k(\mathrm{Fix}_{g_k})$ is either 0 or infinite. Since $f'_k(r_k)$ is equal to the k-th partial derivative p_k, it cannot be 0. It cannot be infinite neither because the definition of differentiability in higher dimensions excludes it—a contradiction.

Pick an arbitrary point $\vec{x} = (x_1, \ldots, x_n) \in I_1 \times \cdots \times I_n$ and define the sequence $\vec{x}^{(k)} = \vec{g}^k(\vec{x})$ which converges to $\vec{r}$. Using the definition of the differential δf, the fact that $\delta f(\vec{r})(\vec{x}) = \vec{p} \cdot \vec{x}$ (denoting $\vec{p} = (p_1, \ldots, p_n)$) and that all $g_0, \ldots, g_n$ are affine contractions with the same strength, we get the equation:

$$\frac{\delta f(\vec{r})(\vec{x} - \vec{r})}{|\vec{x} - \vec{r}|} = \lim_{k \to \infty} \frac{f(\vec{x}^{(k)}) - f(\vec{r})}{|\vec{x}^{(k)} - \vec{r}|} = \lim_{k \to \infty} \frac{(f(\vec{x}) - f(\vec{r})) \cdot w_0^k}{|\vec{x} - \vec{r}| \cdot w_1^k} = \frac{f(\vec{x}) - f(\vec{r})}{|\vec{x} - \vec{r}|}$$

out of which an affine formula for f can be extracted:

$$f(\vec{x}) = \delta f(\vec{r})(\vec{x} - \vec{s}) + f(\vec{r}) = \vec{p} \cdot (\vec{x} - \vec{r}) + f(\vec{r}) = \\ p_1(x_1 - r_1) + \cdots + p_n(x_n - r_n) + f(\vec{r}).$$

$\square$

Now we will recall a simple and well-known characterization of the functions whose arbitrary unary instantiations are affine. They are exactly the polynomials of the following form:

Definition 8.4 (Multi-affine polynomials).
The n-variate polynomials over the variables $x_1, \ldots, x_n$ of the form

$$\sum_{P \subseteq \{1,\ldots,n\}} a_P \prod_{i \in P} x_i$$

where all a_P are real numbers, are called *multi-affine*.

Let us formally state and prove the characterization claim.

Lemma 8.5 (Detecting multi-affine polynomials).
Let $f \colon \mathbb{R}^n \to \mathbb{R}$ $(n > 0)$ be a function such that for every $j \in \{1, \ldots, n\}$ and every tuple $X = (x_i)_{i \neq j} \in \mathbb{R}^{n-1}$, the function $f_{j,X}$ is affine. Then f is a multi-affine polynomial.

Proof. Let us proceed by induction on n. For $n = 1$ the claim is trivial. Suppose it holds for all $n < k > 1$ and prove it for $n = k$. For each $x \in \mathbb{R}$ and $X \in \mathbb{R}^{n-1}$ define the functions g_x and h_X by $g_x(X) = h_X(x) = f(x, X)$. Notice that $h_X = f_{1,X}$. By induction hypothesis and the assumption on $f_{1,X}$ we get:

$$g_x(X) = \sum_{P \subseteq \{1,\ldots,n-1\}} a_P(x) \prod_{i \in P} x_i$$

$$h_X(x) = a(X) \cdot x + b(X)$$

where all a_P are functions of type $\mathbb{R} \to \mathbb{R}$ and a, b are functions of type $\mathbb{R}^{n-1} \to \mathbb{R}$. Since the two expressions above are equal for every x, X, by taking two different fixed $t, t' \in \mathbb{R}$, we get

$$a(X)(t - t') = \sum_{P \subseteq \{1,\ldots,n-1\}} (a_P(t) - a_P(t')) \prod_{i \in P} x_i$$

for every X, which means that a is a multi-affine polynomial. Now it follows that

$$b(X) = -a(X)t + \sum_{P \subseteq \{1,\ldots,n-1\}} a_P(t) \prod_{i \in P} x_i$$

is also a multi-affine polynomial. Altogether, $f(x, X) = h_X(x) = a(X) \cdot x + b(X)$ is a multi-affine polynomial. $\square$

We state a basic property of polynomials which we need in the following proof.

Lemma 8.6 (Polynomials are locally determinable).
If two n-variate real polynomials agree on an open interval $J_1 \times \cdots \times J_n \subseteq \mathbb{R}^n$, then they are equal.

The main limitation theorem first comes in a version where the unary interval J from Proposition 7.5 is replaced by a multi-dimensional interval. In its corollary, this interval is replaced by an arbitrary open connected set.

Theorem 8.7. *Let $\mathcal{T}$ be a total extensional transducer of type $(\mathcal{S}_1 \times \cdots \times \mathcal{S}_n) \to \mathcal{S}_0$ where all $\mathcal{S}_i$ are open affine representations. If the function $f_{\mathcal{T}}$ is continuously differentiable on some* open *interval $J = J_1 \times \cdots \times J_n \subset I_{\mathcal{S}_1} \times \cdots \times I_{\mathcal{S}_n}$, then it is affine on J.*

Proof. For any $k \in \{1, \ldots, n\}$ and any set of values $x_i \in \mathit{Fin}_{\mathcal{S}_i} \cap J_i$ for $i \neq k$, the restriction of $f = f_{\mathcal{T}}$ to the unary function $f_{k,(x_i)_{i \neq k}}$ is also finitely computable by Lemma 3.9. Therefore this unary function is affine on J_k. Since this holds for every $(x_i)_{i \neq k}$ in the dense set $\prod_{i \neq k} \mathit{Fin}_{\mathcal{S}_i} \cap J_i$, it holds for all values in $\prod_{i \neq k} J_i$. Thus by Lemma 8.5, f is a multi-affine polynomial on J.

Let us note which of the arguments of f are not redundant and define a name for the set of their indices as follows:

$$P = \Big\{ j \in \{1, \ldots, n\} \,\Big|\, \Big(\exists (x_1, \ldots, x_n) \in J, x'_j \in J_j\Big) \Big(f(x_1, \ldots, x_n) \neq f(x_1, \ldots, x'_j, \ldots, x_n)\Big)\Big\}.$$

Since the value of f within J does not depend on the value of an argument with an index $i \notin P$, the partial derivative f'_i is constantly zero on all of J. However, for an argument index $i \in P$, f'_i can be zero only on a very limited part of J since it is a non-zero polynomial. Actually, the set $Z_i = \{\vec{x} \in J \,|\, f'_i(\vec{x}) = 0\}$ is a closed set in J with empty interior by Lemma 8.6. Therefore the finite union $\bigcup_{i \in P} Z_i$ is also closed and with empty interior. Its complement C is an open non-empty subset of J where $f'_i(\vec{x})$ is not zero for every $i \in P$ and $\vec{x} \in C$.

Pick an open n-dimensional box $K_1 \times \cdots \times K_n$ within the open set C. By Lemma 2.8, there is a set of inputs $(u_1, \ldots, u_n) \in \mathrm{Dom}(\sigma^{st}_{\mathcal{T}})$ such that

$$[\![u_1]\!] \times \cdots \times [\![u_n]\!] \subset K_1 \times \cdots \times K_n \subseteq C.$$

Since $\mathcal{T}$ is finite there are another sets of inputs u'_i, u''_i such that there is a cycle and therefore a sub-self-similarity as follows:

$$\rho_{\mathcal{S}_0}(u''_0)^{-1} \circ f^q_{\mathcal{T}} = f^q_{\mathcal{T}} \circ (\rho_{\mathcal{S}_1}(u''_1), \ldots, \rho_{\mathcal{S}_n}(u''_n))$$

where $q = \sigma^{st}_{\mathcal{T}}(u_1u'_1, \ldots, u_nu'_n) = \sigma^{st}_{\mathcal{T}}(u_1u'_1u''_1, \ldots, u_nu'_nu''_n)$ and $u''_0 = \sigma^{out}_{\mathcal{T}^q}(u''_1, \ldots, u''_1)$. In this situation we can apply Lemma 8.3 with the set P, to get that $f^q_{\mathcal{T}}$ is affine on all of its domain. This can be translated to a part of the domain of f by the following equation which follows from Proposition 4.2:

$$f^q_{\mathcal{T}} = \rho_{\mathcal{S}_0}(\nu)^{-1} \circ f \circ (\rho_{\mathcal{S}_1}(u_1u'_1), \ldots, \rho_{\mathcal{S}_n}(u_nu'_n))$$

where $\nu = \sigma^{out}_{\mathcal{T}}(u_1u'_1, \ldots, u_nu'_n)$. Since all functions in this equation apart from f are known to be affine, f has to be affine on $[\![u_1u'_1]\!] \times \cdots \times [\![u_nu'_n]\!]$. By Lemma 8.6 and the previously established fact that f is a multi-linear polynomial on J, it follows that f is affine on J. $\square$

Corollary 8.8 (Main theorem for affine transducers).
Let $\mathcal{T}$ be a total extensional transducer of type $(\mathcal{S}_1 \times \cdots \times \mathcal{S}_n) \to \mathcal{S}_0$ where all $\mathcal{S}_i$ are open *affine representations. If the function $f_{\mathcal{T}}$ is continuously differentiable on some* open connected *set $J \subset I_{\mathcal{S}_1} \times \cdots \times I_{\mathcal{S}_n}$, then it is affine on J.*

Proof. By Theorem 8.7, $f|_L$ is affine on each open n-dimensional interval L within J. Such an interval surrounds every point in J since J is open. All we need to prove is that f coincides with the same affine function on every interval within J. Let $x, y \in J$ and $g\colon [0, 1] \to J$ be a continuous injection with $g(0) = x$ and $g(1) = y$. For every $t \in [0, 1]$, let L_t be an interval within J that contains t and $A_t\colon \mathbb{R}^n \to \mathbb{R}$ be the affine function with which f coincides on L_t. Put $\hat{t} = \sup\{t \in [0, 1] \mid A_t = A_0\}$ and prove $\hat{t} = 1$ by contradiction. If $\hat{t} < 1$, then by continuity of g there is an $\varepsilon > 0$ such that $g([\hat{t}, \hat{t}+\varepsilon]) \subset L_{\hat{t}}$. It follows that $A_{\hat{t}+\varepsilon} = A_{\hat{t}} = A_0$, a contradiction.

We have shown that in the neighbourhood of any two points in J, the function f coincides with the same affine function. Therefore f is affine on J. $\square$

Conclusion and Future Work

In this report, we have established that when representing the exact real numbers by infinite composition of affine contractions, any finitely computable function has to be affine on any area on which it is continuously differentiable. On the other hand, the signed binary representation uses affine contractions and can compute a dense set of piecewise affine functions. From these two theorems it can be informally concluded that the signed binary is one of the best among affine representations from the point of view of finite computability. In other words, one can compute with it by finite transducers all 'nice' functions which are not ruled out by the limitation theorem and a simple cardinality comparison.

One might ask whether some IFS representation which uses other than affine contractions could do better than the signed binary representation. The author believes that there is a means to generalize the limitation theorem for almost arbitrary IFS representations. Such a technique should appear in [Konečný 2000]. Out of this generalization, it follows that, for example, when using the LFT representations, any finitely computable function has to coincide with an LFT on any area where it is continuously differentiable[7]. This result does not exclude the possibility that maybe there is an IFS representation with LFT contractions which would compute a dense set of LFT's. The result of Raney [Raney 1973] gives some hope towards possible existence of such a representation.

In exact real number computation, one cannot avoid considering partial and many-valued functions (see e.g. [Brattka 1999]). The techniques used in this report have been extended to many-valued functions in [Konečný 2000].

Acknowledgements

The author would like to thank his supervisor, Prof. Achim Jung, for giving much invaluable support during this research and also the anonymous referees of the article submitted to TCS for pointing him to the relevant research in the Ukraine and giving other important comments.

This research was carried out sponsored by an ORS award of the Committee of Vice-Chancellors and Principals of the Universities of the United Kingdom and by the School of Computer Science, University of Birmingham.

References

Brattka, V. [1999], Recursive and computable operations over topological structures, Informatik Berichte 255, FernUniversität Hagen, Fachbereich Informatik, Hagen. Dissertation.

Brattka, V. & Hertling, P. [2000], Topological properties of real number representations. Accepted for publication in Theoretical Computer Science.

Edalat, A. [1995], 'Dynamical systems, measures and fractals via domain theory', *Information and Computation* **120**(1), 32–48.

Edalat, A. & Sünderhauf, P. [1997], 'A domain theoretic approach to computability on the real line', *Theoretical Computer Science* .

Escardó, M. H. [1997], PCF extended with real numbers: a domain theoretic approach to higher order exact real number computation, PhD thesis, Imperial College.

[7] A direct proof of a unary version of this statement can be found in [Konečný 1998].

Heckmann, R. [1998], The appearance of big integers in exact real arithmetic based on linear fractional transformations, *in* 'Foundations of Software Science and Computation Structures', Vol. 1378 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 172–188.

Hutchinson, J. E. [1981], 'Fractals and self-similarity', *Indiana University Mathematics Journal* **30**(5), 713–747.

Ko, K.-I. [1991], *Complexity Theory of Real Functions*, Birkhäuser, Boston.

Ko, K.-I. & Friedman, H. [1982], 'Computational complexity of real functions', *Theoretical Computer Science* **20**(3), 323–352. Fundamental study.

Konečný, M. [1998], Real functions incrementally computable by finite automatons, Technical Report CSR-98-7, School of Computer Science, University of Birmingham.

Konečný, M. [2000], Many-Valued Real Functions Computable by Finite Transducers using IFS-Representations, PhD thesis, School of Computer Science, The University of Birmingham. submitted for examination.

Lisovik, L. P., Koval, D. A. & Martinez, S. V. [1999], 'R*-transducers and fractal curves', *Kibernetika i Sistemnyi Analiz (formerly Kibernetika)* (3), 95–105. In Russian.

Lisovik, L. P. & Shkaravskaya, O. Y. [1995], 'Functions defined by push-down transducers', *Dopovidi Natsionalnoji Akademiji Nauk Ukrainy* (9), 57–59. In Russian.

Lisovik, L. P. & Shkaravskaya, O. Y. [1998], 'About real functions defined by transducers', *Kibernetika i Sistemnyi Analiz (formerly Kibernetika)* (1), 82–93. In Russian.

Mazenc, C. [1993], On the redundancy of real number representation systems, Research Report 93-16, LIP, Ecole Normale Supérieure de Lyon.

Ménissier-Morain, V. [1996], 'Arbitrary precision real arithmetic: design and algorithms', Submitted to the Journal of Symbolic Computation.

Potts, P. & Edalat, A. [1997], 'A new representation of exact real numbers', *Electronical Notes in Theoretical Computer Science* **6**.

Potts, P. J. [1998], Computable Real Arithmetic Using Linear Fractional Transformations, PhD thesis, Imperial College, Department of Computing.

Raney, G. N. [1973], 'On continued fraction and finite automata', *Mathematische Annalen* **206**, 265–283.

Shkaravskaya, O. Y. [1998], 'On affine mapping defined by finite transducers', *Kibernetika i Sistemnyi Analiz (formerly Kibernetika)* (5), 178–181. In Russian.

Trakhtenbrot, B. A. & Barzdin, Y. M. [1973], *Finite Automata—Behaviour and Synthesis*, Vol. 1 of *Fundamental Studies in Computer Science*, North Holland, Amsterdam-London.

Weihrauch, K. [1997], A foundation for computable analysis, *in* D. S. Bridges, C. S. Calude, J. Gibbons, S. Reeves & I. H. Witten, eds, 'Combinatorics, Complexity, and Logic', DMTCS'96: Discrete Mathematics and Theoretical Computer Science, Springer, Singapore, 9-13 December 1996, Auckland, New Zealand, pp. 66–89.