Analog Computers and Recursive Functions Over the Reals

Daniel Silva Graça
CLC and ADM/FCT, Universidade do Algarve, C. Gambelas, 8000-062 Faro, Portugal

José Félix Costa
DM/IST, Universidade Técnica de Lisboa, Av. Rovisco Pais, 1049-001 Lisboa, Portugal

March 14, 2003

Abstract

In this paper we show that Shannon's General Purpose Analog Computer (GPAC) is equivalent to a particular class of recursive functions over the reals with the flavour of Kleene's classical recursive function theory.

We first consider the GPAC and several of its extensions to show that all these models have drawbacks and we introduce an alternative continuous-time model of computation that solves these problems. We also show that this new model preserves all the significant relations involving the previous models (namely, the equivalence with the differentially algebraic functions).

We then continue with the topic of recursive functions over the reals, and we show full connections between functions generated by the model introduced so far and a particular class of recursive functions over the reals.

1 Introduction

Digital computation has been, since the thirties, the most important computational model, mainly due to the unifying work of Turing. Turing clarified the notion of *algorithm*, giving it a precise meaning, and introduced a coherent framework for discrete computation.

Nevertheless, computers need not be digital. In fact, the first computers were *analog computers*. In an analog computer, the internal states are continuous, rather than discrete as in digital computation. The first analog computers were especially well suited to solve ordinary differential equations and were effectively used to solve many military and civilian problems during World War II (e.g. gunfire control) and in the fifties and sixties (e.g. aircraft design). Unfortunately, because of the nonexistence of a coherent theoretical basis for analog computation and the fact that analog computing technology almost didn't improve when compared with its digital counterpart in the last half century, analog computation was about to be forgotten with the emergence of digital computation.

Despite this period of oblivion, analog computation is again regaining interest. The search for new models that could provide an adequate notion of computation and complexity for the dynamical systems that are currently used to model the physical world contributed to change the situation. However, relatively little work exists on a general theory of analog computation, and this still seems far away.

In this paper we will go to the roots of analog computation by recalling Shannon's General Purpose Analog Computer (GPAC) [14]. We will point out some problems that appear in the scope of this model and also in its subsequent versions. We will then present an alternative model (the feedforward GPAC: FF-GPAC) that solves the problems referred to for the GPAC. We will also show that the FF-GPAC model is more robust than the GPAC, but still preserves the fundamental properties (equivalence with differentially algebraic functions).

Our objective is to develop some of the ideas presented in the seminal paper [11]. In this paper, Moore introduced a recursion theory over the reals and proved some results establishing links with classical recursion theory and also with the GPAC. Unfortunately, these results presented some gaps and, hence, the primary goal of our work is precisely to provide an adequate framework to obtain, wherever possible, similar results to those presented by Moore. In particular, we are interested in the connections between recursive functions over the reals and functions generated by the GPAC. While working on this topic we have found some further problems not referred to in the existing literature (at least to our knowledge) and a major revision of the GPAC was needed for our purposes.

2 The GPAC model

In 1941, Shannon introduced the General Purpose Analog Computer [14] as a mathematical model of an analog device, the Differential Analyzer [1]. This model basically consists of circuits composed of (a finite number of) "black boxes" as indicated in fig. 2.1 (the so-called *analog units*).[1]

It is required that two inputs and two outputs can never be interconnected. It is also required that each input is connected to, at most, one output.

The GPAC model has been mainly applied to functions depending on one variable. In this case, we say that a unary function y is generated by a GPAC $\mathcal{U}$ on some interval $I \subseteq \mathbb{R}$ if we can prescribe some initial conditions to the

[1] For reasons of simplicity, we will usually identify objects with functions. The context will be enough to decide which is the case. For instance, in figure 2.1, $\int_{to}^{t} u(x)dv(x)$ should be read as $\lambda t. \int_{to}^{t} u(x)dv(x)$.

integrators of $\mathcal{U}$ at $x = a \in I$ such that if x is applied to every input that is not connected to an output, then y equals the output of some unit for values of x in I. For functions of more than one variable we may easily adapt this definition.

We will assume that if we have more than one distinct input for a GPAC, then they all depend on a parameter t that we will call the *time* (this is not very clear in existing literature. Shannon [14] assumes independent variables as inputs. Pour-El [13] talks on inputs depending on the time, but then assumes independent variables). For the one variable case, these approaches are equivalent: if x is the input of a GPAC, simply take x as the independent variable.

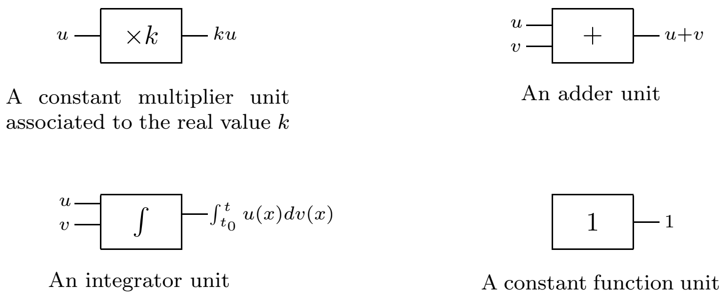

Figure 2.1: Representations of different types of units in a GPAC.

Definition 1 *A unary function y is differentially algebraic (d.a.) on some set I if there exists some natural number n (where $n > 0$) and some $(n+2)$-ary nonzero polynomial P with real coefficients such that*

$$P(x, y(x), ..., y^{(n)}(x)) = 0,$$

for every $x \in I$.[2]

In what follows, we will assume that $I \subseteq \mathbb{R}$ is a closed, bounded interval with non-empty interior.

In his paper [14], Shannon presented an argument for the following fact:

Claim 2 *A unary function can be generated by a GPAC if and only if the function is differentially algebraic.*

This result indicates that a large class of functions, such as polynomials, trigonometric functions, elliptic functions, etc., could actually be generated by a GPAC. As a corollary some functions such as the Gamma function,

$$\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt,$$

cannot be generated because they are not d.a. functions (cf. [2, pp. 49,50]).

[2]Note that this definition of differential algebraic function does not imply a definite degree for the function being defined (see definition 12).

Unfortunately, the original proof of claim 2 has some gaps, as indicated on pp. 13-14 of [13]. In this paper, Pour-El was also concerned with showing some relations in the spirit of claim 2. In order to achieve this result, she introduced an alternative definition for the GPAC based on differential equations, that we will call the theoretic GPAC (T-GPAC: this is not the usual notation, but we use it in order to distinguish this model from Shannon's GPAC). In essence, y is generated by a T-GPAC if there is some system of differential equations

$$A(x,\mathbf{y})\frac{d\mathbf{y}}{dx} = b(x,\mathbf{y}),$$

where $\mathbf{y} = (y_1, ..., y_n)$, such that $A(x,\mathbf{y})$ and $b(x,\mathbf{y})$ are $n \times n$ and $n \times 1$ matrices, respectively, with linear entries and, moreover, y is one of the y_i's. The original paper [13] can be referred for more details on the T-GPAC. Pour-El proved (with some corrections made by Lipshitz and Rubel in [9]) the following results:

Theorem 3 (Pour-El) *Let y be a differentially algebraic function on I. Then there exists a closed subinterval $I' \subseteq I$ with non-empty interior such that, on I', y can be generated by a T-GPAC.*

Theorem 4 (Pour-El, Lipshitz, Rubel) *If y is generable on I by a T-GPAC, then there is a closed subinterval $I' \subseteq I$ with non-empty interior such that, on I', y is differentially algebraic.*

Although the model presented by Pour-El is apparently different from Shannon's GPAC, Pour-El presented the following result:

Claim 5 *If a function y is generated on I by a GPAC, it is generated on I by a T-GPAC.*

This claim is the main reason why people referring to the GPAC actually recall Pour-El's definition, instead of Shannon's GPAC. Therefore the T-GPAC could be seen as a model that extends the GPAC, being able to compute functions generated by a Differential Analyzer. However, the original proof of claim 5 [13, proposition 1] has some gaps.[3]

Hence we will not deal with the T-GPAC because we don't have known relationships between functions generated by this model and functions generated by some existing analog computers (the Differential Analyzer).

We could therefore turn to Shannon's GPAC, but this model itself presents some problems. Besides the problem indicated with claim 2, we also have to deal with some more subtle details.

Consider the GPAC indicated in fig. 2.2.

[3](We assume that the reader is referring to [13] for details) In that proposition she, in fact, presents an argument where she states that if y is generated by a GPAC, then there exist functions $(y_2, ..., y_n)$ satisfying the equation of condition 1 and condition 2 of definition 10. But she never shows that $(y_2, ..., y_n)$ is the unique solution of that equation and she also does not show condition 3. She only gives a physical argument to condition 3 on p. 12 when defining domain of generation, but does not give any formal proof.

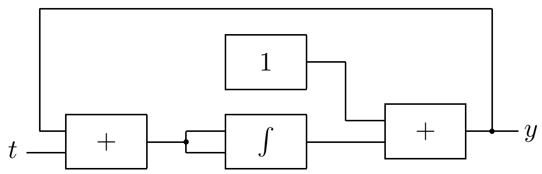

Figure 2.2: A circuit that admits two distinct solutions as outputs.

It is not difficult to see that y at time t is given by

$$y(t) = 1 + \int_0^t (y(x) + x)d(y(x) + x).$$

We are supposing that $t_0 = 0$ and that the initial output of the integrator is 0. When we start the computation, we get two possible solutions:

$$y_{\pm}(t) = 1 \pm \sqrt{-2t} - t.$$

Therefore, we cannot have a physical implementation of this circuit (in particular, there is no Differential Analyzer that simulates this circuit). This is very problematic. We will solve this problem in the next section.

3 The FF-GPAC model

In this section, we will introduce a new model based on Shannon's GPAC (it is a particular form of it), the FF-GPAC, that does not have the problems indicated in the previous section for the GPAC. In particular, we will be able to show that the output of each unit is unique on some "maximal interval" and that the relations indicated in theorems 3 and 4 will remain valid if we substitute the T-GPAC by the FF-GPAC model.

We had occasion to see that the output of a unit in a GPAC can be non-unique. We believe that this is due to the lack of restrictions on the GPAC, that allows connections with arbitrary feedback. Of course, feedback is desirable since, without it, we would get an uninteresting model. However, we believe that too much feedback can be harmful. In particular, we believe that this is the cause for the abnormal behavior of circuit 2.2. So, in this new model, we will restrict the feedback allowed.

Let us first give a brief overview of the standard procedure used to deal with the GPAC (with one input corresponding to the independent variable x). We can consider that every output of a GPAC is the output of an integrator. In fact, suppose that the output y is the output of some unit $\mathcal{A}_k$. Hence, we may introduce a new integrator and connect the input associated to the variable of integration to the output of $\mathcal{A}_k$, and the input associated to the integrand to a constant function unit. If we set the initial condition of the output of the integrator to be $y(t_0)$, the output of this integrator will be precisely y.

Proceeding in a similar way as in [14] (or [13]), let $\mathcal{U}$ be a GPAC with $n-1$ integrators $\mathcal{U}_2, ..., \mathcal{U}_n$, having as outputs $y_2, ..., y_n$, respectively. Let $y_0 = 1$ and let $y_1 = x$. Then, each input of an integrator must be the output of one of the following: an adder, an integrator, a constant function unit, or a constant multiplier. Hence, the integrand of $\mathcal{U}_k$ can be expressed as $\sum_{i=0}^{n} a_{ki} y_i$ and the variable of integration as $\sum_{i=0}^{n} b_{ki} y_i$, for some suitable constants a_{ki}, b_{ki}. Thus, the output of $\mathcal{U}_k$, y_k, can be expressed as

$$y_k = \int_{x_0}^{x} \sum_{i=0}^{n} a_{ki} y_i d\left(\sum_{j=0}^{n} b_{kj} y_j\right) + c_k, \quad \text{or}$$

$$y_k = \int_{x_0}^{x} \sum_{i,j=0}^{n} a_{ki} b_{kj} y_i dy_j + c_k.$$

We may simplify the last expression by taking $c_{ij}^k = a_{ki} b_{kj}$. It follows that

$$y_k' = \sum_{i,j=0}^{n} c_{ij}^k y_i y_j', \qquad k = 2, ..., n. \tag{1}$$

Hence, we could assert that it is equivalent to say that y is generated by a GPAC (in the sense that it is a solution for a system of equations where each equation is associated to a unit) and that y is a solution of some system of differential equations with the particular structure of (1). Although this seems a clear and natural procedure, we believe that we have to be more careful with it. For instance, why should we consider that the inputs of the integrator $\mathcal{U}_k$ could be expressed as $\sum_{i=0}^{n} a_{ki} y_i$ and $\sum_{i=0}^{n} b_{ki} y_i$ for some constants a_{ki}, b_{kj}? We could have as input the output of a circuit like the one of fig. 3.1.

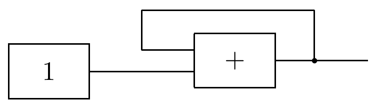

Figure 3.1: A circuit that admits no solutions as outputs.

This circuit follows the definition of a GPAC, but we cannot say that its output is a linear combination of the input, because it does not exist (the output of the circuit is a solution of the equation $x = x + 1$). However these circuits are admissible in Shannon's or Pour-El's frameworks. But we would like to keep relations as in (1). To do so, we introduce the concept of linear circuit as follows.

Definition 6 *A* linear circuit *is an acyclic GPAC built only with adders, constant multipliers, and constant function units in the following inductive way:*

1. *A constant function unit is a linear circuit with zero inputs and one output;*
2. *A constant multiplier is a linear circuit with one input and one output;*

3. *An adder is a linear circuit with two inputs and one output;*

4. *If $\mathcal{A}$ is a linear circuit and if we connect the output of $\mathcal{A}$ to a constant multiplier, the resulting GPAC is a linear circuit. It has as inputs the inputs of $\mathcal{A}$ and as outputs, the output of the constant multiplier;*

5. *If $\mathcal{A}$ and $\mathcal{B}$ are linear circuits and if we connect the outputs of $\mathcal{A}$ and $\mathcal{B}$ to an adder, then the resulting circuit is a linear circuit in which the inputs are the inputs of $\mathcal{A}$ and $\mathcal{B}$, and the output is the output of the adder.*

The proof of the following proposition will be left to the reader.

Theorem 7 *If $x_1, ..., x_n$ are the inputs of a linear circuit, then the output of the circuit will be $y = c_0 + c_1x_1 + ... + c_nx_n$, where $c_0, c_1, ..., c_n$ are appropriate constants. Reciprocally, if $y = c_0 + c_1x_1 + ... + c_nx_n$, then there is a linear circuit with inputs $x_1, ..., x_n$, and output y.*

We next introduce a new type of unit.

- *Input unit*: A zero-input, one-output unit.

The input units may be considered as interface units for the inputs from outside world in order that they can be used by a GPAC (although we may pick an output directly from the GPAC).

We now present the main definition of this section.

Definition 8 *Consider a GPAC $\mathcal{U}$ with n integrators $\overline{\mathcal{U}}_1, ..., \overline{\mathcal{U}}_n$. Suppose that to each integrator $\overline{\mathcal{U}}_i$, $i = 1, ..., n$, we can associate two linear circuits, $\mathcal{A}_i$ and $\mathcal{B}_i$, with the property that the integrand and the variable of integration inputs of $\overline{\mathcal{U}}_i$ are connected to the outputs of $\mathcal{A}_i$ and $\mathcal{B}_i$, respectively. Suppose also that each input of the linear circuits $\mathcal{A}_i$ and $\mathcal{B}_i$ is connected to one of the following: the output of an integrator or to an input unit. $\mathcal{U}$ is said to be a feedforward GPAC (FF-GPAC) iff there exists an enumeration of the integrators of $\mathcal{U}$, $\mathcal{U}_1, ..., \mathcal{U}_n$, such that the variable of integration of the kth integrator can be expressed as*

$$c_k + \sum_{i=1}^{m} c_{ki}x_i + \sum_{i=1}^{k-1} \bar{c}_{ki}y_i, \qquad \textit{for all } k = 1, ..., n, \tag{2}$$

(see fig. 3.2) where y_i is the output of $\mathcal{U}_i$, for $i = 1, ..., n$, x_j is the input associated to the jth input unit, and $c_k, c_{kj}, \bar{c}_{ki}$ are suitable constants, for all $k = 1, ..., n$, $j = 1, ..., m$, and $i = 1, ..., k-1$.

Note that in a FF-GPAC we can link the output of $\mathcal{U}_k$ directly to the input of $\mathcal{U}_r$ (as long as (2) is satisfied) because this is equivalent to having a constant multiplier associated with the value 1 between them.

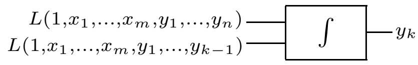

Figure 3.2: Schema of the inputs and outputs of $\mathcal{U}_k$ in the FF-GPAC $\mathcal{U}$. $L(z_1, ..., z_r)$ denotes a linear combination (with real coefficients) of $z_1, ..., z_r$.

We can also have a notion of function generated by a FF-GPAC using a straightforward adaptation of the corresponding definition for the case of the GPAC. We can also suppose that each function generated by a FF-GPAC is the output of an integrator.

Before continuing with our work, we have to set up more conditions on this model. We have admitted that the inputs $x_1, ..., x_m$ of a GPAC are functions of a parameter t, but we didn't make any assumption on these functions (about computability, differentiability, or whatever). When we consider the integrator units, one problem still arises: if $I = [a, b]$ is a closed interval, the Riemann-Stieltjes integral $\int_I \varphi(t)d\psi(t)$ is not defined for every pair of functions φ, ψ, even if they are continuous [16].

However, it is possible to show that if φ, ψ are continuously differentiable on I, then $\int_I \varphi(x)d\psi(x)$ is defined. So, from now on, we will always assume that the inputs are continuously differentiable functions of the time. And if the outputs of all units are defined for all $t \in I$, where I is an interval, then we will also assume that they are continuous in that interval. This is needed for the following results and may be seen as physical constraints to which all units are subjected.

4 Properties of the FF-GPAC

We now state the following theorems. Their proofs can be found in the dissertation [6].

Theorem 9 *Suppose that the input functions of a FF-GPAC are of class C^r on some interval I, for some $r \geq 1$, possibly ∞. Then the outputs are also of class C^r on I.*

Theorem 10 *Consider a FF-GPAC with m inputs $x_1, ..., x_m$ of class C^2 on an interval $[t_0, t_f)$, where t_f may possibly be ∞. Then there exists an interval $[t_0, t^*)$ (with $t^* \leq t_f$) where each output exists and is unique. Moreover, if $t^* < t_f$ then there exists an integrator with output y such that y gets unbounded as $t \to t^*$.*

These results show that the FF-GPAC is a "well-behaved" model that is dynamically more robust than the GPAC. We will now show relations between functions computable by the FF-GPAC and differentiably algebraic functions in the spirit of claim 2 and theorems 3 and 4. We will only consider a FF-GPAC with one input x, the independent variable, and we will also only consider functions of class C^∞.

Definition 11 *Pol_n denotes the set of n-ary polynomials with real coefficients, for $n \in \mathbb{N}$ and $n \geq 1$.*

Definition 12 *Let y be a differentially algebraic function on J. For each $n \in \mathbb{N}$, let*

$$P_n(y) = \{P \in Pol_{n+2} : P \neq 0 \text{ and } \forall x \in J,\ P(x, y(x), ..., y^{(n)}(x)) = 0\}.$$

The order *of y on J is given by*

$$order(y) = \min\{n \in \mathbb{N} : P_n(y) \neq \varnothing\}.$$

Theorem 13 *If y is generable on I by a FF-GPAC with n integrators, then there exists a nonzero polynomial p with real coefficients such that*

$$p\left(x, y, y', ..., y^{(n)}\right) = 0, \qquad on\ I.$$

Proof. Suppose that we have a FF-GPAC $\mathcal{U}$ with $n-1$ integrators $\mathcal{U}_2, ..., \mathcal{U}_n$ in an appropriate order of enumeration, with outputs $y_2, ..., y_n$, respectively. We only have to show that the functions $y_2, ..., y_n$ are d.a. on I. We will show that y_2 is d.a.. Let $y_0 = 1$ and $y_1 = x$. Then using the standard procedure for the GPAC indicated in the previous section, we conclude that

$$y_k' = \sum_{i=0}^{n} \sum_{j=1}^{k-1} c_{ij}^k y_i y_j', \qquad k = 2, ..., n,$$

where c_{ij}^k are suitable constants. We can write this as

$$y_k' - \sum_{j=2}^{k-1} \left(\sum_{i=0}^{n} c_{ij}^k y_i \right) y_j' = \sum_{i=0}^{n} c_{i1}^k y_i, \qquad k = 2, ..., n.$$

This system may be written in the following way

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ -\sum_{i=0}^n c_{i2}^3 y_i & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -\sum_{i=0}^n c_{i2}^n y_i & -\sum_{i=0}^n c_{i3}^n y_i & \cdots & 1 \end{bmatrix} \begin{bmatrix} y_2' \\ y_3' \\ \vdots \\ y_n' \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^n c_{i1}^2 y_i \\ \sum_{i=0}^n c_{i1}^3 y_i \\ \vdots \\ \sum_{i=0}^n c_{i1}^n y_i \end{bmatrix}$$

or simply

$$A\mathbf{y}' = \mathbf{b}, \qquad \text{where } \mathbf{y} = \begin{bmatrix} y_2 \\ \vdots \\ y_n \end{bmatrix}.$$

It is easily seen that $\det A = 1$. Hence A is invertible and we have

$$A^{-1} = A_{cof},$$

where A_{cof} is the transpose of the matrix in which each entry is its respective cofactor with respect to A. So, each entry in A^{-1} is a polynomial in $x, y_2, ..., y_n$. The same happens with $\mathbf{b}$. We know that

$$\mathbf{y}' = A^{-1}\mathbf{b}.$$

Then we may write

$$\begin{aligned} y_2' &= \bar{P}_2(x, y_2, ..., y_n) \\ &\vdots \\ y_n' &= \bar{P}_n(x, y_2, ..., y_n) \end{aligned} \tag{3}$$

where each $\bar{P}_i$ is an n-ary polynomial, for $i = 2, ..., n$. Differentiating $y_2' = \bar{P}_2(x, y_2, ..., y_n)$ with respect to x and using (3), we get

$$y_2^{(k)} = P_k(x, y_2, ..., y_n), \qquad k = 1, ..., n-1, \tag{4}$$

where each P_k is a n-ary polynomial. Now consider the field of n-ary rational functions over $\mathbb{R}$,

$$Rat_n = \left\{ \frac{p}{q} : p, q \in Pol_n, \text{ and } q \neq 0 \right\},$$

(for the results on algebra, cf. [7, pp. 311-317]). It is easy to see that $x, y_2 \in Rat_n$ and that $P_k(x, y_2, ..., y_n) \in Rat_n$, for $k = 1, ..., n-1$. But a transcendence basis of Rat_n can only have n elements. Hence, the $n+1$ polynomials

$$x, y_2, P_k(x, y_2, ..., y_n), \quad k = 2, ..., n,$$

must be algebraically dependent, i.e., there exists a nonzero polynomial $p \in Pol_n$ such that $p(x, y_2, P_2(x, y_2, ..., y_n), ..., P_n(x, y_2, ..., y_n)) = 0$. Using (4), we get

$$p\left(x, y_2, y_2', ..., y_2^{(n-1)}\right) = 0,$$

as we wanted to prove. ■

Notice that we can conclude from the proof of this theorem that if a function y is generated by a FF-GPAC, then it is a solution of a certain continuous dynamical system $\mathbf{y}' = \mathbf{p}(\mathbf{y}, t)$ where y is a component of $\mathbf{y} = (y_1, ..., y_n)$ and $\mathbf{p}$ is a vector of polynomials. It is also not difficult to show that the converse result is also true.

Corollary 14 *y is generated by a FF-GPAC iff it is a component of the solution $\mathbf{y} = (y_1, ..., y_n)$ of a differential equation $\mathbf{y}' = \mathbf{p}(\mathbf{y}, t)$ where $\mathbf{p}$ is a vector of polynomials.*

Corollary 15 *If y is generable on I by a FF-GPAC, then y is differentially algebraic on I.*

Corollary 16 *Suppose that y is generable on I by a FF-GPAC. Then there exists a closed subinterval $I' \subseteq I$ with non-empty interior such that, on I', y can be generated by a T-GPAC.*

Proof. This result follows from theorem 3 and corollary 15 . ■

We now show a converse to corollary 15.

Theorem 17 *Suppose that y is differentially algebraic on I. Then there is a closed subinterval $I' \subseteq I$ with non-empty interior such that y can be generated by a FF-GPAC on I'.*

Proof. This proof follows much along the lines [13, theorem 4]. Let n be the order of y and let $P(x_1, ..., x_{n+2})$ be the expression of a polynomial P of lowest degree in x_{n+2} such that

$$P(x, y, y', ..., y^{(n)}) = 0$$

on I. Differentiating formally the last equation (with respect to x),[4] we get

$$\frac{\partial P}{\partial x} + \frac{\partial P}{\partial y} y' + \frac{\partial P}{\partial y'} y'' + ... + \frac{\partial P}{\partial y^{(n)}} y^{(n+1)} \equiv 0, \quad \text{or}$$

$$R(x, y, y', ..., y^{(n)}) y^{(n+1)} - Q(x, y, y', ..., y^{(n)}) \equiv 0 \tag{5}$$

on I, where R and Q are $(n+2)$-ary polynomials having the property that R is of lower degree than P in the last variable. Therefore, we must have $R(x, y, y', ..., y^{(n)}) \not\equiv 0$ on I, i.e., there exists some $a \in I$ that satisfies the condition $R(a, y(a), ..., y^{(n)}(a)) \neq 0$. Then we may pick some closed, bounded intervals $J, J_0, ..., J_n$ (with non-empty interiors) such that

(i) $J \subseteq I$,
(ii) $(a, y(a), ..., y^{(n)}(a)) \in J \times J_0 \times ... \times J_n$,
(iii) If $(c, c_0, ..., c_n) \in J \times J_0 \times ... \times J_n$, then $R(c, c_0, ..., c_n) \neq 0$,
(iv) If $x \in J$, then $(x, y(x), ..., y^{(n)}(x)) \in J \times J_0 \times ... \times J_n$.

We can rewrite (5) as a system of $n+2$ first-order equations, with $n+2$ variables defined on $J \times J_0 \times ... \times J_n$. Hence, this system satisfies a Lipschitz condition in J and the solution is unique. So, y is the unique solution possessing initial conditions $a, y(a), y'(a), ..., y^{(n)}(a)$ which satisfies (5) on J. Next, we are going to prove that y can be generated by a FF-GPAC on J. We begin by finding the equations that define the corresponding FF-GPAC. Solving for $y^{(n+1)}$ in (5), we get

$$y^{(n+1)} = \frac{Q(x, y, y', ..., y^{(n)})}{R(x, y, y', ..., y^{(n)})}.$$

Introducing the variables $y_1 = y, y_2 = y', ..., y_{n+1} = y^{(n)}, y_{n+2} = y^{(n+1)}$, the previous equation becomes

$$y'_k = y_{k+1}, \quad k = 1, ..., n+1, \tag{6}$$

$$y_{n+2} = \frac{Q(x, y_1, ..., y_{n+1})}{R(x, y_1, ..., y_{n+1})}.$$

[4] Notice that we can differentiate the equation because we assumed in the beginning of this section that y is a function of class C^∞.

It is not difficult to show that y can be generated by a FF-GPAC having only one independent variable x iff there are functions $z_2, ..., z_g$ satisfying the relations

$$z_k' = \sum_{i=0}^{g} \sum_{j=1}^{k-1} c_{ki} c_{kj} z_i z_j', \qquad k = 2, ..., g. \tag{7}$$

where c_{ki}, c_{kj} are reals, and $z_0 = 1$, $z_1 = x$, with $y = z_i$ for some $i = 2, ..., g$. Although the first $n+1$ equations of (6) satisfy (7), the same does not happen for the last equation (y_{n+2} is expressed as the quotient of two polynomials). Hence, we will substitute the last equation of (6) by a system of equations satisfying (7).

Consider the polynomial Q. Each term of Q that is not a constant is of the form $bv_1...v_r$, where each v_i is x or one of the variables y_j and b is a real number. Suppose that the first term is $bv_1...v_r$. Taking

$$y_{n+3}' = bv_1 v_2' + bv_2 v_1'$$

with initial condition $y_{n+3}(a) = bv_1(a)v_2(a)$, we have $y_{n+3} = bv_1v_2$. In a similar way, taking

$$y_{n+4}' = v_3 y_{n+3}' + y_{n+3} v_3'$$

with initial conditions $y_{n+4}(a) = v_3(a)y_{n+3}(a)$, we get $y_{n+4} = bv_1v_2v_3$. Continuing with this procedure we will have $y_{n+r+1} = bv_1v_2...v_r$. We can apply this method to all non-constant terms of Q and R, obtaining new equations. Let the y's corresponding to the non-constant terms of Q be $w_1, ..., w_s$ and let $w_1^*, ..., w_t^*$ be those corresponding to the non-constant terms of R. Then

$$y_{n+2} = \frac{\left(\sum_{i=1}^{s} w_i + c\right)}{\left(\sum_{i=1}^{t} w_i^* + c^*\right)}, \tag{8}$$

where c and c^* are real constants. Now, we must reduce the last equation to a form that fits (7). Suppose that the last y, w_t^*, was y_{p-1}. Let

$$y_p' = -y_{p+1} \sum_{i=1}^{t} (w_i^*)', \tag{9}$$

$$y_{p+1}' = 2y_p y_p',$$

with initial conditions

$$y_p(a) = \left(\sum_{i=1}^{t} w_i^*(a) + c^* \right)^{-1}, \quad y_{p+1}(a) = y_p^2(a).$$

Then $y_p = \left(\sum_{i=1}^{t} w_i^* + c^*\right)^{-1}$, $y_{p+1} = y_p^2$, and we may replace (8) by three equations: equations (9) and

$$y_{n+2}' = y_p \sum_{i=1}^{s} (w_i)' + \left(\sum_{i=1}^{s} w_i + c \right) y_p',$$

where $y_{n+2}(a) = \left(\sum_{i=1}^{s} w_i(a) + c\right) .y_p(a)$. We conclude that (6) can be replaced by the following system of equations

$$\begin{aligned}
y'_k &= y_{k+1}, \quad k = 1, ..., n+1 \\
y'_{n+3} &= bv_1 v'_2 + bv_2 v'_1, \quad \text{where } v_1, v_2 \in \{x, y_1, ..., y_{n+1}\} \\
&\vdots \\
y'_{p-1} &= y_{p-2} v'_m + v_m y'_{p-2}, \quad \text{where } v_m \in \{x, y_1, ..., y_{n+1}\} \\
y'_p &= -\sum_{i=1}^{t} y_{p+1} (w_i^*)' \\
y'_{p+1} &= 2 y_p y'_p \\
y'_{n+2} &= \sum_{i=1}^{s} y_p . w'_i + \left(\sum_{i=1}^{s} w_i + c \right) y'_p
\end{aligned}$$

It is easily seen that this system is in the form of (7) (the respective sequence of $z_2, ..., z_g$ is $y_1, ..., y_{n+1}, y_{n+3}, ..., y_{p+1}, y_{n+2}$) and that $y = y_1$ on J. So, y is computable by a FF-GPAC on J. ■

Corollary 18 *If y is generable on I by a T-GPAC, then there is a closed subinterval $I' \subseteq I$ with non-empty interior such that y is generable by a FF-GPAC on I'.*

Proof. Follows from theorems 4 and 17. ■

5 Analog circuits and Recursive functions over $\mathbb{R}$

We now recall recursion theory over the reals [11]. Adapting the definition presented in [11], we say that a function $h : D \subseteq \mathbb{R}^n \to \mathbb{R}$ is $\mathbb{R}$-recursive if it can be defined inductively from the constants $0, 1$, the projections $U_i(\mathbf{x}) = x_i$, and the following operators:[5]

- *Composition:* Suppose that g is an p-ary function, with $p \geq 1$, and that $f_1, ..., f_p$ are p n-ary functions. Then the composition operator applied to these functions in that order yields the n-ary function h given by $h(\mathbf{x}) = g(f_1(\mathbf{x}), ..., f_p(\mathbf{x}))$;
- *Integration:* Suppose that $f_1, ..., f_m$ are n-ary functions, and $g_1, ..., g_m$ are $n + m + 1$-ary functions. The integration operator applied to $f_1, ..., f_m$, $g_1, ..., g_m$, in that order, yields the $n + 1$-ary function h defined for each

[5] A minimalization operator was also introduced in [11]. But we will not use it here.

$(\mathbf{x}, y) \in \mathbb{R}^{n+1}$ as follows. Let I be the largest interval in which a unique unary continuous function $\mathbf{s}$ satisfying

$$\mathbf{s}(0) = \mathbf{f}(\mathbf{x})$$
$$\partial_z \mathbf{s}(z) = \mathbf{g}(\mathbf{x}, z, \mathbf{s}(z)),$$

exists. Then, if $y \in I$, $h(\mathbf{x}, y) = s_1(y)$. Otherwise $h(\mathbf{x}, y)$ is undefined.

This theory has the flavour of the classical recursion theory. It is clear that this definition pretends to match the primitive recursive functions over $\mathbb{N}$, where integration corresponds to primitive recursion. Moore was able to show that the most usual functions used in analysis are $\mathbb{R}$-recursive. For the sake of completeness, we also note that Campagnolo and Costa added several contributions to this theory [3], [4], [5].

It was also indicated in [11] that a subclass of $\mathbb{R}$-recursive functions corresponded exactly to functions generated by a T-GPAC [11, proposition 9]. Unfortunately, we believe that this result presents some gaps.[6] Therefore, we would like to present a computational model for at least a subclass of $\mathbb{R}$-recursive functions.

Consider the alphabet constituted by the following symbols:

$$\begin{array}{ll} \text{Basic Functions} & \left\{ \begin{array}{l} 0, 1, -1, +, \times \\ U_i^n, \text{ for } 1 \leq i \leq n \text{ and } n \in \mathbb{N}. \end{array} \right. \\ \text{Operators} & \text{CM}, \text{INT} \\ \text{Punctuation} & (,), , \end{array}$$

We will first focus on the definition of descriptions with *active* and *locked variables* (that we will abbreviate by AV and LV, respectively).

In what follows, we will introduce the FF-GPAC as a model for functions associated to these descriptions and, informally, a variable x_i of a description is active if the value of the corresponding function depends on x_i and if this variable may be freely updated when the function is implemented in a FF-GPAC. A variable x_i is locked if the value of the corresponding function depends on x_i, but the value of x_i must be fixed a priori, when generating the function in a FF-GPAC. Note that, because we have not introduced a set of variables in our alphabet, when we refer to the variable x_i of the function f, we are referring to the ith argument of f.

Definition 19 *Descriptions with active and locked variables are expressions formed with the following rules:*

1. $\mathbb{R}$ *is a set of* 0*-ary descriptions without AVs or LVs;*

[6](We assume that the reader is referring to [11]) The proof of this proposition relies on the lemma preceding it that states that M_0 (a class defined on [11]) is closed under differentiation and inversion. However, the proof of this lemma presents some vicious circles when considering the operation of integration.

2. *U_i^n is a description of arity n, for $1 \leq i \leq n$ and $n \in \mathbb{N}$. It has one active variable, x_i, and no LVs;*

3. *$\times$ and $+$ are descriptions of arity 2. They have x_1 and x_2 as AVs, and no LVs;*

4. *If $G, F_1, ..., F_m$ are descriptions, where $F_1, ..., F_m$ have arity n, and G has arity m, then $\mathrm{CM}(G, F_1, ..., F_m)$ is a description of arity n. Let $x_{k_1}, ..., x_{k_s}$ and $x_{u_1}, ..., x_{u_r}$ be the AVs and LVs of G, respectively. Moreover, let A_i, L_i be the sets containing the AVs and LVs of F_i, respectively, for $i = 1, ..., m$. Then the sets of LVs and AVs of $\mathrm{CM}(G, F_1, ..., F_m)$ are given by*

$$L = \left(\bigcup_{i=1}^{s} L_{k_i} \right) \cup \bigcup_{i=1}^{r} (A_{u_i} \cup L_{u_i}) \quad , \quad A = \left(\bigcup_{i=1}^{s} A_{k_i} \right) \backslash L,$$

   respectively;

5. *Let $F_1, ..., F_m$ be descriptions of arity n, and $G_1, ..., G_m$ be descriptions of arity $n+m+1$. Suppose that the set constituted by all the AVs and LVs of F_i, G_i, for every $i = 1, ..., m$ is S_1. Take $S = S_1 \cap \{x_1, ..., x_n\}$. If G_i doesn't have any LV among the variables $x_{n+1}, ..., x_{n+m+1}$ for every $i = 1, ..., n$, then $\mathrm{INT}(F_1, ..., F_m, G_1, ..., G_m)$ is a description of arity $n+1$. It has as sets of AVs and LVs, $\{x_{n+1}\}$ and S, respectively.*

To each n-ary description, we associate an n-ary function in the following way:

1. To each description $a \in \mathbb{R}$ corresponds the 0-ary constant function with the corresponding value;

2. To each U_i^n corresponds the projection $f : \mathbb{R}^n \to \mathbb{R}$ defined by $f(\mathbf{x}) = x_i$;

3. To the descriptions $\times$ and $+$ corresponds binary addition and product.[7]

4. If a description is given by $\mathrm{CM}(G, F_1, ..., F_m)$ and to $G, F_1, ..., F_m$ are associated the functions $g, f_1, ..., f_m$ (where n is the arity of the f_i's), then we associate to $\mathrm{CM}(G, F_1, ..., F_m)$ the n-ary function h defined by $h(\mathbf{x}) = g(f_1(\mathbf{x}), ..., f_m(\mathbf{x}))$;

5. Suppose that a description is given by $\mathrm{INT}(F_1, ..., F_m, G_1, ..., G_m)$ and that to $F_1, ..., F_m, G_1, ..., G_m$ are associated the functions $f_1, ..., f_m, g_1, ..., g_m$ of arities $n, ..., n, n+m+1, ..., n+m+1$, respectively. Then we associate to $\mathrm{INT}(F_1, ..., F_m, G_1, ..., G_m)$ the $n+1$-ary function h defined for each

[7] Notice that we could obtain binary addition and product from the other functions and operators. Nevertheless, this would associate one locked variable to both functions, and this is not desirable when defining other functions that use the addition and the product. Therefore, we choose to introduce them as basic functions without associated LVs to avoid this problem. However, different approaches to this problem might be possible.

$(\mathbf{x}, y) \in \mathbb{R}^{n+1}$ as follows. Let I be the largest interval in which a unique unary function $\mathbf{s}$ satisfying

$$\begin{aligned} \mathbf{s}(0) &= \mathbf{f}(\mathbf{x}) \\ \partial_z \mathbf{s}(z) &= \mathbf{g}(\mathbf{x}, z, \mathbf{s}(z)), \quad \forall z \in I, \end{aligned}$$

exists. Then, if $y \in I$, $h(\mathbf{x}, y) = s_1(y)$. Otherwise $h(\mathbf{x}, y)$ is undefined.

Notice that the inclusion of active and locked variables in the definition of INT restricts this operator comparatively to the integration operator. Therefore, descriptions using INT are essentially equivalent to descriptions that use, at most, one application of the integration operator.

It is easy to verify (by structural induction) that if a description H has as sets of AVs and LVs, A and L, respectively, then $A \cap L = \varnothing$. Moreover, if h is the function associated to H and $x_k \notin A \cup L$, then h does not depend on x_k.

Definition 20 *The class $\mathcal{I}_\mathbb{R}$ is the class constituted by the functions that can be obtained from $\mathbb{R}$, the binary addition and product, and the projections by applying the operators* CM *and* INT *a finite number of times.*

It is not difficult to see (use similar arguments to those presented in [11, proposition 2]) that the most usual functions also belong to $\mathcal{I}_\mathbb{R}$ (notice that they use, at most, one application of the integration operator). We now relate the class $\mathcal{I}_\mathbb{R}$ with the functions generated by the FF-GPAC as a main contribution to recursive function theory over the reals of Moore, Campagnolo, and Costa.

Theorem 21 *Suppose that a unary function f can be generated by a FF-GPAC $\mathcal{U}$. Then $f \in \mathcal{I}_\mathbb{R}$.*

Proof. Suppose that f is generated by a FF-GPAC $\mathcal{U}$, with integrators $\mathcal{U}_2, ..., \mathcal{U}_n$ in an appropriate order, with outputs $y_2, ..., y_n$, respectively. Then, if $y_0 = 1$, $y_1 = x$, we have

$$y'_k = \sum_{i=0}^{n} \sum_{j=1}^{k-1} c^k_{ij} y_i y'_j, \qquad k = 2, ..., n,$$

for suitable constants c^k_{ij}. Consider the functions $g_k \in \mathcal{I}_R$ of arity $n + k - 2$, for $k = 2, ..., n$, defined as

$$g_k = c^k_{01} + \sum_{i=1}^{n} \left(c^k_{i1} x_i \right) + \sum_{j=2}^{k-1} \left(c^k_{0j} z_j \right) + \sum_{i=1}^{n} \sum_{j=2}^{k-1} \left(c^k_{ij} x_i z_j \right).$$

Note that

$$y'_k = g_k \left(y_1, ..., y_n, y'_2, ..., y'_{k-1} \right).$$

Next, we prove by induction on k that there are functions $g^*_k \in \mathcal{I}_\mathbb{R}$, $k = 2, ..., n$, such that

$$y'_k = g^*_k(y_1, ..., y_n).$$

For $k = 2$ the result is immediate. Take $g_2^* = g_2$. For arbitrary $k > 2$ we know that

$$y'_k = g_k\left(y_1, ..., y_n, y'_2, ..., y'_{k-1}\right), \tag{10}$$

and also, by induction hypothesis,

$$y'_j = g_j^*(y_1, ..., y_n), \qquad \text{for } j = 2, ..., k-1.$$

Substituting the last $k-2$ equations in (10), we get

$$y'_k = g_k\left(y_1, ..., y_n, g_2^*(y_1, ..., y_n), ..., g_{k-1}^*(y_1, ..., y_n)\right).$$

If we take g_k^* as the composition of g_k with $U_1^n, ..., U_n^n, g_2^*, ..., g_{k-1}^*$, then we get the desired function that also belongs to $\mathcal{I}_{\mathbb{R}}$.

Now, suppose that f is defined on an interval I, and that the initial conditions of the integrators are prescribed at $x_0 = a$. We can, without loss of generality, consider $a = 0$ for our next purpose.

Then, y_2 can be obtained by integration in the following way:

$$\begin{bmatrix} y_2(0) \\ \vdots \\ y_n(0) \end{bmatrix} = \begin{bmatrix} a_2 \\ \vdots \\ a_n \end{bmatrix} \quad , \qquad \partial_x \begin{bmatrix} y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} g_2^* \\ \vdots \\ g_n^* \end{bmatrix}. \tag{11}$$

Switching the rows in the last equations, we can show that $y_2, ..., y_n \in \mathcal{I}_{\mathbb{R}}$. Therefore $y \in \mathcal{I}_{\mathbb{R}}$. ■

Note that this result cannot be extended for functions of more than one variable. For instance, consider a FF-GPAC that generates the function $z(w) = \int_0^w (x^2(t) + y(t))dt$, for inputs x and y. If x and y correspond to the inputs t and t^2, then $z(1) = 5/3$. On the other hand if we switch the inputs, then $z(1) = 17/10$. Therefore, if theorem 21 could be extended for functions of more than one variable, we should have $f(x = 1, y = 1) = 5/3 = 17/10$, and this is not possible.

We now show a converse of theorem 21.

Definition 22 *Let $f \in \mathcal{I}_{\mathbb{R}}$ be an n-ary function defined on $D \subseteq \mathbb{R}^n$ where, without loss of generality, $x_1, ..., x_m$ are the locked variables. A function $\varphi : I \to D$, where $I \subseteq \mathbb{R}$, is said to be a $\mathcal{I}_{\mathbb{R}}$-path if it is a constant function on the locked variables.*

Theorem 23 *Suppose that $f \in \mathcal{I}_{\mathbb{R}}$ is an n-ary function where, without loss of generality, $x_1, ..., x_m$ are the locked variables and $x_{m+v}, ..., x_n$ are the active variables. Let φ be a continuously differentiable $\mathcal{I}_{\mathbb{R}}$-path on $[t_0, t_f)$. Then $f \circ \varphi$ can be generated by a FF-GPAC working with inputs $\varphi_{m+v}, ..., \varphi_n$ on $[t_0, t_f)$.*

Proof. We basically have to show that the function

$$\tilde{f}(t) = f(a_1, ..., a_m, 0, ..., 0, \varphi_{m+v}(t), ..., \varphi_n(t)),$$

where $a_1, ..., a_m$ are some real constants, can be generated by a FF-GPAC working with inputs $\varphi_{m+v}, ..., \varphi_n$ on $[t_0, t_f)$. The proof is done by induction on the structure of the description of f. The constant functions in $\mathbb{R}$ can be obtained in a straightforward way. Similarly U_i^n can be obtained using a circuit built only with input units, by taking the output of the input unit associated to x_i.

For the case of $+$, it is enough to use an adder and two input units connected to the adder to get the result. In the case of product, we use the circuit sketched in figure 5.1.

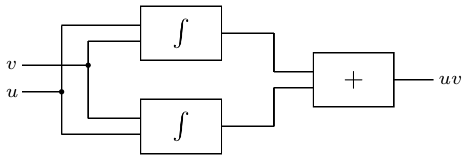

Figure 5.1: A circuit that computes uv.

To show that this circuit works, one only has to consider the formula of integration by parts (for definite integration)

$$u(t)v(t) = \int_{t_0}^{t} u(x)dv(x) + \int_{t_0}^{t} v(x)du(x) + u(t_0)v(t_0).$$

Inductive step: consider now the two operators. Suppose that we have $f = CM(g, f_1, ..., f_k)$. Suppose, without loss of generality, that g has $x_1, ..., x_s$ and $x_{s+d}, ..., x_k$ as LVs and AVs, respectively. Because the locked variables of f are $x_1, ..., x_m$, we conclude that $f_1, ..., f_s$ cannot depend on $x_{m+1}, ..., x_n$. Let $x_{i_1^r}, ..., x_{i_\alpha^r}$, and $x_{j_1^r}, ..., x_{j_\beta^r}$ be the LVs and AVs of f_r, for $r = 1, ..., k$, with

$$i_1^r < ... < i_\alpha^r \text{ and } j_1^r < ... < j_\beta^r.$$

We have $i_\alpha^r, j_\beta^r \leq m$, for $r = 1, ..., s$ (and $i_\alpha^r \leq m$ for $r = s + d, ..., k$). Using the induction hypothesis, there exists, for $r = 1, ..., k$, a FF-GPAC $\mathcal{F}_r$ that generates the function

$$f_r(a_{i_1^r}, ..., a_{i_\alpha^r}, 0, ..., 0, \varphi_{j_1^r}(t), ..., \varphi_{j_\beta^r}(t), 0, ..., 0),$$

(the actual order of the variables could be different, but this is not important for us). Now substitute in $\mathcal{F}_r$, for $s + d, ..., k$, the input units associated to x_t by a_t, for $t = 1, ..., m$, respectively (hence, these FF-GPACs compute f_r, with the first m arguments fixed). We still denote these new FF-GPACs by $\mathcal{F}_r$, for $r = 1, ..., k$. Let

$$b_r = f_r(a_{i_1^r}, ..., a_{i_\alpha^r}, 0, ..., 0, a_{j_1^r}, ..., a_{j_\beta^r}, 0, ..., 0),$$

for $r = 1, ..., s$. Also by the induction hypothesis, there exists a FF-GPAC $\mathcal{G}$ that computes the function

$$g(b_1, ..., b_s, 0, ..., 0, \varphi_{s+d}(t), ..., \varphi_k(t)).$$

With these FF-GPACs, we can build a FF-GPAC that computes the composition of functions. This is sketched in figure 5.2.

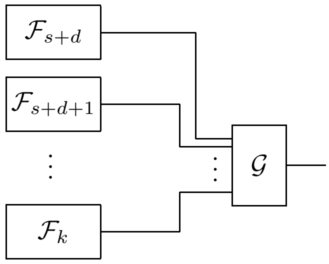

Figure 5.2: A circuit that computes the composition of functions.

Finally, suppose that $\mathrm{INT}(F_1, ..., F_k, G_1, ..., G_k)$ is the description associated to f. By similar arguments to the previous case, we can obtain FF-GPACs $\mathcal{G}_1, ..., \mathcal{G}_m$ such that $\mathcal{G}_j$ computes the function

$$g_j(a_1, ..., a_m, 0, ..., 0, \varphi_{n+1}(t), ..., \varphi_{n+k}(t)).$$

Finally

$$\tilde{f}(t) = f(a_1, ..., a_m, 0, ..., 0, \varphi_n(t))$$

can be obtained by the FF-GPAC indicated in figure 5.3, where the integrator connected to $\mathcal{G}_i$, for $i = 1, ..., k$, takes the following initial value

$$f_i(a_1, ..., a_m, 0, ..., 0).$$

To see that we can pick an appropriate enumeration of the integrators of the circuit represented in figure 5.3, pick appropriate enumerations for the various sub-FF-GPACs and then take the following general enumeration:

$$\int_1, ..., \int_k, \mathcal{G}_1, ..., \mathcal{G}_k,$$

where $\int_i$ denotes the integrator connected to $\mathcal{G}_i$, for $i = 1, ..., k$. Therefore, the circuit indicated in figure 5.3 is a FF-GPAC. ∎

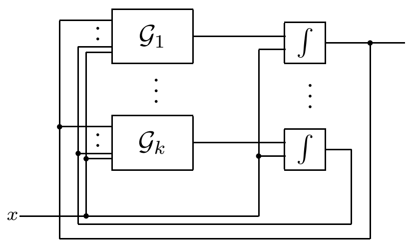

Figure 5.3: A circuit that solves integration.

6 Conclusion

We have shown that Shannon's GPAC and Pour-El's alternative definition still present too many problems in order for them to be considered as appropriate mathematical models for analog computers. Therefore, we have introduced a new model (the FF-GPAC) that preserves the desirable properties of the GPAC (namely, it is a model based on circuits that preserves equivalence with differentially algebraic functions) but that has a "nice behavior". We have further showed the relevance of this model by establishing full connections with a class of recursive functions over the reals.

Some questions can be considered. For example, can we introduce natural complexity measures in defining the FF-GPAC? Indeed, we saw in theorem 13 that the number of integrators in a FF-GPAC can be related to the order of the differentially algebraic function that it generates (a d.a. function of order n can only be generated by a FF-GPAC with at least n integrators). This result indicates that this task is apparently feasible, although possibly with some limitations.

The results proved in this paper can also be seen as a different route in order to establish connections with the classical theory of computation. In [3], [4], [5], several connections between recursive functions over the naturals and recursive functions over the reals are established. This work does not rely in the coding of Turing machines into some real numbers as in [8], [10], [15], and hence presents an alternative path to the traditional work in the field.

Acknowledgments. We would like to thank Cris Moore and Manuel Campagnolo for several useful discussions related to this work. In particular, we acknowledge M. Campagnolo for the idea that would led to the concept of $I_{\mathbb{R}}$-path.

This work was partially supported by FCT and FEDER via the Center for Logic and Computation - CLC, Laboratório de Modelos e Arquitecturas Computacionais - LabMAC, and is inserted in the FCT project ConTComp.

References

[1] V. Bush. The differential analyzer. A new machine for solving differential equations. *J. Franklin Institute*, 212:447-488, 1931.

[2] B. C. Carlson. *Special Functions of Applied Mathematics*, Academic Press, 1977.

[3] M. L. Campagnolo and C. Moore. Upper and lower bounds on continuous-time computation. In I. Antoniou, C. S. Calude, and M. J. Dinneen, editors, *Unconventional Models of Computation, UMC'2K*, DMTCS, pp. 135-153, Springer-Verlag, 2001.

[4] M. L. Campagnolo, C. Moore, and J. F. Costa. An analog characterization of the Grzegorczyk hierarchy. *Journal of Complexity*, to appear.

[5] M. L. Campagnolo, C. Moore, and J. F. Costa. Iteration, inequalities, and differentiability in analog computers. *Journal of Complexity*, 16(4):642-660, 2000.

[6] D. S. Graça. *The General Purpose Analog Computer and Recursive Functions Over the Reals*, MSc thesis, IST/UTL, 2002. (available online at `http://w3.ualg.pt/~dgraca`.)

[7] T. W. Hungerford. *Algebra*, Springer-Verlag, 8th printing, 1996.

[8] P. Koiran, M. Cosnard, and M. Garzon. Computability with low-dimensional dynamical systems. *Theoretical Computer Science*, 132:113-128, 1994.

[9] L. Lipshitz and L. A. Rubel. A differentially algebraic replacement theorem, and analog computation. *Proceedings of the A.M.S.*, 99(2):367-372,1987.

[10] C. Moore. Unpredictability and undecidability in dynamical systems. *Physical Review Letters*, 64:2354-2357, 1990.

[11] C. Moore. Recursion Theory on the reals and continuous-time computation. *Theoretical Computer Science*, 162:23-44, 1996.

[12] P. Odifreddi. *Classical Recursion Theory*, Elsevier, 1989.

[13] M. B. Pour-El. Abstract computability and its relations to the general purpose analog computer. *Transactions Amer. Math. Soc.*, 199:1-28, 1974.

[14] C. Shannon. Mathematical theory of the differential analyzer. *J. Math. Phys. MIT*, 20:337-354, 1941.

[15] H. T. Siegelmann. *Neural Networks and Analog Computation: Beyond the Turing Limit*, Birkhäuser, 1999.

[16] D. W. Stroock. *A Concise Introduction to the Theory of Integration*, Birkhäuser, 3rd edition,1999.