Some Characterizations of Functions Computable in On-Line Arithmetic

Jean-Michel Muller

Abstract—**After a short introduction to on-line computing, we prove that the functions computable in on-line by a finite automaton are piecewise affine functions whose coefficients are rational numbers (i.e., the functions $f(x) = ax + b$, or $f(x, y) = ax + by + c$ where a, b, and c are rational). A consequence of this study is that multiplication, division, and elementary functions of operands of arbitrarily long length cannot be performed using bounded-size operators.**

Index Terms—**Computer arithmetic, finite automata, on-line arithmetic.**

I. Introduction

On-line arithmetic was introduced in 1977 by Ercegovac and Trivedi. During on-line computations, the operands and the results flow through arithmetic units serially, digit by digit, starting from the most significant digit. An important characteristic of an on-line algorithm is its *delay*, i.e., the number δ such that the jth digit of the result is generated after the entrance of the $j+\delta$th digit of the operands.

On-line arithmetic enables a digit-level pipelining, which makes it possible to compute large expressions in an efficient manner. A consequence of the flow from the most significant digit to the least significant one is the need to use a redundant number system. Although on-line arithmetic would have been possible using *carry-save* representation of numbers, the number systems used in the literature are Avizienis' *signed-digit* systems [1].

A lot of on-line algorithms have been proposed, in the literature, for arithmetic operations [2]–[6] and elementary functions [7], [8]). An overview is given in [9]. Some authors have studied general properties of on-line arithmetic: For instance, Owens [10] points out several limitations of on-line arithmetic (he shows that fixed-point multiplication and division cannot be performed without assuming restrictions—e.g., quasi-normalization for division—on the values of the input operands), then he gives solutions to partially overcome these limitations. Sips and Lin [11] give a model for on-line arithmetic that makes it possible to evaluate tight bounds on the minimum delay of an arbitrary arithmetic function, and to perform on-line computations by table look-up. Independently, Duprat, Herreros, and Muller [12] found similar bounds on on-line delays. In a more general and theoretical context (computations with infinite objects), Wiedmer [13] gives some results on "approximately computable functions," which are applicable to on-line calculations (for example, he shows that a function computable in on-line arithmetic is continuous). Vuillemin [14] represents computable real numbers by continued fractions. He gives serial algorithms for computing sums and products of such fractions. His arithmetic has properties very similar to those of on-line arithmetic.

When designing algorithms and arithmetic units for on-line computations, the different functions of one or more variables computable in on-line seem to be classifiable in three different classes:

Manuscript received May 2, 1991; April 7, 1992. This work is partially supported by the PRC Architectures de Machines Nouvelles of the French Ministère de la Recherche et de l'Espace.

The author is with CNRS, Laboratoire LIP-IMAG, Ecole Normale Supérieure de Lyon, France.

IEEE Log Number 9209025.

1) First Class: Functions that can be computed in on-line using an operator of bounded size (i.e., independent from the length of the input operands). Addition, maximum or minimum of two numbers, absolute value, multiplication *by a constant integer*, division *by a constant integer* belong to this class. To our knowledge, it has not been clearly stated that division by a constant integer may be performed using an operator of bounded size, but it is a consequence of the fact that when performing the following "parallel-serial" division algorithm (we divide $x = 0.x_1x_2x_3x_4\cdots$ by y, and we obtain the quotient digits q_i serially):

$$x^{(0)} = 0..x_1x_2$$
$$x^{(n+1)} = 2x^{(n)} + \tfrac{1}{4}x_{n+3} - q_n y$$
$$q_n = \begin{cases} 1 & \text{if } x^{(n)} > 0 \\ -1 & \text{if } x^{(n)} < 0 \\ 0 & \text{if } x^{(n)} = 0. \end{cases}$$

The intermediate values $x^{(i)}$ are multiples of 1/4 and satisfy $-y + 1/4 \le x^{(i)} \le y - 1/4$; therefore their storage and their computation only require a bounded space.

A consequence of this remark is that affine functions (i.e., functions of the form $f(x, y) = \alpha x + \beta y + \gamma$) whose coefficients are rational numbers belong to this class.

2) Second Class: Functions that can be computed in on-line using an operator whose size is proportional to the length of the operands. Squaring, multiplication, division, and square-root (and the expressions obtained by combining these functions) belong to this class.

3) Third Class: Functions for which the known operators have a size that grows more than linearly with the size of the operands. For instance, the trigonometric functions belong to this class (as an example, the shift-and-add Cordic-like methods seem to need operators of size proportional to the length of the operands, but they need the storage of constants, and the space occupied by these constants is proportional to the square of the length of the operands).

This separation of functions into three classes is purely empirical. Our purpose is to give the beginning of a theoretical explanation of this separation by delimiting the first class. In the following, a bounded-size on-line operator of delay δ is represented by a transducer, i.e., a finite automaton (Fig. 1) which computes, at step n, its new state (i.e., its new "memory configuration") e_{n+1} and its new output digit $y_{n-\delta}$ from its previous state e_n and its input digit x_n (or its input digits $x_{n,1}; \cdots; x_{n,p}$ if we compute a function of p variables). The state e_n is assumed to belong to a *finite* state set. Since the number system we use is redundant, a given number x can be represented by several different digit chains. If these claims are fed into the automaton, the output chains can be different, but they must represent the same number. Thus we do not deal with any automaton, but only with those who satisfy this requirement, i.e., with transducers whose inputs and outputs can be interpreted as numbers.

We show in this brief contribution that if a function f is computable by an on-line finite automaton, and if f has a piecewise continuous second derivative (i.e., if f is "sufficiently regular"), then f is a piecewise affine function whose coefficients are rational numbers. As a consequence, it is not possible to perform on-line multiplication or division, or to compute square roots of arbitrarily long length numbers using an operator of bounded size (i.e., an operator independent of the length of the input operands).

In the following, numbers are represented using a fixed-point radix-r sign-digit system. The digits belong to the digit set $D_1 =$

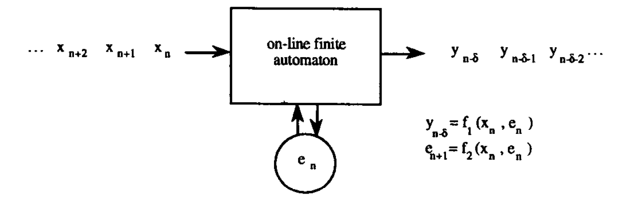

Fig. 1. An on-line finite automaton.

$\{-a, -a+1, \cdots, 0, 1, \cdots, a\}$, with $0 \leq a \leq r-1$. We also assume that the numbers manipulated here are elements of the interval [0, 1] (however, the results shown in this brief contribution are obviously valid in any interval). The digit chain $0.x_1x_2x_3\cdots x_i\cdots x_i \in D_a$, represents the number $\Sigma_{i=1}^{\infty} x_i r^{-i}$. If d is a digit, we denote $\overline{d}$ the digit equal to $-d$.

II. Functions of One Variable

In this section, we show that if a function of one variable has piecewise continuous second derivatives, and if this function is computable in on-line by a finite automaton, then it is an affine function with rational coefficients. As a consequent, functions like $x^2, \mathrm{sqrt}\,(x), \sin\,(x), \cdots$, cannot be computed using an on-line operator of bounded size.

First of all, we need to prove that if a function f is computable in on-line by a finite automaton and derivable, then in any interval there exist two distinct points x and y such that $f'(x) = f'(y)$, where f' is the first derivative of f. This result is obtained from the two following lemmas.

Lemma 1: Let α and β be two numbers satisfying $0 \leq \alpha < \beta \leq 1$. There exist an integer p and a p-digit chain $0.x_1x_2x_3\cdots x_p$ such that the interval $[0.x_1x_2x_3\cdots x_p\overline{aaaaa}\cdots, 0.x_1x_2x_3\cdots x_p aaaaa\cdots]$ (i.e., the set of the numbers representable with $x_1, x_2, x_3, \cdots, x_p$ as first p digits) is included in $[\alpha, \beta]$. □

Proof: Define

$$\begin{cases} m = \dfrac{\alpha+\beta}{2} = 0.m_1m_2m_3m_4\cdots, \\ d = \beta - \alpha, \end{cases}$$

we have obviously:

$$\begin{cases} \beta - m = m - \alpha = \dfrac{d}{2}, \\ \text{for any } i, |m - 0.m_1m_2m_3\cdots m_i| \leq \displaystyle\sum_{k=i+1}^{\infty} ar^{-k} = \dfrac{ar^{-i}}{r-1} \leq r^{-i}. \end{cases}$$

Let p be an integer satisfying $r^{-p} \leq d/4$. From the relations:

$$\begin{cases} 0.m_1m_2m_3\cdots m_p - 0.m_1m_2m_3\cdots m_p\overline{aaaaa}\cdots \leq r^{-p} \leq \dfrac{d}{4}, \\ 0.m_1m_2m_3\cdots m_p aaaaa\cdots - 0.m_1m_2m_3\cdots m_p \leq r^{-p} \leq \dfrac{d}{4}, \end{cases}$$

we deduce easily:

$$\begin{aligned} 0.m_1m_2\cdots m_p aaaaa\cdots &\leq 0.m_1m_2\cdots m_p \\ &\quad + \frac{d}{4} \leq m + \frac{d}{2} = \beta, \\ 0.m_1m_2\cdots m_p\overline{aaaaa}\cdots &\geq 0.m_1m_2\cdots m_p \\ &\quad - \frac{d}{4} \geq m - \frac{d}{2} = \alpha. \end{aligned}$$

Then the lemma is proved, by choosing $0.x_1x_2\cdots x_p = 0.m_1m_2\cdots m_p$. □

Lemma 2: Let α and β be two real numbers satisfying $0 \leq \alpha < \beta \leq 1$. Let f be a derivable function of one variable, computable in on-line mode by a finite automaton. There exist two distinct real numbers x and $z \in [\alpha, \beta]$, such that $f'(x) = f'(z)$. □

Proof: Assume that f is computable in on-line with delay δ by a finite automaton, and let M be the number of states of the automaton. Since, from Lemma 1, there exist an integer p and a p-digit chain such that each number whose representation begins with this digit chain belongs to $[\alpha, \beta]$, we assume in the sequel of the proof that all the numbers taken into account here begin with this digit chain. Then we renumber the digits: the $p+1$st digit of a real number x is called x_1, and we ignore the previous entrances of digits in the automaton (they are the same for every number): this $p+1$st digit is now considered the first one, and the step when it enters the automaton is now considered as the first one.

So, at step 1, the automaton receives the first digit x_1 of the input data. There are $2a+1$ possible values for this digit, namely $\overline{a}, \overline{a}+1, \overline{a}+2, \cdots, a-1, a$. The number constituted by this digit (i.e., the real number $0.x_1$) also has $2a+1$ possible values, $-a/r, (-a+1)/r, \cdots, +a/r$.

At step 2, the automaton receives the second digit x_2 of the input data. There are $(2a+1)^2$ possible input chains x_1x_2; however, there are only $2ar+2a+1$ different numbers representable by these digits, namely the numbers:

$$\frac{-ar-a}{r^2}, \frac{-ar-a+1}{r^2}, \frac{-ar-a+2}{r^2}, \cdots, \frac{ar+a}{r^2}.$$

This difference is due to the redundancy of the signed-digit representation (for instance, in radix 2, $0.1\overline{1} = 0.01$).

At step n, the automaton receives the nth digit x_n of the input data. There are $(2a+1)^n$ possible input chains $x_1x_2x_3\cdots x_n$, and $2ar^{n-1}+2ar^{n-2}+\cdots+2ar^2+2ar+2a+1 = 2a(r^n-1)/(r-1)+1$ possible numbers $0.x_1x_2x_3\cdots x_n$. Let n be an integer such that the number $2a(r^n-1)/(r-1)+1$ of possible n-digit numbers is greater than the number M of states of the automaton. Then there exists at least a state ϕ, two distinct n-digit numbers x and z, with signed-digit representations $0.x_1x_2\cdots x_n$ and $0.z_1z_2\cdots z_n$, such that the entering of the chain $x_1x_2x_3\cdots x_n$ in the automaton produces at step n the state $e_{n+1} = \phi$, as well as the entering of the chain $z_1z_2z_3\cdots z_n$.

Now, assume that after the chains $x_1x_2x_3\cdots x_n$ or $z_1z_2z_3\cdots z_n$, for both numbers the same sequence $t_{n+1}t_{n+2}t_{n+3}t_{n+4}\cdots$ of digits enters the automaton. Since for both computations the state e_{n+1} at step n is the same, we deduce that the outputs $y_{n+1-\delta}, y_{n+2-\delta}, y_{n+3-\delta}, \cdots$ and the states $e_{n+2}, e_{n+3}, e_{n+4}, \cdots$ will be the same.

Let us denote:

$$\begin{aligned} x &= 0.x_1x_2x_3\cdots x_n\,000000\cdots \\ x' &= 0.x_1x_2x_3\cdots x_n t_{n+1}t_{n+2}t_{n+3}t_{n+4}\cdots \\ z &= 0.z_1z_2z_3\cdots z_n\,000000\cdots \\ z' &= 0.z_1z_2z_3\cdots z_n t_{n+1}t_{n+2}t_{n+3}t_{n+4}\cdots \\ t &= 0.000\cdots 0t_{n+1}t_{n+2}t_{n+3}t_{n+4}\cdots \end{aligned}$$

Since the first n digits of x and x' (of z and z', respectively) are the same, if x or x' (z or z', respectively) enter the automaton, then the first $n-\delta$ digits of the result, namely $y_1, y_2, \cdots, y_{n-\delta}$ are the same. Therefore, $f(x') - f(x)$ only depends upon the digits $y_{n+1-\delta}, y_{n+2-\delta}, y_{n+3-\delta}, \cdots$, therefore $f(x') - f(x) = f(z') -$

$f(z)$, therefore:

$$\frac{f(x') - f(x)}{0.0000\cdots 0t_{n+1}t_{n+2}t_{n+3}t_{n+4}\cdots} = \frac{f(z') - f(z)}{0.0000\cdots 0t_{n+1}t_{n+2}t_{n+3}t_{n+4}\cdots}$$

Since this last result holds for any value of the digits $t_{n+1}, t_{n+2}, t_{n+3}, \cdots$, we deduce that for any positive real number ϵ sufficiently small (we take $\epsilon = 0.0000\cdots 0t_{n+1}t_{n+2}t_{n+3}t_{n+4}\cdots$):

$$\frac{f(x+\epsilon) - f(x)}{\epsilon} = \frac{f(z+\epsilon) - f(z)}{\epsilon}$$

Therefore, by taking the limit value of these terms when ϵ goes to zero:

$$f'(x) = f'(z). \qquad \square$$

Theorem 1: Let f be a function of one variable, with a piecewise continuous second derivative. If f is computable in on-line mode by a finite automaton, then in each interval where f is continuous, f is an affine function of the form $f(x) = ax + b$, where a and b are *rational numbers* . □

Proof: Let I be an interval where f has a continuous second derivative. First of all, we show that in this interval, f is an affine function, i.e., that there exist a and b such that for any $x \in I, f(x) = ax+b$ (after we will prove that a and b are rational numbers). In order to show this, it suffices to prove that $f''(x) = (d^2f/dx^2)(x) = 0$ for any $x \in I$.

Let N be an integer. Let us divide I into N subintervals of equal sizes (if $I = [c, d]$ then the kth interval is $[c+(k-1)(d-c/N), c+k(d-c)/N)]$). From Lemma 2, in each of these subintervals, there exist two distinct real numbers x and z satisfying $f'(x) = f'(z)$. Since f'' is continuous, we deduce that there exists a number $t\,(t \in [x, z])$ such that $f''(t) = 0$.

We have shown that if we divide I in N subintervals of equal sizes (N may be as large as wanted), in each of these intervals there exists a point where f'' equals zero. Now, since f is continuous, by taking $N \to \infty$, we deduce that $f''(x) = 0$ for any $x \in I$. Thus, there exist two numbers a and b such that for any $x \in I, f(x) = ax + b$.

Now, we have to show that a and b are rational numbers. We start from the well-known fact that, in a classical nonredundant numeration system, the rational numbers are the numbers whose representation is eventually periodic. Let M be the number of possible states of the automaton. Assume that an eventually periodic sequence of period p enters the automaton. After at most M entrances of the beginning of the periodic part, the state of the automaton will be equal to a state already encountered during an entrance of the beginning of the periodic part. Therefore, the output is also eventually periodic, with a period bounded by Mp.

A surprising property of signed-digit systems is that a rational number may have an aperiodic representation: if a number is equal to $0.\alpha\alpha\alpha\alpha\cdots$, where α is a representation of the period, and if there exists another representation β of this period, then $0.\beta\alpha\beta\alpha\alpha\beta\alpha\alpha\alpha\beta\alpha\alpha\alpha\alpha\beta\cdots$ is aperiodic and represents the same number. However a rational number has also eventually periodic representations.

- In radix 2, it is obvious since its classical nonredundant representation is also a representation using the digit set $\{-1, 0, 1\}$.
- In radix $r > 2$, let us start from the nonredundant radix- r representation of this number, say $x = 0.x_1x_2x_3x_4\cdots$. Since x is rational, this representation is eventually periodic. If the signed-digit number system used is *maximally redundant* (i.e., if $a = r - 1$), then this representation is a signed-digit representation too. If $a < r - 1$, we can obtain an eventually periodic signed-digit representation of x in the number system we deal with as the following.
- By subtracting a to each digit x_i of x, we obtain an eventually periodic signed-digit representation of $x - 0.aaaaaaaa\cdots$.
- By adding to this last result the number $0.aaaaaaaa\cdots$ using Avizienis's algorithm [1], we obtain an eventually periodic signed-digit representation of x in the number system we deal with.

Let r_1 be a rational number. If an eventually periodic representation of r_1 enters the automaton, then, since the output is also eventually periodic, $f(r_1) = s_1$ is a rational number too. Let us consider $f(r_2) = s_2$, where r_2 is another rational number. From the relations $ar_1+b = s_1$ and $ar_2+b = s_2$, we deduce that $s = (a_1-s_2)/(r_1-r_2)$ and $b = (s_2r_1 - s_1r_2)/(r_1 - r_2)$ are rational numbers. □

Now, we can delimit a little more carefully the set of the functions computable in on-line using a finite automaton. We saw previously that if such a function has piecewise second derivatives, then it is a piecewise affine function. Let us show that the points where the second derivative of this function is not continuous are *rational* numbers.

Theorem 2: Let f be a function of one variable, with a piecewise continuous second derivative. If f is computable in on-line mode by a finite automaton, then the *breakpoints* of f, i.e. the points where the second derivative of f is not continuous, are rational numbers.□

Proof: From Theorem 1, we deduce that f is a piecewise affine function whose coefficients are rational. In the neighborhood of a breakpoint α of f, at the left of α, f is equal to an affine function, say $ax + b$, at the right of α, f is equal to another affine function $cx + d$, where a, b, c, and d are rational numbers. From previous studied [12], [13] we know that f is a continuous function. Therefore, $a\alpha+b = c\alpha+d$. Therefore $\alpha = (d-b)/(a-c)$ is a rational number.□

It is possible to show that a continuous piecewise affine function with one rational breakpoint is computable in on-line by a finite automaton. Let f be such a function. We keep the same notations (values α, a, b, c, and d) as in the proof of Theorem 2. Define a function $g(x) = f(x) = cx - d$. Since $g(x)$ is computable using a finite automaton (we can compute affine functions with rational coefficients, and minimum of two numbers, and $g(x)$ is equal to $(a-c)(\min(x, \alpha)) + (b-d)$, we deduce that $f(x) = g(x) + cx + d$ is computable in on-line using a finite automaton. Thus, we have proved that a continuous piecewise affine function with one rational breakpoint is computable by an on-line finite automaton. This can be extended to a function with a *finite* number of rational breakpoints.

III. Functions of Two Variables

Now, we are going to extend Theorem 1 to functions of two variables. The proof may be generalized to functions of more than two variables. A consequence of the following result is that arithmetic operations like multiplication or division cannot be computed in on-line using a bounded-size operator.

Theorem 3: Let f be a function of two variables, with continuous second derivatives $(\partial^2 f)/(\partial x^2)$, $(\partial^2 f)/(\partial y^2)$, $(\partial^2 f)/(\partial x\partial y)$, and $(\partial^2 f)/(\partial y\partial x)$ in rectangles. If f is computable in on-line mode by a finite automaton, then in each rectangle where f'' is continuous, f is an affine function of the form $f(x) = \beta x + \gamma y + \delta$ where β, γ, and δ are *rational numbers*. This result can be generalized to functions of more than two variables. □

Proof: Let us consider a rectangle where the second derivatives $(\partial^2 f)/(\partial x^2), (\partial^2 f)/(\partial y^2), (\partial^2 f)/(\partial x\partial y), and (\partial^2 f)/(\partial y\partial x)$ are continuous. For any *rational number* y_0, on the straight line $y = y_0$, the function $f_{y_0}(x) = f(x, y_0)$ satisfies the conditions of Theorem 1. Therefore $f(x, y_0) = a(y_0)x + b(y_0)$, where $a(y_0)$ and $b(y_0)$ are rational numbers. It is worth noticing that this is not necessarily true if y_0 is irrational: Theorem 1 can be applied if and only if a finite automaton with two entries (namely x and y_0), which computes a function of x and y_0 can be assimilated to a finite automaton with one entry (namely x), which computes the function $f_{y_0}(x)$. It is true if y_0 is a rational number, since the entrance of this number may be replaced by the knowledge of a finite amount of information (e.g., the periodic part of its representation), while it may be false if y_0 is an irrational number: For instance, the function $f_\pi(x) = x+\pi$ cannot be computed by a finite automaton, while the function $f(x,y) = x+y$ can be computed by an on-line finite automaton at point (x, π).

However, since for any *rational* number $y_0, (\partial^2 f)/(\partial x^2)(x, y_0) = 0$, and since $\partial^2 f/\partial x^2$ is a continuous function, we deduce that for any **real** number $y_0, (\partial^2 f/\partial x^2)(x, y_0) = 0$. Therefore, for any real number $y_0, f(x, y_0) = a(y_0)x + b(y_0)$ (but $a(y_0)$ and $b(y_0)$ are not necessarily rational numbers).

Now, let us apply Theorem 1 to the computation of $f(x_0, y) = a(y)x_0 + b(y)$ for a given rational number x_0. We deduce:

$$\frac{\partial^2 f}{\partial y_2}(x_0, y) = a''(y)x_0 + b''(y) = 0.$$

Since this relation holds for any rational number x_0 (and in practice for any real number x_0 since $\partial^2 f/\partial y^2$ is continuous), we deduce that for any $y, a''(y) = b''(y) = 0$. Therefore, there exist four real numbers α, β, γ, and δ satisfying:

$$\begin{cases} a(y) = \alpha y + \beta, \\ b(y) = \gamma y + \delta. \end{cases}$$

Therefore, in the rectangle that we considered: $f(x,y) = \alpha xy + \beta x + \gamma y + \delta$.

Now, since the function $\phi(x) = f(x,x) = \alpha x^2 + (\beta+\gamma)y + \delta$ satisfies the conditions of Theorem 1, ϕ is an affine function, therefore α is equal to zero. As in the proof of Theorem 1, from the fact that if eventually periodic sequences enter the automaton then the output is eventually periodic too, we deduce that if x and y are rational numbers, then $f(x,y)$ is rational too. An obvious consequence of this is that β, γ, and δ are rational numbers. □

IV. Conclusion

We have proved here that if a (sufficiently regular) function is computable in on-line using a bounded size operator, then this function is a piecewise affine function whose coefficients are rational. Therefore, although it is possible to design, for instance, infinite precision on-line adders, it is impossible to design infinite precision on-line multipliers or dividers.

Hence, we have delimited the first class of functions presented in the introduction. Since multiplication, division, and square root (and more generally nth root) belong to the second class, we conjecture that this class is the set of the algebraic functions. However, we did not succeed in proving that.

References

[1] A. Avizienis, "Signed-digit number representations for fast parallel arithmetic," *IRE Trans. Electron. Comput.*, vol. 10, pp. 389–400, 1961.

[2] M. D. Ercegovac and K. S. Trivedi, "On line algorithms for division and multiplication," *IEEE Trans. Comput.*, vol. C-26, no. 7, pp. 681–687, July 1977.

[3] M. D. Ercegovac and P. K. G. Tu, "A radix-4 on-line division algorithm," presented at *8th Symp. Comput. Arithmetic*, Como, Italy, May 1987.

[4] M. J. Irwin and R. M. Owens, "Digit-pipelined arithmetic as illustrated by the paste-up system: A tutorial," *IEE Trans. Comput.*, vol. 36, pp. 61–73, Apr. 1987.

[5] M. J. Irwin, "An arithmetic unit for Online computation," Ph.D. dissertation, Dept. of Comput. Sci., Univ. of Illinois, Champaign-Urbana, IL; Tech. Rep. UIUCDCS-R-77-873, 1977.

[6] J. G. Rusnak and K. S. Trivedi, "Higher radix on-line division," in *Proc. 4th IEEE Symp. Comput. Arithmetic*, Oct. 1978, pp. 183–189.

[7] M. D. Ercegovac, "An on-line square rooting algorithm," in *Proc. 4th IEEE Symp. Comput. Arithmetic*, Oct. 1978, pp. 183–189.

[8] R. M. Owens, "Digit on-line algorithms for pipeline architectures," Ph.D. dissertation, Dept. of Comput. Sci., The Pennsylvania State Univ., University Park, Aug. 1980.

[9] M. D. Ercegovac, "On-line arithmetic: An overview," *SPIE*, vol. 495, *Real Time Signal Processing* VII, 1984, pp. 86–93.

[10] R. M. Owens, "Techniques to reduce the inherent limitations of fully digit on-line arithmetic," *IEEE Trans. Comput.*, vol. C-32, no. 4, Apr. 1983.

[11] H. J. Sips and H. X. Lin, "A new model for on-line arithmetic with an application to the reciprocal calculation," *J. Parallel Distrib. Computing*, vol. 8, pp. 218–230, Mar. 1990.

[12] J. Duprat, Y. Herreros, and J. M. Muller, "Some results about on-line computation of functions," presented at *9th Symp. on Comput. Arithmetic*, Santa Monica, CA, Sept. 1989.

[13] E. Wiedmer, "Computing with infinite objects," *Theorem. Comput. Sci.*, vol. 10, pp. 133–155, 1980.

[14] J. Vuillemin, "Exact real computer arithmetic with continued fractions," *IEEE Trans. Comput.*, vol. 39, no. 8, pp. 1087–1105, Aug. 1990.

On the Genus of Star Graphs

David A. Hoelzeman and Saïd Bettayeb

Abstract—**The star graph has recently been suggested as an alternative to the hypercube. The star graph has a rich structure and symmetry properties as well as desirable fault-tolerant characteristics. The star graph's maximum vertex degree and diameter, viewed as functions of network size, grow less rapidly than the corresponding measures in a hypercube. We investigate the genus of the star graph and compare it with the genus of the hypercube.**

Index Terms—**Genus, hypercube, permutation, rotational embedding, star network.**

I. Introduction

The star graph was proposed by Akers *et al.* as an interconnection network for parallel computation [1], [2]. The star graph belongs to a class of graphs called *Cayley graphs*, a family of graphs that possesses group theoretic properties. The star graph's maximum vertex degree and diameter, viewed as functions of network size, grow less rapidly than the corresponding measures in the binary hypercube. The star graph also has a rich structure and symmetry properties as well as desirable fault-tolerant characteristics [1]. The star graph $G_n = (S_n, E_n)$ of dimension n is defined as follows: $S_n = \{P | P$ is a permutation of $\{1, 2, \cdots, n\}\}$ and $E_n = \{(P, Q) |$ there is a transposition $(1 \quad k)$ for $2 \leq k \leq n$ such that $Q = P(1 \quad k)\}$. Fig. 1 illustrates star graphs G_2 and G_3, and Fig. 2 depicts G_4.

Manuscript received July 6, 1992, revised May 28, 1993. This work was supported in part by the Louisiana Education Quality Support Fund under Grant LEQSF (90-92)-RD-A-01.

D. A. Hoelzeman was with Louisiana State University, Baton Rouge, LA 70803 USA. He is now with the Department of Computer Science, Southeastern Oklahoma State University, Durant, OK 74701 USA.

S. Bettayeb is with the Deparatment of Computer Science, Louisiana State University, Baton Rouge, LA 70803 USA.

IEEE Log Number 9213775.