*Complex Systems* **1** (1987) 211–216

# Fast Computation of Additive Cellular Automata

**Arch D. Robison**
*Department of Computer Science, University of Illinois,*
*1304 West Springfield Avenue,*
*Urbana, IL 61801, USA*

**Abstract.** Direct simulation of an additive cellular automaton takes a time $O(t^2)$ to compute an arbitrary site value $t$ time steps into the future. For the case of a single initial nonzero site, the problem is equivalent to computing a coefficient residue of a polynomial power. An algorithm is derived which computes an arbitrary site's value in time $O(\log t)$ .

## 1. Introduction

A cellular automaton consists of a row of cells which change state over time [1]. The value of a site at position $i$ and time $t$ is denoted $a_i^{(t)}$ . An additive cellular automaton [2] has a rule of the form:

$$a_i^{(t)} = \sum_j s(j) a_{i-j}^{(t-1)} \mod m \tag{1.1}$$

where $s$ specifies the rule. If the automaton's sites are viewed as coefficients of a polynomial, then each row is obtained by multiplying the previous row by a rule polynomial. Since polynomial multiplication is associative, the problem reduces to computing powers of the rule polynomial. Let $A^{(t)}(x)$ and $S(x)$ be the automaton state and rule polynomials respectively.

$$A^{(t)}(x) = \sum_i a_i^{(t)} x^i \tag{1.2}$$

$$S(x) = \sum_i s_i x^i \tag{1.3}$$

Then the state of the automaton after $t$ time steps is given by

$$A^{(t)}(x) = A^{(0)}(x) S^t(x) \mod m \tag{1.4}$$

Via the Chinese Remainder Theorem, our problem reduces to computing solutions for moduli which are powers of primes, i.e. $m = p^\gamma$ . The rest of this paper develops an algorithm for quickly computing any $a_i^{(t)}$ for the case $m = p^\gamma$ and $A^{(0)}(x) = 1$.

## 2. Notation

All polynomials in this paper are formal power series; the powers of $x$ are placeholders only. A polynomial $Q(x)$ is written

$$Q(x) = \sum_{i=r_Q}^{l_Q} q_i x^i \tag{2.1}$$

where $r_Q$ and $l_Q$ are the minimum and maximum degrees of the terms. The terms may have negative degree. We define the width $w_Q$ of polynomial $Q$ as $w_Q = l_Q - r_Q$ . Subscripts are omitted where only a single polynomial is under consideration. The notation

$$Q(x) \equiv Q'(x) \mod m \tag{2.2}$$

means that the coefficients of $Q(x)$ and $Q'(x)$ are congruent, i.e.

$$q_i \equiv q'_i \mod m \tag{2.3}$$

for all integers $i$.

## 3. Self-Similar Polynomials

We call a polynomial $Q$ with integer coefficients *self-similar* $\bmod m$ if there exists a *scaling exponent* $\beta$ such that:

$$Q^\beta(x) \equiv Q(x^\beta) \mod m. \tag{3.1}$$

This section will show that self-similar polynomials may be generated for moduli of the form $p^\gamma$, where $p$ is prime and $\gamma$ is a positive integer. Self-similar polynomials are the key to the algorithm, since exponentiating such polynomials is much easier than exponentiating arbitrary polynomials.

**Lemma 1.** *The sum of self-similar polynomials* $\bmod p$ *is self-similar for scaling exponent* $p$. *Thus, given a prime* $p$ *and self-similar polynomials* $Q(x)$ *and* $R(x)$:

$$[Q(x) + R(x)]^p \equiv Q(x^p) + R(x^p) \mod p. \tag{3.2}$$

*Proof.*

$$[Q(x) + R(x)]^p \equiv Q^p(x) + \sum_{i=1}^{p-1} \binom{p}{i} Q^i(x) R^{p-i}(x) + R^p(x) \tag{3.3}$$

The terms of the summation vanish because [3]

$$\binom{p}{i} \equiv 0 \mod p \quad 1 \le i \le p-1 \tag{3.4}$$

which leaves us with

$$[Q(x) + P(x)]^p \equiv Q^p(x) + R^p(x) \equiv Q(x^p) + R(x^p) \mod p \quad \blacksquare \tag{3.5}$$

**Lemma 2.** *Monomials are self-similar* mod $p$ *with a scaling exponent of* $p$, *so that given a prime* $p$ *and monomial* $Q(x) = ax^n$,

$$Q^p(x) \equiv Q(x^p) \mod p. \tag{3.6}$$

*Proof.* This follows immediately from Fermat's little theorem:

$$Q^p(x) = a^p x^{np} \equiv ax^{np} = Q(x^p) \mod p \quad \blacksquare \tag{3.7}$$

**Theorem 1.** *All polynomials are self-similar* mod $p$ *for scaling exponent* $p$, *so that given prime* $p$ *and polynomial* $Q(x)$,

$$Q^p(x) = Q(x^p) \mod p. \tag{3.8}$$

*Proof.* Since monomials are self-similar, their sum $Q(x)$ is also self-similar. ■

**Theorem 2.** *All polynomials of the form* $Q^{p^{\gamma-1}}(x)$ *are self-similar* mod $p^\gamma$ *for scaling exponent* $p$, *so that given prime* $p$, *polynomial* $Q(x)$, *and non-negative integer* $\gamma$,

$$\left[Q^{p^{\gamma-1}}(x)\right]^p \equiv Q^{p^{\gamma-1}}(x^p) \mod p^\gamma. \tag{3.9}$$

*Proof.* (Induction on $\gamma$.) We have already shown that the theorem is true for $\gamma = 1$. Assuming that the theorem is true for $\gamma' = \gamma - 1$:

$$Q^{p^{\gamma-1}}(x) \equiv Q^{p^{\gamma-2}}(x^p) \mod p^{\gamma-1} \tag{3.10}$$

Then there must exist a polynomial $R(x)$ such that:

$$Q^{p^{\gamma-1}}(x) \equiv Q^{p^{\gamma-2}}(x^p) + p^{\gamma-1}R(x) \mod p^\gamma \tag{3.11}$$

$$\begin{aligned} \left[Q^{p^{\gamma-1}}(x)\right]^p &\equiv Q^{p^{\gamma-1}}(x^p) + \sum_{j=1}^{p-1} \binom{p}{j} \left(p^{\gamma-1}\right)^j Q^{p-j}(x) R^j(x) \\ &\quad + \left(p^{\gamma-1}\right)^p R^p(x) \mod p^\gamma \end{aligned} \tag{3.12}$$

For $1 \leq j \leq p-1$,

$$\binom{p}{j} \left(p^{\gamma-1}\right)^j \equiv 0 \mod p^\gamma \tag{3.13}$$

which causes all terms in the sum to vanish. Finally, since $\gamma > 1$ and $p > 1$ the last term must also vanishes. ■

## 4. Computation of Powers of Self-Similar Polynomials

In this section, $Q(x)$ is a self-similar polynomial $\bmod m$ with scaling exponent $\beta$. Let $q(b,i)$ be the $i$ th coefficient of the expansion of $Q^b(x) \bmod m$, i.e.

$$Q^b(x) \equiv \sum_i q(b,i)x^i \mod m. \tag{4.1}$$

We show how to compute the $i$ th coefficient of $Q^b(x)$ in $O(\log b)$ time.

**Lemma 3.** *Given a table of $Q^k(x)$ for $0 \le k < \beta$, we can compute $Q^b(x)$ with $\log b/\log\beta$ polynomial multiplications (convolutions of coefficients).*

*Proof.* Define $k\, MOD\, m$ for integers $k, m$ as the least non-negative residue of $k$ modulo $m$. We can rewrite $b$ as

$$b = b\, MOD\, \beta + \beta \lfloor b/\beta \rfloor \tag{4.2}$$

$$Q^b(x) \equiv Q^{b\, MOD\, \beta}(x) Q^{\lfloor b/\beta \rfloor}(x^\beta) \mod m \tag{4.3}$$

Since $b$ is divided by $\beta$ on each application of the recurrence, we need apply the recurrence at most $\log b/\log\beta$ times. ■

**Theorem 3.** *If we compute $q(b,i)$ for $r_i \le i \le l_i$ by the convolutions in the lemma, and $l_i - r_i \le w$ , where $w$ is the width of $Q(x)$, then each convolution takes time*

$$O(\beta w \log w) \tag{4.4}$$

*Proof.*

$$q(b,i) \equiv \sum_j q(b\, MOD\, \beta, i - \beta j)\, q(\lfloor b/\beta \rfloor, j) \mod m \tag{4.5}$$

By considering the width of successive powers of $Q(x)$, we can see that $q(b,k)$ is zero for $k < l_Q b$ or $k > r_Q b$. Therefore $i - j\beta$ must be constrained as follows:

$$r_Q(\beta-1) \ge i - \beta j \ge l_Q(\beta - 1) \tag{4.6}$$

$$\frac{r_Q(\beta-1)+r_i}{\beta} = r_j \le j \le l_j = \frac{l_Q(\beta-1)+l_i}{\beta}. \tag{4.7}$$

From this we can show:

$$l_j - r_j \le w. \tag{4.8}$$

By induction we see that this bound holds for the recursive evaluations of $q(b,j)$. By Fourier methods, we can convolve two sequences of width $w$ in time $O(w \log w)$. The convolution as written is not an ordinary convolution in that the "traveling" subscripts change at different rates, so that the changing subscripts are $i - j\beta$ and $j$. We actually need to do $\beta$ ordinary convolutions, i.e. a convolution for each $i$ in $\{0, \ldots, \beta - 1\}$. Each convolution computes all $q(b, i + \beta k)$ for all $k$:

$$q(b, i+\beta k) = \sum_j q(b\, MOD\, \beta, i + \beta(k-j))\, q(\lfloor b/\beta \rfloor, j) \tag{4.9}$$

Thus we compute $\beta$ convolutions of width $w$. ■

**Lemma 4.** *Given a polynomial $P(x)$ of width $w$, we can compute the first $n$ powers of $P$ in time $O(nw^n \log w)$.*

*Proof.* We compute $P^k(x) = P(x)P^{k-1}(x)$. The width of $P^k(x)$ is $w^k + 1$. By use of the Fourier transform, the time to compute $P^k(x)$ from $P^{k-1}(x)$ is

$$O\left((w^k + 1)\log(w^k + 1)\right) = O\left(kw^k \log w\right). \tag{4.10}$$

The time to compute the first $n-1$ powers of $P(x)$ is

$$O\left(\sum_{k=1}^{n-1} kw^k \log w\right) \subset O\left((n-1)w^n\right). \tag{4.11}$$

The time to compute the $n$ th power of $P(x)$ is

$$O\left(nw^n \log w\right), \tag{4.12}$$

which dominates the computation time for the first $n-1$ powers of $P(x)$. ∎

**Theorem 4.** *We can compute $q(b,i)$ for $r_i \leq i \leq l_i$ where $l_i - r_i \leq w_Q$ in time*

$$O\left((w \log w)\frac{\beta}{\log \beta}\log b + \beta w^{\beta-1}\log w\right) \tag{4.13}$$

*Proof.* The first term is the product of the number of convolutions and operations per convolution. The second term is the table construction time. The table contains the first $\beta - 1$ powers of $Q(x)$, which are computed by the previous lemma. ∎

**Theorem 5.** *Given a polynomial $S(x)$ such that $S^\alpha(x)$ is self similiar $\bmod m$ with scaling exponent $\beta$, i.e.*

$$S^{\alpha\beta}(x) \equiv S^\alpha(x^\beta) \mod m \tag{4.14}$$

*we can compute coefficient $k$ of $S^t(x) \bmod m$ in time*

$$O\left(\alpha w \log(\alpha w)\frac{\beta}{\log \beta}\log t + \beta(\alpha w)^{\beta-1}\log(\alpha w) + \alpha w^{\alpha-1}\log w\right) \tag{4.15}$$

*Proof.* We can rewrite $S^t(x)$ as

$$S^t(x) = S^{t\,MOD\,\alpha}(x)\left[S^\alpha(x)\right]^{\lfloor t/\alpha \rfloor} \tag{4.16}$$

Let $Q(x) = S^\alpha(x)$. Note that the extreme degrees of $Q(x)$ are $l_Q = \alpha l_S$ and $r_Q = \alpha r_S$.

$$S^t(x) = S^{t\,MOD\,\alpha}(x)Q^{\lfloor t/\alpha \rfloor}(x) \tag{4.17}$$

Define $s(t,k)$ as the $k$ th coefficient of $S^t(x)$:

$$S^t(x) = \sum_i s(t,i)x^i \tag{4.18}$$

$$s(t,k) = \sum_i s\,(t\,MOD\,\alpha)\,q\,(\lfloor t/\alpha \rfloor, i). \tag{4.19}$$

Since $t\,MOD\,\alpha < \alpha$, we have the constraint

$$(\alpha - 1)l_S \leq k - i \leq (\alpha - 1)r_S \tag{4.20}$$

$$(\alpha - 1)l_S + k \geq i \geq (\alpha - 1)r_S + k \tag{4.21}$$

$$l_i - r_i \leq (l_S - r_S)(\alpha - 1) \leq (l_S - r_S)\alpha = l_Q - r_Q. \tag{4.22}$$

Therefore we can compute the necessary coefficients of $Q^{\lfloor t/\alpha \rfloor}(x)$ within the previously proven time bound. ∎

The latter two terms in the theorem are table construction times. The table $q$ contains the first $\beta - 1$ powers of $S^\alpha(x)$; the table $s$ contains the first $\alpha - 1$ powers of $S(x)$.

## 5. Evolution from a single site seed

Given rule $S(x)$ and a single-site seed $A(x) \equiv 1$, $a_k^{(t)} = s(t,k)$. By setting $\alpha = p^{\gamma-1}$ and $\beta = p$, we can use the previously derived algorithm to compute $s(t,k)$.

## Acknowledgements

The author thanks Stephen Wolfram for suggesting this problem. The author is supported by a Shell Fellowship in Computer Science.

## References

[1] Stephen Wolfram, *Theory and Applications of Cellular Automata*, (World Scientific Publishing Co., 1986).

[2] Olivier Martin, Andrew M. Odlyzko, and Stephen Wolfram, "Algebraic Properties of Cellular Automata", *Communications in Mathematical Physics*, **93** (1984) 219-258; reprinted in [1].

[3] Donald E. Knuth, *The Art Of Computer Programming*, vol. 1, (Addison-Wesley, 1973) p. 68.