Abstract

The set of unary functions of complexity classes satisfying simple constraints and defined by using concatenation recursion on notation is inductively characterized by means of concatenation iteration on notation. In particular, $AC^0$, $TC^0$, $NC^1$ and $NC$ unary functions are then inductively characterized using addition, composition and concatenation iteration on notation.

Keywords: concatenation recursion on notation, $AC^0$, $TC^0$, $NC^1$ and $NC$ computable functions.

MSC (2000) 03D20, 03D15.

The characterization of functions computable with bounded resources is an important issue of computational complexity. One of the main trends comes down from Computability Theory and concerns the inductive definition of complexity classes using a set of basic functions and functional operators like function substitution and weak forms of primitive recursion [16, 3].

In classical Computability Theory many efforts have been done to characterize the subset of unary (one-argument) functions of primitive and general recursive functions, see [13, 14, 7, 8, 4] and also [5, 6, 9, 10, 15] for recent results. This is motivated by the simple underlying data structure and by the fact that unary functions constitute a monoid with respect to function composition. Moreover, functions can be “simulated” by unary functions, i. e. any function of finite arity can be defined as the composition of a unary function with an encoding of its arguments.

On the other hand, for no subclass of the polynomial time computable functions a characterization of the unary functions has been given up to now (however, unary linear-space computable functions, unary polynomial-space computable functions and unary elementary functions have been characterized as the inductive closure of a finite set of functions with respect to the composition and the bounded primitive iteration operators [11]).

In this paper we characterize the set of unary functions of well known complexity classes, namely the class $AC^0$ of functions computable by poly-size, constant depth boolean circuits, the class $TC^0$ of functions computable by poly-size, constant depth threshold circuits, the class $NC^1$ of functions computable by poly-size, logarithmic depth boolean circuits and the class $NC$ of functions computable in polylogarithmic time by PRAMs with a polynomial number of
processors.  

Any of these classes has been defined as the closure of certain basic functions with respect to substitution, concatenation recursion on notation (CRN) and, for NC only, weak bounded recursion on notation [2, 3, 1].

Here, we consider concatenation iteration on notation (CIN), a simple variant of CRN, and redefine $AC^0, TC^0, NC^1$ and $NC$ as the closure of certain basic functions with respect to substitution, concatenation iteration on notation and, for NC only, weak bounded iteration.

Moreover, we define $AC^0, TC^0, NC^1$ and $NC$ unary functions as the closure of certain basic unary functions with respect to function addition, function composition, concatenation iteration on notation and, for $NC$ functions only, weak bounded iteration on notation.

After the Preliminaries, Section 2 introduces the pairing function used in this paper and recalls the method of simulation [6] for characterizing the unary functions of a given function set.

Section 3 states a series of properties concerning function classes closed with respect to CIN and, in Section 4, it is shown that such classes are closed with respect to CRN if extra functions are added.

Then, we obtain new characterizations of $AC^0, TC^0$ and $NC^1$ by specializing the results of Section 4. In Section 6, characterizations of $AC^0, TC^0$ and $NC^1$ unary functions are obtained by means of the simulation method.

Finally, in Section 7, new characterizations of $NC$ and of $NC$ unary functions are obtained by means of concatenation iteration on notation and weak bounded iteration.

1 Preliminaries

In this paper, we will only consider functions with finite arity on the set $N$ of natural numbers. Let $F_1$ be the set of unary functions of a function set $F$.

From now on, we agree that $x, y, z, u, v, w, i, j, n, m$ range over $N$, that $a, b$ range over positive integers, that $x, y$ range over sequences (of fixed length) of natural numbers and that $f, g, h$ (possibly with apices or indices) range over functions.

We will use the following unary functions: the successor function $S : x \mapsto x + 1$; the binary successor functions $s_0 : x \mapsto 2x$ and $s_1 : x \mapsto 2x + 1$; the identity function $id : x \mapsto x$; the constant functions $C_y : x \mapsto y$; the signum function
\[
sg : \begin{array}{c}
x+1 \\ 0
\end{array} \longmapsto \begin{array}{c}1 \\ 0
\end{array};
\]

the cosignum function
\[
\cosg : \begin{array}{c}
x+1 \\ 0
\end{array} \longmapsto \begin{array}{c}0 \\ 1
\end{array};
\]

the quadratum function $quad : x \mapsto x^2$; the division function $div_2 : x \mapsto \lfloor x/2 \rfloor$; the remainder function $rem_2 : x \mapsto x - 2 \lfloor x/2 \rfloor$; the length function $len : x \mapsto |x| = \lceil \log_2(x+1) \rceil$; the unary smash function $us : x \mapsto 2^{|x|} + 1$; the next power of two function $pow : x \mapsto 2^{|x|}$.

1 Names $AC^0, TC^0, NC^1$ and $NC$ are usually intended to denote language classes. However, in this paper they will always denote function classes, since no misunderstanding is possible.
Remember that $|x| = \lceil \log_2(x + 1) \rceil$ is the number of bits of the binary representation of $n$.

Furthermore, we will use the following operators on unary functions:

- the addition operator $f + g$ transforming functions $f, g$ into the function $f + g : n \mapsto fn + gn$;

- for any function $J : N^2 \to N$, the pairing operator $(f, g) J$ transforming functions $f, g$ into the function $(f, g) J : n \mapsto J(fn, gn)$;

- the conditional operator transforming functions $f, g, h$ into the function $\text{cond}(f, g, h) : n \mapsto gn$ if $fn = 0$, $hn$ otherwise;

- the composition operator $f \circ g$ transforming functions $f, g$ into the function $f \circ g : n \mapsto g(fn)$;

- the concatenation iteration on notation operator $\text{CIN}(h_0, h_1)$ transforming unary functions $h_0$ and $h_1$ with values in $\{0, 1\}$ into the function $f(0) = 0$, $f(x) = sh_i(x)(f(x))$ where $i \in \{0, 1\}$. The second equation holds for $x > 0$ when $i = 0$.

Moreover, recall that $f^n = id \circ f \circ \ldots \circ f$ denotes the $n$-th iterate of $f$, obtained by composing $n$ times function $f$ with id.

We will also use the following functions: the addition function $\text{add} : x, y \mapsto x + y$; the multiplication function $\text{mult} : x, y \mapsto xy$; the modified subtraction function $x, y \mapsto x - y = \max(x - y, 0)$; the conditional function $\text{cond}(x, y, z) = \{y \text{ if } x = 0, z \text{ otherwise}\}$;

the bit function $\text{bit} : x, y \mapsto \text{rem}_2(|x|2^y)$; the concatenation function $\text{conc} : x, y \mapsto x + y = x2^{|y|} + y$; the smash function $x, y \mapsto x\#y = 2^{|x|2^{|y|}}$; the most significant part function $\text{MSP} : x, y \mapsto \lfloor x/2^{|y|}\rfloor$; the least significant part function $\text{LSP} : x, y \mapsto x \mod 2^{|y|}$; the log most significant part function $\text{msp} : x, y \mapsto \lfloor x/2^{|y|}\rfloor$; the log least significant part function $\text{sp} : x, y \mapsto x \mod 2^{|y|}$; the front half function $\text{FH} : x \mapsto \lfloor x/2^{|x|/2}\rfloor$; the back half function $\text{BH} : x \mapsto x \mod 2^{|x|/2}$.

Finally, we will use the following operators on functions:

- the substitution operator $\text{SUBST}(g_1, \ldots, g_b, h)$ transforming functions $g_1, \ldots, g_b : N^a \to N$ and function $h : N^b \to N$ into the function $f : N^a \to N$ such that $f(x) = h(g_1x, \ldots, g_bx)$;
the concatenation recursion on notation operator \( CRN(g, h_0, h_1) \) transforming functions \( g : N^a \rightarrow N, \ h_0 : N^{a+1} \rightarrow N \) and \( h_1 : N^{a+1} \rightarrow N \) into the function \( f : N^{a+1} \rightarrow N \) such that

\[
\begin{align*}
  f(0, y) &= g(y), \\
  f(s_i(x), y) &= s_{h_i(x,y)}(f(x,y))
\end{align*}
\]

where \( i \in \{0, 1\} \). The second equation holds for \( x > 0 \) when \( i = 0 \) and functions \( h_0 \) and \( h_1 \) take values in \( \{0, 1\} \). Some authors [12] avoid the bound on the values of \( h_0 \) and \( h_1 \) by replacing the second equation above with

\[
f(s_i(x), y) = s_{rem_2(h_i(x,y))}(f(x,y)).
\]

Obviously, CIN is a simpler variant of the CRN operator.

The following statement defines the CRN operator by means of the summation operator. The easy proof is left to the reader.

**Lemma 1 (Fundamental Lemma).** For any \( g, h_0, h_1 \),

\[
\begin{align*}
  f(0, y) &= g(y), \\
  f(s_i(x), y) &= s_{h_i(x,y)}(f(x,y))
\end{align*}
\]

iff

\[
f(x, y) = g(y)2^{|x|} + \left( \sum_{i < |x|} h_{\text{bit}_i(x,y)}(\lfloor x/2^{i+1} \rfloor, y) \cdot 2^i \right) \Box
\]

**Corollary 2.** For any \( h_0, h_1 \),

\[
\begin{align*}
  f(0, y) &= 0, \\
  f(s_i(x), y) &= s_{h_i(x,y)}(f(x,y))
\end{align*}
\]

iff

\[
f(x, y) = \sum_{i < |x|} h_{\text{bit}_i(x,y)}(\lfloor x/2^{i+1} \rfloor, y) \cdot 2^i \Box
\]

For any set \( F \) of functions and any set \( \Omega \) of operators on functions, the *inductive closure of \( F \) with respect to \( \Omega \), \( \text{clos}(F, \Omega) \), is the least set of functions containing \( F \) and closed with respect to the operators of \( \Omega \).

For any set \( \{f_1, \ldots, f_a\} \) of functions, let

\[
\text{clos}(f_1, \ldots, f_a; op_1, \ldots, op_b) = \text{clos}(\{f_1, \ldots, f_a\} \cup \{op_1, \ldots, op_b\})
\]

and

\[
\text{clos}(I, f_1, \ldots, f_a; op_1, \ldots, op_b) = \text{clos}(I \cup \{f_1, \ldots, f_a\} \cup \{op_1, \ldots, op_b\})
\]

where \( I \) is the set of the projection functions

\[
I^a[i] : \ x_1, \ldots, x_a \mapsto x_i \quad (1 \leq i \leq a)
\]

with any arity \( a \).
2 Pairings

This section recalls the basic notions of pairings and the method of simulation (relative to a pairing) [6] for characterizing the unary functions of a given function class.

A pairing is an injective binary function $J$, equipped with two unary functions $K, L$ such that

$$K(J(x, y)) = x, \quad L(J(x, y)) = y$$

for every $x$ and $y$.

Pairings can be used to define codings $J^a : N^a \rightarrow N$ of sequences of natural numbers:

$$J^1 x = x, \quad J^{a+1}(x, x_1, \ldots, x_a) = J(x, J^a(x_1, \ldots, x_a)) .$$

So, for $J^a[i] = L^{-1} \circ K$, where $1 \leq i < a$ and $J^a[a] = L^{-1}$, we have

$$J^a[i](J^a(x_1, \ldots, x_a)) = x_i$$

for every $x_1, \ldots, x_a, a$ and $1 \leq i \leq a$. A function $f : N^a \rightarrow N$ is simulated by a unary function $f'$ relatively to a pairing $J$,

$$f \square_J f' ,$$

iff, for every $x_1, \ldots, x_a$,

$$f(x_1, \ldots, x_a) = f'(J^a(x_1, \ldots, x_a)) ,$$

see [6].

A function set $F$ is simulated by a unary function set $G$ relatively to a pairing $J$,

$$F \square_J G ,$$

iff

$$\forall f \in F \exists f' \in G \ f \square_J f' .$$

Lemma 3 (Simulation Lemma). For any pairing $J : F \square_J G \Rightarrow F_1 \subseteq G .$

We conclude the section by introducing the pairing

$$J(x, y) = 2^{\lvert y \rvert} \ast y \ast x$$

which will be used from now on.

3 Basic results

We show now a series of technical lemmata concerning a function class $F$. We assume that $F$ is a class closed with respect to substitution and CIN which contains the function set $\{ s_0, s_1, \text{rem}_2, \text{len}, \text{add} \} \cup I$.

Let $\text{ones}(x) = 2^{\lvert x \rvert} - 1$ and let $\text{dl}(x) = x \cdot 2^{\lvert x \rvert - 1}$. The latter function returns the argument $x$ with the most significant bit cleared.
Lemma 4. \( C_0, C_1, S, \text{ones}, sg, \text{cosg}, \text{dl}, \text{div} \in \mathcal{F} \).

Proof. Let
\[
\begin{align*}
C_0(x) &= \text{rem}_2(s_0(x)), \quad C_1(x) = \text{rem}_2(s_1(x)), \quad S(x) = x + C_1(x),
\end{align*}
\]
and note that
\[
\begin{align*}
\text{ones}(0) &= 0, \\
\text{ones}(s_1(x)) &= s_1(\text{ones}(x)), \\
\end{align*}
\]
i.e. \( \text{ones} = \text{CIN}(C_1, C_1) \). Then, we have
\[
\begin{align*}
sg(x) &= \text{rem}_2(\text{ones}(x)), \\
\text{cosg}(x) &= \text{rem}_2(\text{ones}(x) + 1)
\end{align*}
\]
and
\[
\begin{align*}
dl(0) &= 0, \\
dl(s_1(x)) &= \begin{cases} 
  s_0(dl(x)) & \text{if } x = 0 \\
  s_i(dl(x)) & \text{otherwise}
\end{cases}
\]
i.e. \( dl = \text{CIN}(C_0, sg) \).

Finally, it is easy to show by induction that \( \text{div}_2 \) can be defined by \( \text{CIN} \) as follows
\[
\begin{align*}
\text{div}_2(0) &= 0, \\
\text{div}_2(s_1(x)) &= s_{\text{rem}_2(x)}(\text{div}_2(x)).
\end{align*}
\]

Lemma 5. The conditional function belongs to \( \mathcal{F} \).

Proof. Consider the function
\[
\begin{align*}
2^{\text{ones}(y) + \text{sg}(x)} + y
\end{align*}
\]
such that
\[
\begin{align*}
2^{\text{ones}(y) + \text{sg}(x)} + y &= \begin{cases} 
  1 \ast y & \text{if } x = 0 \\
  2 \ast y & \text{otherwise}
\end{cases}
\]
and
\[
\begin{align*}
|2^{\text{ones}(y) + \text{sg}(x)}| + y &= \begin{cases} 
  |y| + 1 & \text{if } x = 0 \\
  |y| + 2 & \text{otherwise}
\end{cases}.
\end{align*}
\]
Set
\[
\begin{align*}
h_0(z) = 0, \quad h_1(z) = \text{cosg}(\text{rem}_2(|z| + |dl(z)|))
\end{align*}
\]
and note that
\[
\begin{align*}
h_1(0) &= 1, \quad h_1(1 \ast y) = 0, \quad h_1(2 \ast y) = 1.
\end{align*}
\]
Then, for 
\[ \text{case}(0) = 0, \]
\[ \text{case}(s_i(x)) = s_{h_i(x)}(\text{case}(x)), \]
it is easy to prove by induction that 
\[ \text{case}(1 * y) = 2^{|y|} \]
\[ \text{case}(2 * y) = 2^{|y|+1} + y \]
so that 
\[ \text{dl}(\text{case}(1 * y)) = 0 \]
\[ \text{dl}(\text{case}(2 * y)) = y \]
and the lemma follows immediately from Lemma 4 in so far as 
\[ \text{cond}(x, y, z) = \text{dl}(\text{case}(2^{|\text{ones}(z)|+sg(x)} + z)) + \text{dl}(\text{case}(2^{|\text{ones}(y)+\cosg(x)}| + y)). \]

Let 
\[ \text{neg}(0) = 0, \]
\[ \text{neg}(s_i(x)) = s_{\cosg(i)}(\text{neg}(x)), \]
i.e. \(\text{neg} = CIN(C_1, C_0)\), and set 
\[ \text{compl}_2(x, y) = \text{neg}(2^{|x|} + y) + 1. \]

It is easy to show by induction that \(\text{neg}(x)\) is the number whose binary representation is the one's complement of \(x\) for any \(x > 0\) and that \(\text{compl}_2(x, y)\) is the two’s complement of \(y\) with \(|x|\) bits for any \(x\) and \(y\) with \(0 < y < 2^{|x|} - 1\) (note that \(\text{compl}_2(x, 0) = 2^{|x|}\) has \(|x| + 1\) bits).

Let 
\[ \text{diff}(x, y, z) = x + \text{compl}_2(z, y) \]
and set 
\[ \text{diff}(x, y) = x + \text{compl}_2(2x, y). \]

**Lemma 6.** For any \(x, y, z\), if \(x, y \leq z\) and \(y \neq 0\) then 
\[ x \geq y \implies |\text{diff}(x, y, 2z)| = |\text{compl}_2(2z, y)| + 1, \]
\[ x < y \implies |\text{diff}(x, y, 2z)| = |\text{compl}_2(2z, y)|. \]

**Proof.** Let \(u = \text{compl}_2(2z, y)\). We avoid the case \(y = 0\) because \(u = \text{compl}_2(2z, 0) = 2^{|z|+1}\) does not satisfy the hypothesis that \(|u| = |z| + 1\).

By hypothesis, the binary representation of \(x\) coincides with its two’s complement representation with \(|z| + 1\) bits and \(u\) is the two’s complement representation with \(|z| + 1\) bits of \(-y\).

Let \(v = \text{diff}(x, y, 2z) = x + u\). Since the absolute value of \(x - y\) is bounded by \(z\) and \(\text{bit}(x, |z|) = 0\), \(\text{bit}(y, |z|) = 1\), we have the following two cases:
1. If \( x \geq y \) then the bit sign \( \text{bit}(v,|z|) \) must be zero. This implies a carry from the \(|z| - 1\) position which propagates to the \(|z| + 1\) position, i.e. \( \text{bit}(v,|z| + 1) = 1 \). Therefore, \(|v| = |u| + 1\).

2. If \( x < y \) then the bit sign \( \text{bit}(v,|z|) \) must be one. This implies no carry from the \(|z| - 1\) position and \( \text{bit}(v,|z| + 1) = 0 \). Therefore, \(|v| = |u|\).

Let

\[
\text{cmp}(x, y, z) = \text{rem}_2(|\text{diff}(x, y, 2z)| + |\text{compl}_2(2z, y)|)
\]

and

\[
\text{sub}(x, y, z) = \text{dl}(x + \text{compl}_2(2z, y)), \quad \text{sub}(x, y) = \text{dl}(x + \text{compl}_2(2x, y)).
\]

The following statements are immediate consequences of the lemma above.

**Corollary 7.** For any \( x, y, z \), if \( x, y \leq z \) and \( y \neq 0 \) then

\[
x < y \iff \text{cmp}(x, y, z) = 0
\]

**Lemma 8.** For any \( x, y, z \), if \( y \leq x \) and \( y \neq 0 \) then

\[
x - y = \text{sub}(x, y)
\]

and if \( y \leq x \leq z \) and \( y \neq 0 \) then

\[
x - y = \text{sub}(x, y, 2z)
\]

**Lemma 9.** The modified subtraction \( \hat{x} - y \) belongs to \( \mathcal{F} \).

**Proof.** The lemma follows from the corollary and the lemmata above in so far as

\[
\hat{x} - y = \left\{ \begin{array}{ll}
x & \text{if } y = 0 \\
\text{sub}(x, y, 2(x + y)) & \text{if } y > 0 \land x \geq y \\
0 & \text{if } y > 0 \land x < y
\end{array} \right.
\]

From Corollary 7, Lemma 9 and Lemma 4 we obtain immediately the following result by noting that \( x = y \iff (\hat{x} - y) + (y - \hat{x}) = 0 \).

**Lemma 10.** \( \mathcal{F} \) contains the characteristic functions of the predicates definable by means of the comparison predicates \( x < y, x \leq y, x > y, x \geq y, x = y, x \neq y \) and the boolean operations. □
4 Replacing CRN with CIN

We show now that any class \( \mathcal{F} \) satisfying the closure properties stated in the previous section is closed with respect to CRN provided that \( \mathcal{F} \) contains also the function \( mp : x \mapsto 2^{\|x\|}x \).

**Lemma 11 (Basic Lemma).** There are functions \( ml, cn \in \mathcal{F} \) such that

\[
ml(x, y) = x 2^{\|y\|}, \quad cn(x, y) = x * y
\]

for any \( x, y \) with \( |y| \geq |x| \).

**Proof.** Assume that \( |y| \geq |x| \). Then, define

\[
ml(x, y) = dl(div_2(mp(2^{\|y\|} + x)))
\]

and note that

\[
ml(x, y) = dl(div_2(2^{\|y\|+1}(2^{\|y\|} + x))) = dl(2^{2\|y\|} + 2^{\|y\|}x) = 2^{\|y\|}x,
\]

\[
cn(x, y) = ml(x, y) + y.
\]

We show now that the decodings \( K, L \) of pairing \( J \) belong to \( \mathcal{F} \).

For \( z = 2^{\|v\|} * v \), note that

\[
|z| = |y| + |v| + 1,
\]

\[
dl(z) = v,
\]

\[
|dl(z)| = |v|.
\]

Let

\[
isCode(z) = \begin{cases} 
1 & \text{if } \exists x, y J(x, y) = z \\
0 & \text{otherwise}
\end{cases}
\]

be the characteristic function of the image of pairing \( J \).

**Lemma 12.** \( isCode \in \mathcal{F} \).

**Proof.** It is easy to see that

\[
isCode(z) = 1 \iff \exists x, y (z = 2^{\|y\|} * y * x)
\]

\[
\iff \exists_{n,v} (z = 2^n * v) \land (|v| \geq n)
\]

\[
\iff (z > 0) \land (|dl(z)| \geq |z| - (|dl(z)| + 1))
\]

and the lemma follows immediately by Lemma 9 and Lemma 10.

**Lemma 13.** \( K \in \mathcal{F} \).
Proof. For $i \in \{0, 1\}$, consider the functions

$$h_i(z) = \begin{cases} i & \text{if Code}(z) \\ 0 & \text{otherwise} \end{cases}$$

such that

$$h_i(z) = \begin{cases} i & \exists y,u(z = 2^{|y|} \ast y \ast u) \\ 0 & \text{otherwise} \end{cases}$$

Then, for $K(0) = 0$, $K(s_i(z)) = s_{h_i(z)}(K(z))$

we have

$$K(2^{|y|} \ast y \ast x) = x$$

for any $x, y$.

The proof of the statement above can be carried out by induction on $x$ and is left to the reader.

The proof that $L \in \mathcal{F}$ is not as easy as that of the lemma above. We need first to define some auxiliary functions. For any unary function $f$, let

$$\tilde{f}(x) = 2^{|f(x)|} \ast x.$$

**Lemma 14.** For any unary function $f$ such that $f(x) \leq x$ for any $x$, we have $\tilde{f} \in \mathcal{F}$.

**Proof.** The lemma follows immediately by the Basic Lemma and by noting that

$$\tilde{f}(x) = 2^{|f(x)|} \ast x = \text{div}_2(\text{cn}(2^{|f(x)|}, s_1(x)))$$

in so far as, for any $x$,

$$|2^{|f(x)|}| = |f(x)| + 1 \leq |x| + 1 = |s_1(x)|$$

**Lemma 15.** There is a function $\text{get} \in \mathcal{F}$ such that

$$\text{get} : 2^n \ast 2^m \ast v \mapsto \text{bit}(v,n)$$

for $|v| > n$.

**Proof.** Let $z = 2^{|v|} \ast 2^n \ast 2^m \ast v$ where $|v| > n$ and set $z^{(j)} = \left\lfloor z/2^j \right\rfloor$. Then, for $j \leq |v|$,

$$z^{(j)} = 2^{|v|} \ast 2^n \ast 2^m \ast \left\lfloor v/2^j \right\rfloor,$$

$$|z^{(j)}| = 2^{|v|} + m + n + 3 - j,$$

$$|\text{dl}(z^{(j)})| = |v| + m + n + 2 - j.$$
\[ |dl^2(z^{(j)})| = |v| + m + 1 - j, \]
\[ |dl^3(z^{(j)})| = |v| - j, \]
\[ |z^{(j)}| - |dl(z^{(j)})| = |v| + 1, \]
\[ |dl^2(z^{(j)})| - |dl^3(z^{(j)})| = m + 1 \]

and, in particular,
\[ z^{(n)} = 2^{|v|} \times 2^n \times 2^m \times \lfloor v/2^n \rfloor, \]
\[ |z^{(n)}| = 2^{|v|} + m + 3. \]

Therefore, for \( x \in \{ z^{(j)} \}, \)
\[ x = z^{(n)} \iff \exists j < |v| : x = z^{(j)} \land |x| = 2^{|v|} + m + 3 \]
\[ \iff dl^3(x) > 0 \land |x| = 2(|x| - |dl(x)|) + (|dl^2(x)| - |dl^3(x)|). \]

Let \( c(x) \) be the characteristic function of predicate
\[ (dl^3(x) > 0) \land |x| = 2(|x| - |dl(x)|) + (|dl^2(x)| - |dl^3(x)|) \]

and consider the function
\[ h_i(z) = \begin{cases} 0 & \text{if } c(s_i(z)) = 0 \\ i & \text{otherwise} \end{cases} \]
such that
\[ h_{bit(v,j)}(z^{(j+1)}) = \begin{cases} bit(v,n) & \text{if } j = n \\ 0 & \text{otherwise}. \end{cases} \]

Then, for
\[ h'(0) = 0, \]
\[ h'(s_i(z)) = s_{h_i(z)}(h'(z)), \]
by the Fundamental Lemma we have
\[ h'(2^{|v|} \times 2^n \times 2^m \times v) = bit(v,n) \cdot 2^n \]
and for \( get(x) = sg(h'(dl^2(x))) \), we have
\[ get(2^n \times 2^m \times v) = sg(h'(dl^2(2^n \times 2^m \times v))) = sg(h'(2^{|v|} \times 2 \times 2^m \times v)) = bit(v,n) \]

where the function \( \tilde{dl}^2(x) = 2^{|dl^2(x)|} \times x \) belongs to \( \mathcal{F} \) by Lemma 14. \( \Box \)

**Lemma 16.** \( L \in \mathcal{F}. \)
Proof. Let \( z = 2^{|u|} \cdot J(u, y) = 2^{|u|} \cdot 2^{|y|} \cdot y \cdot u \), set \( z^{(j)} = \lfloor z/2^j \rfloor \) and note that, for \( j \leq |y| + |u|, \)

\[
\begin{align*}
  z^{(j)} &= 2^{|u|} \cdot 2^{|y|} \cdot \lfloor y \cdot u/2^j \rfloor, \\
  dl(z^{(j)}) &= 2^{|y|} \cdot \lfloor y \cdot u/2^j \rfloor, \\
  dl^2(z^{(j)}) &= \lfloor y \cdot u/2^j \rfloor, \\
  |z^{(j)}| &= 2|y| + 2|u| + 2 - j, \\
  |dl(z^{(j)})| &= 2|y| + |u| + 1 - j, \\
  |dl^2(z^{(j)})| &= |y| + |u| - j, \\
  |z^{(j)}| \cdot |dl(z^{(j)})| &= |y| + 1, \\
  |dl(z^{(j)})| \cdot |dl^2(z^{(j)})| &= |y| + 1.
\end{align*}
\]

Therefore, for \( x \in \{z^{(j)}\}, \)

\[
x \in \{z^{(j)}|j < |y|\} \iff \exists j < |y| \land x = z^{(j)} \land |x| > |y| + 2|u| + 2 \iff \exists j < |y| \land \big| dl^2(x) > 0 \land |x| > 2(|x| - |dl(x)|) + (|dl(x)| - |dl^2(x)|) \cdot 1.
\]

Let \( c(x) \) be the characteristic function of predicate

\[
\big( dl^2(x) > 0 \land |x| > 2(|x| - |dl(x)|) + (|dl(x)| - |dl^2(x)|) \cdot 1. \]

Now, consider the function

\[
h_i(z) = \begin{cases} 
  \text{get}(s_i(z)) & \text{if } c(s_i(z)) = 1 \\
  0 & \text{otherwise}
\end{cases}
\]

and, by the lemma above, we have

\[
h_{bit(v, j)}(z^{(j+1)}) = \begin{cases} 
  \text{get}(z^{(j)}) & \text{if } j < |y| \\
  0 & \text{otherwise}
\end{cases} = \begin{cases} 
  \text{bit}(y, j) & \text{if } j < |y| \\
  0 & \text{otherwise}
\end{cases}
\]

where \( v = y \cdot u \). Then, for

\[
\begin{align*}
  L'(0) &= 0, \\
  L'(s_i(z)) &= s_{h(z)}(L'(z)),
\end{align*}
\]

by the Fundamental Lemma we have

\[
L'(2^{|u|} \cdot 2^{|y|} \cdot y \cdot u) = y
\]

for any \( u, y \). Finally, we have \( L(x) = L'(\tilde{K}(x)) \) in so far as

\[
L'(\tilde{K}(J(u, y))) = L'(\tilde{K}(2^{|y|} \cdot y \cdot u)) = L'(2^{|u|} \cdot 2^{|y|} \cdot y \cdot u) = y
\]

and the Lemma follows immediately by the Basic Lemma. \( \square \)
We are now able to state a series of closure properties for $\mathcal{F}$, which culminate in the closure of $\mathcal{F}$ with respect to CRN.

**Lemma 17.** Let

$$f(0, y) = 0,$$

$$f(s_i(x), y) = s_{h_i(x,y)}(f(x, y))$$

where $h_0$ and $h_1$ take values in $\{0, 1\}$. Then,

$$f(x, y) = f'(J(x, y))$$

where

$$f'(0) = 0,$$

$$f'(s_i(z)) = s_{h'_i(z)}(f'(z))$$

and

$$h'_i(z) = \begin{cases} h_i(K(z), L(z)) & \text{if } isCode(z) \\ 0 & \text{otherwise} \end{cases}.$$ 

**Proof.** Note first that

$$h'_i(z) = \begin{cases} h_i(u, y) & \text{if } \exists y, u (z = J(u, y)) \\ 0 & \text{otherwise} \end{cases}$$

and let $z^{(j)} = \lfloor (2^{|y|} * y * x) / 2^j \rfloor$. Then, for any $j \leq |x|,$

$$z^{(j)} = 2^{|y|} * y * \lfloor x / 2^j \rfloor,$$

$$h'_i(z^{(j)}) = h_i(\lfloor x / 2^j \rfloor, y)$$

and $h_i(z^{(j)}) = 0$ for $|x| < j$. Then, by the corollary of the Fundamental Lemma we obtain

$$f(x, y) = f'(J(x, y)).$$

**Lemma 18.** Let

$$f(0, y) = 0,$$

$$f(s_i(x), y) = s_{h_i(x,y)}(f(x, y))$$

where $h_0$ and $h_1$ take values in $\{0, 1\}$. If $h_0, h_1 \in \mathcal{F}$ then there is a function $g \in \mathcal{F}$ such that $g(x, y) = f(x, y)$ for any $x, y$ with $|x| \geq |y|$. 

\[\square\]
Proof. By the Lemma above,

\[ f(x, y) = f'(J(x, y)) \]

where

\[ f'(0) = 0, \]

\[ f'(s_i(z)) = s_{h'(i)}(f'(z)) \]

and the functions \( h'_i \) are defined by means of \( h_i, K, L \) and other functions in \( F \). Then \( f' \in F \). On the other hand, by the Basic Lemma, for \(|x| \geq |y|\),

\[ J(x, y) = 2^{|y|} \cdot y \cdot x = \text{div}_2(cn(2^{|y|}, s_1(y \cdot x))) = \text{div}_2(cn(2^{|y|}, s_1(cn(y, x)))) \]

and the theorem follows immediately by noting that

\[ g(x, y) = f'(\text{div}_2(cn(2^{|y|}, s_1(cn(x, y))))) = f'(J(x, y)) = f(x, y). \]

We show now that some well known functions are in \( F \).

**Lemma 19.** \( \text{bit} \in F \).

**Proof.** Consider the function

\[ B(0, z) = 0 \]

\[ B(s_i(x), z) = \begin{cases} 
 s_i(B(x, z)) & \text{if } |x| + 1 = z \\
 s_0(B(x, z)) & \text{otherwise}
\end{cases} \]

and note that, for \( y < |x| \),

\[ B(x, |x| - y) = \text{bit}(x, y) \cdot 2^y. \]

Then,

\[ \text{bit}(x, y) = \begin{cases} 
 \text{sg}(B(x, |x| - y)) & \text{if } y < |x| \\
 0 & \text{otherwise}
\end{cases} \]

and the lemma follows immediately from Lemma 18 by noting that \(|x| \geq ||x| - y|\).

**Lemma 20.** \( \text{MSP} \in F \).

**Proof.** Let

\[ M(0, y) = 0 \]

\[ M(s_i(x), y) = s_{\text{bit}(s_i(x), y)}(M(x, y)) \].
Then the lemma follows immediately from the lemma above and Lemma 18 in so far as
\[
MSP(x, y) = \begin{cases} 
M(x, y) & \text{if } y \leq |x| \\
0 & \text{otherwise}
\end{cases}.
\]

Lemma 21. The function \( x, y \mapsto x \cdot 2^{|y|} \) belongs to \( \mathcal{F} \).

Proof. Recall first that, by Lemma 5 and Lemma 10, function \( \max \) belongs to \( \mathcal{F} \). Then, for \( f(x, y) = ml(x, \max(x, y)) = x \cdot 2^{\max(|x, y|)} \), we have
\[
x \cdot 2^{|y|} = MSP(f(x, y), \max(x, y)|-|y|).
\]

Corollary 22. \( \text{conc}, J \in \mathcal{F} \).

The following corollary follows from Lemma 18.

Corollary 23. Let
\[
f(0, y) = 0,
f(s_i(x), y) = s_{h_i(x, y)}(f(x, y))
\]
where \( h_0 \) and \( h_1 \) take values in \( \{0, 1\} \). If \( h_0, h_1 \in \mathcal{F} \) then \( f \in \mathcal{F} \).

Lemma 24. Let
\[
f(0, y) = g(y),
f(s_i(x), y) = s_{h_i(x, y)}(f(x, y))
\]
and let
\[
f'(0, y) = 0,
f'(s_i(x), y) = s_{h_i(x, y)}(f'(x, y))
\]
where \( h_0 \) and \( h_1 \) take values in \( \{0, 1\} \). Then,
\[
f(x, y) = g(y)2^{|x|} + f'(x, y).
\]

Proof. By induction on \( x \).

Theorem 25. \( \mathcal{F} \) is closed with respect to CRN.

Proof. The theorem follows immediately by the Lemma above, Lemma 21, and Corollary 23. The case of two or more parameters \( y_1, \ldots, y_a \) can be reduced to the one parameter case by encoding the sequence \( y_1, \ldots, y_a \) into the number \( y = J^a(y_1, \ldots, y_a) \).
Lemma 26. \( LSP, msp, lsp, FH, BH \in F \).

Proof. Let

\[
LSP'(0, y, z) = 0
\]

\[
LSP'(s_i(x), y, z) = \begin{cases} 
    s_i(LSP'(x, y, z)) & \text{if } |x| + 1 + y \geq |z| \\
    s_0(LSP'(x, y, z)) & \text{otherwise}
\end{cases}
\]

Then,

\[
LSP(x, y) = LSP'(x, y, x),
\]

\[
msp(x, y) = MSP(x, |y|),
\]

\[
lsp(x, y) = LSP(x, |y|),
\]

\[
BH(x) = LSP(x, \lceil |x|/2 \rceil) = LSP(x, x - \lfloor |x|/2 \rfloor),
\]

\[
FH(x) = MSP(x, \lfloor |x|/2 \rfloor) = msp(x, BH(x)).
\]

\[\square\]

Lemma 27. If \( u \in F \) then the smash function belongs to \( F \).

Proof. Note first that

\[
2(|x| + |y|)^2 = 2|x+y|^2,
\]

\[
|x|^2 + |y|^2 = \text{div}_2(2|x|^2 \ast 2|y|^2).
\]

Then

\[
f(x, y) = MSP(2(|x| + |y|)^2, |x|^2 + |y|^2) = \left( \frac{2(|x| + |y|)^2}{2|x|^2 + |y|^2} \right)
\]

and

\[
x \# y = 2|x| \cdot |y| = FH(2^2|x| \cdot |y|) = FH(f(x, y)).
\]

\[\square\]

5 Characterizations of complexity classes based on CIN

We now apply the results of the two sections above to show that the class \( AC^0 \)
of functions computable by polysize, constant depth boolean circuits, the class \( TC^0 \)
of functions computable by polysize, constant depth threshold circuits and the class \( NC^1 \)
of functions computable by polysize, logarithmic depth boolean circuits can be characterized using CIN instead of CRN. In the next section, using such characterizations and the simulation lemma, we will characterize the set of unary functions of such classes.

Here and in the following sections, we will refer to \( \text{clos}(f_1, \ldots, f_a; op_1, \ldots, op_b) \)
as \( F_n \) in the proof of theorems of the form

\[\text{Theorem n. } < \text{function class} > = \text{clos}(f_1, \ldots, f_a; op_1, \ldots, op_b).\]
Theorem 28. Classes $AC^0$, $TC^0$ and $NC^1$ can be defined as follows:

1. $AC^0 = \text{clos}(C_0, I, s_0, s_1, \#, \text{len}, \text{bit}; \text{SUBST}, \text{CRN})$,
2. $TC^0 = \text{clos}(C_0, I, s_0, s_1, \#, \text{len}, \text{bit}, \text{mult}; \text{SUBST}, \text{CRN})$,
3. $NC^1 = \text{clos}(C_0, I, s_0, s_1, \#, \text{len}, \text{bit}, \text{tree}; \text{SUBST}, \text{CRN})$.

Proof. Statements (1,3) have been proved in [2], whereas statement (2) has been proved in [1].

Function $\text{tree}$ occurring in statement (3) is a unary function taking values in $\{0, 1\}$. We do not define $\text{tree}$ here because its definition is rather involved and not relevant to our aims. The interested reader may find a detailed description of $\text{tree}$ at page 64 of [2].

Theorem 29.

$$AC^0 = \text{clos}(I, s_0, s_1, \text{rem}_2, \text{us}, \text{len}, \text{mp}, \text{add}; \text{SUBST}, \text{CIN})$$

Proof. From statement (1) of the theorem above, Theorem 25, Lemma 19 and Lemma 27, we have

$$AC^0 \subseteq \mathcal{F}_{29}.$$ 

To prove that

$$\mathcal{F}_{29} \subseteq AC^0,$$

by statement (1) it suffices to show that

$$\text{rem}_2, \text{us}, \text{add}, \text{mp} \in \text{clos}(C_0, I, s_0, s_1, \#, \text{len}, \text{bit}; \text{SUBST}, \text{CRN}).$$

To this aim, the reader may see Section 3.1 of [3].

Theorem 30.

$$TC^0 = \text{clos}(I, s_0, s_1, \text{rem}_2, \text{us}, \text{len}, \text{quad}, \text{add}; \text{SUBST}, \text{CIN}).$$

Proof. By Lemma 4 and Lemma 9, we have that $\text{mult}, \text{mp} \in \mathcal{F}_{30}$ in so far as

$$xy = \text{div}_2(((x+y)^2-x^2)-y^2),$$

$$mp(x) = x \cdot 2^{|x|}.$$ 

Then, from statement (2) of Theorem 28, Theorem 25, Lemma 19 and Lemma 27, we have

$$TC^0 \subseteq \mathcal{F}_{30}.$$ 

To prove that

$$\mathcal{F}_{30} \subseteq TC^0,$$

by statement (2) of Theorem 28, it suffices to show that

$$\text{rem}_2, \text{us}, \text{add}, \text{quad} \in \text{clos}(C_0, I, s_0, s_1, \#, \text{len}, \text{bit}, \text{mult}; \text{SUBST}, \text{CRN}).$$
The following theorem can be proved in the same way of the two theorems above.

**Theorem 31.**

\[ \text{NC}^1 = \text{clos}(I, s_0, s_1, \text{rem}_2, \text{mp}, \text{us}, \text{len}, \text{tree}, \text{add}; \text{SUBST}, \text{CIN}). \]

### 6 Characterizations of the unary functions of complexity classes based on CRN

Now, we apply the simulation lemma to the new characterizations of \( \text{AC}^0, \text{TC}^0 \) and \( \text{NC}^1 \) for obtaining new characterizations of the unary functions of \( \text{AC}^0, \text{TC}^0 \) and \( \text{NC}^1 \).

Let \( \mathcal{F} \) be a class of unary functions closed with respect to function addition, composition and CIN which contains the function set \( \{s_0, s_1, \text{rem}_2, \text{len}, \text{mp}\} \).

**Lemma 32.** ones, pow, sg, cosg, div, \( C_0, C_1, S \in \mathcal{F} \).

**Proof.** The proof is the same of Lemma 4. \( \square \)

Let \( \otimes \) be any function (predicate) of arity \( a \). We denote with \( \otimes(f_1, \ldots, f_a) \) the operator mapping unary functions \( f_1, \ldots, f_a \) into the function (predicate)

\[ f : x \mapsto \otimes(f_1(x), \ldots, f_a(x)) \]

and when \( \otimes \) is binary, we use infix notation:

\[ f \otimes g : x \mapsto (f(x)) \otimes (g(x)). \]

Applying the above transformation to a function definition corresponds to replace variables with functions, e.g. \( x, y, z \) are replaced by \( f(u), g(u), h(u) \) where \( u \) is a new variable. Therefore, if we apply the substitution to all of the function definitions in Section 3 and Section 4 we derive functional operators satisfying the following two statements.

**Theorem 33.** \( \mathcal{F} \) is closed with respect to the operators

\[ f \rightarrow g, \text{cond}(f, g, h), f < g, f \leq g, f > g, f \geq g, f = g, f \neq g, f \wedge g, f \vee g, \neg f. \]

**Theorem 34.** Functions \( K, L \) belong to \( \mathcal{F} \) and \( \mathcal{F} \) is closed with respect to the operators \( f * g, (f, g)_J \).

We are now ready to introduce the characterizations of \( \text{AC}^0_1, \text{TC}^0_1 \) and \( \text{NC}^1_1 \).

**Theorem 35.**

\[ \text{AC}^0_1 = \text{clos}(s_0, s_1, \text{rem}_2, \text{us}, \text{len}, \text{mp}; f + g, f \circ g, \text{CIN}). \]

**Proof.** By the Simulation Lemma, to show that

\[ \text{AC}^0_1 \subseteq \mathcal{F}_{35}, \]

we need to prove that

\[ \text{AC}^0_1 \sqcap_J \mathcal{F}_{35} \]
and, by Theorem 29, this is equivalent to
\[ \text{clos}(I, s_0, s_1, \text{rem}_2, \text{us}, \text{len}, \text{mp}, \text{add}; \text{SUBST}, \text{CIN}) \mathbin{\square} F_{35}. \]

We show the statement above by induction on
\[ \text{clos}(I, s_0, s_1, \text{rem}_2, \text{us}, \text{len}, \text{mp}, \text{add}; \text{SUBST}, \text{CIN}). \]

Induction Basis. Obviously, \( f \mathbin{\square} f \) for \( f \in \{\text{id}, s_0, s_1, \text{rem}_2, \text{us}, \text{len}, \text{mp}\} \) where \( \text{id} = s_0 \circ \text{div}_2 \). Moreover,
\[ \text{add} \mathbin{\square} (K + L), \]
\[ I^{a}[i] \mathbin{\square} J^{a}[i] \]
where \( J^{a}[i] = L^{i-1} \circ K \), for \( 1 \leq i < a \) and \( J^{a}[a] = L^{a-1} \) belong to \( F_{35} \) by the theorem above.

Induction Step. It suffices to note that
\[ \text{CIN}(h_0, h_1) \mathbin{\square} \text{CIN}(h_0, h_1), \]
\[ f(g_1, \ldots, g_b) \mathbin{\square} f'(g'_1, \ldots, g'_b) \]
where \( f \mathbin{\square} f' \), \( g_1 \mathbin{\square} g'_1, \ldots, g_b \mathbin{\square} g'_b \) and, for \( a \geq 2 \), \( J^{a} \) is the coding operator obtained by composing \( a - 1 \) times the pairing operator:
\[ J^2(f, g) = (f, g)_f, J^{a+2}(f_1, \ldots, f_{a+2}) = (f_1, J^{a+1}(f_2, \ldots, f_{a+2})), \]
The converse inclusion,
\[ F_{35} \subseteq AC_{1}^{0}, \]
is straightforward and the theorem follows immediately.

**Theorem 36.**
\[ TC_{1}^{1} = \text{clos}(s_0, s_1, \text{rem}_2, \text{sqr}, \text{us}, \text{len}; f + g, f \circ g, \text{CIN}). \]

**Proof.** The proof can be carried out like that of the theorem above, by using Theorem 30 and by recalling that \( F_{36} \) is closed with respect to the pairing and the concatenation operator in so far as the multiplication operator can be defined as
\[ f \cdot g = (((((f + g) \circ \text{quad}) \circ \text{quad}) \circ \text{quad}) \circ \text{quad}) \circ \text{div}_2, \]
and therefore
\[ \text{mp} = \text{id} \cdot \text{pow}. \]

**Theorem 37.**
\[ NC_{1}^{1} = \text{clos}(s_0, s_1, \text{rem}_2, \text{mp}, \text{us}, \text{len}, \text{tree}; f + g, f \circ g, \text{CIN}). \]

**Proof.** By using Theorem 31, the proof can be carried out like that of Theorem 35.
7 Characterization of unary \( NC \) functions

Many complexity classes are inductively defined using restricted forms of primitive recursion (on notation) besides substitution and concatenation recursion on notation. This allows us to apply the results of the previous sections in order to inductively define the set of unary functions of such classes. We do this for the complexity class \( NC \).

Again, we have to simulate functions with unary functions; the crucial step here is to simulate primitive recursion on notation.

Let \( WBIN \) be the weak bounded iteration operator transforming unary functions \( f, g, h \) into the function

\[
WBIN(f, g, h) : n \mapsto g^{\|f_n\|}n
\]

provided that \( g^i n \leq h n \) for any \( i \leq \|f_n\| \). Note that \( \|x\| = \text{len}^2(x) \).

Let \( WBRN \) be the weak bounded recursion on notation operator transforming functions \( g, h_0, h_1, k \) into the function \( f \) such that

\[
\begin{align*}
f(x, y) &= F(|x|, y), \\
F(0, y) &= g(y), \\
F(s_0(x), y) &= h_0(x, y, F(x, y)), \text{ if } x \neq 0 \\
F(s_1(x), y) &= h_1(x, y, F(x, y))
\end{align*}
\]

provided that \( F(x, y) \leq k(x, y) \) for every \( x, y \). The following statement has been proved in [2].

Lemma 38.

\[
NC = \text{clos}(C_0, I, s_0, s_1, \#, \text{len}, \text{bit}; \text{SUBST}, \text{CRN}, \text{WBRN}).
\]

First, we prove the following technical lemma.

Lemma 39.

\[
\text{clos}(I, s_0, s_1, \text{rem}_2, \text{us}, \text{len}, \text{mp}, \text{add}; \text{SUBST}, \text{CIN}, \text{WBIN})
\]

is closed with respect to weak bounded recursion on notation.

Proof. We show the proof for a single parameter \( y \). The proof can be immediately generalized to any number of parameters. Define

\[
g'(x, y) = J^4(x, 0, y, g(y))
\]

and let

\[
h'(z) = J^4(J^4[1](z), m(z), J^4[3](z), h(b(z), J^4[2](z), J^4[3](z), J^4[4](z)))
\]

where

\[
m(z) = \text{MSP}(|J^4[1](z)|, |J^4[1](z)| - |J^4[2](z)| - 1),
\]

\[
b(z) = \text{bit}(|J^4[1](z)|, |J^4[1](z)| - |J^4[2](z)| - 1),
\]

20
Then it is easy to see that
\[ g'(x, y) = J^4(x, 0, y, F(0, y)) \]
and, for any \( i < |x| \),
\[
\begin{align*}
  h'(J^4(x, \text{MSP}(|x|, i + 1), y, F(\text{MSP}(|x|, i + 1), y))) &= J^4(x, \text{MSP}(|x|, i), y, F(\text{MSP}(|x|, i), y)).
\end{align*}
\]
Therefore,
\[
WBRN(g, h_0, h_1, k) = g' \circ WBIN(J^4[1], h', k') \circ J^4[4]
\]
where, for any \( i < |x| \),
\[
\begin{align*}
k'(z) &= J^4(J^4[1](z), J^4[1](z), J^4[3](z), k(|J^4[1](z)|, J^4[3](z))) + z \\
&\geq J^4(J^4[1](z), \text{MSP}(|J^4[1](z)|, i), J^4[3](z), \text{MSP}(|x|, i), J^4[3](z))) + z \\
&\geq h^4(z).
\end{align*}
\]
Here, we have assumed without loss of generality that \( k \) is monotone increasing.

Now, we obtain a new characterization of \( NC \) by means of CIN and WBIN

**Theorem 40.**

\[ NC = \text{clos}(I, s_0, s_1, \text{rem}_2, \text{us}, \text{len}, \text{mp}; \text{SUBST}, \text{CIN}, \text{WBIN}). \]

**Proof.** From Lemma 38, Lemma 39, Theorem 25, Lemma 19 and Lemma 27, we have
\[
NC \subseteq \mathcal{F}_{40}.
\]
On the other hand, to prove that
\[
\mathcal{F}_{40} \subseteq NC,
\]
it suffices to show that
\[
\text{rem}_2, \text{us}, \text{add}, \text{mp} \in \text{clos}(C_0, I, s_0, s_1, \#, \text{len}, \text{bit}; \text{SUBST}, \text{CRN}, \text{WBRN}).
\]
To this aim, the reader may see Section 3.1 of [3].

Finally, by applying the Simulation Lemma, we obtain easily the following result.

**Theorem 41.**

\[ NC_1 = \text{clos}(s_0, s_1, \text{rem}_2, \text{us}, \text{len}, \text{mp}; f + g, f \circ g, \text{CIN}, \text{WBIN}). \]

**Proof.** The proof is analogous to that of Theorem 35.
References


