Minimal DFA's for Divisibility Testing (LSB first)

Jez Snelson & Joshua Obayomi

January 30, 2026

1 Introduction

2 Non-Distinguishability Criteria

In order to write this proof we have to define a way of working with binary strings $s_x \in \{0, 1\}$ such that we can assign a value to them based on the Least Significant Bit first processing order.

Definition 2.1. We define the value of a binary string according to LSB first as

$$val(c_1 \circ c_2 \circ \cdots \circ c_n) = 2^0 \times c_1 + 2^1 \times c_2 + \cdots + 2^{n-1} \times c_n$$

We want to determine the minimal amount of states for Divisibility Testing a binary string by a number p. To start with we want to declare our alphabet $\{0, 1\}$, in this paper we will denote binary strings as $s_x \in \{0, 1\}^*$ and their corresponding values $x \in \mathbb{N}$ (also notated as $val(s_x)$). For our accepting criteria we have that s_x is accepting if $val(s_x) = x \equiv 0 \pmod{p}$
In order to prove we have a minimal amount of states we rely on the Myhill Nerode Theorem and so the first thing we must do is reason with non distinguishability criteria.

Definition 2.2. We say that two strings s_x, s_y are non distinguishable if and only if $\forall s_w \in \Sigma^*$ the run of $s_x \circ s_w$ is accepting if and only if the run of $s_y \circ s_w$ is accepting.

And so with respect to our problem we have that two binary strings s_x, s_y are non distinguishable if and only if $\forall s_w \in \{0, 1\}^*$ $val(s_x \circ s_w) \equiv 0 \iff val(s_x \circ s_w) \equiv 0$

2.1 Initial ND Criteria

Definition 2.3. We define the residue of a string with respect to a certain base to be

Lemma 2.1. *The binary strings s_x, s_y are non distinguishable if and only if* $\forall d \in \mathbb{N}\ r_2(s_x)d + x \equiv 0 \iff r_2(s_y)d + y \equiv 0$

Proof. □

2.2 Revised ND Criteria

Lemma 2.2. *For all $\alpha \in \mathbb{Z}/p\mathbb{Z}$ there exists $\alpha^{-1} \in \mathbb{Z}/p\mathbb{Z}$ such that* $a \times a^{-1} \equiv 1$ *(mod p) if a, p are coprime and $a \neq 0$*

Proof. If we pick a and we have that α and p are coprime we have by Bezout's identity we have that there exists integers x and y such that $\alpha x + py = 1$ which implies that

$$\alpha x + py \equiv \alpha x + 0 \equiv \alpha x \equiv 1 \pmod{p}$$

And so we take $\alpha^{-1} = x$ which fulfills our required properties □

Lemma 2.3. *The strings s_x, s_y are non distinguishable if and only if* $(r_2(s_x))^{-1}x \equiv (r_2(s_y))^{-1}y$ *(mod p)*

Proof. □

3 Equivalence Relation Classes

As we have shown our distinguishability equivalence relation $_{d,p}$ is equivalent to $r_2(s_y)x \equiv r_2(s_x)y \pmod{p}$ and we want to construct our distinguishing set from this which leads us to.

Lemma 3.1. *The amount of equivalence classes under $=_{d,p}$ is exactly p*
Also said as $\Sigma^*/ND = p$

Proof. Firstly since there is only p possible values for the numbers to be congruent to mod p as they are integers we have that the amount of equivalence

classes is $\leq p$.
Now all we need to do is find p possible equivalence classes of distinguishability which will force it to be p.
Consider the strings of 0 to $p-1$.
Prepend 0s to the start of these strings to make them all the same length so we have $r_2(s_x) = \alpha$ for all of them.
We then have that they are all distinct under $=_{d,p}$ as for any two such strings $s_x, s_y, x \neq y$ assuming they are non distinguishable we have by 2.3

$$(r_2(s_x))^{-1}x \equiv (r_2(s_y))^{-1}y \text{ (mod p)}$$
$$\alpha^{-1}x \equiv \alpha^{-1}y \text{ (mod p)}$$
$$x \equiv y \text{ (mod p)}$$

This forms a contradiction as we picked them to be distinct numbers between 0 and $p-1$ and so they must all be distinguishable and hence in different equivalence classes.
Thus we have found p distinct equivalence classes and so the amount of equivalence classes is exactly p. $\square$

4 Even Numbers

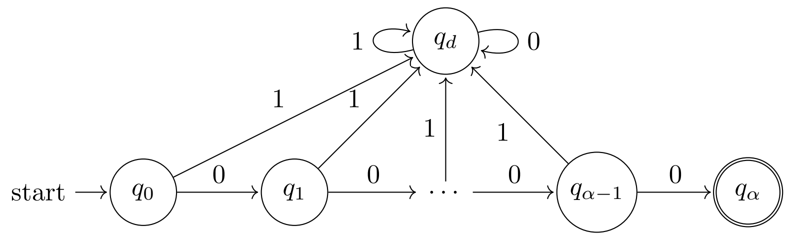
Now we aim to extend our Minimal DFA's to work with even numbers specifically numbers of the form $2^\alpha p$ where $\alpha, p \in \mathbb{N}$ and p is odd.

4.1 Construction

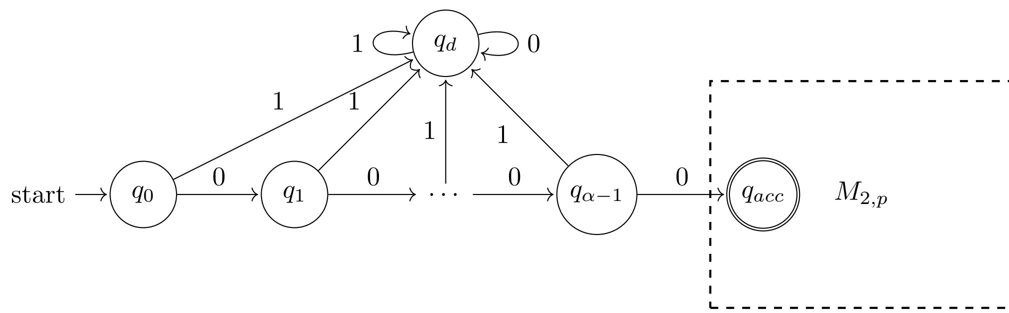
We will construct a proposed minimal DFA and prove that it works aswell as proving minimality.

4.1.1 Checking for Divisibility by 2^α

Firstly it is easy to see that for checking if a binary string is divisible by 2^α you only need to check the first α digits and so we can construct a DFA for that fairly easily.



Next we can construct a DFA as a joining of those two DFAs because if the first α digits are 0s that doesnt affect the divisibility by p.



4.2 Proof of it working

Lemma 4.1. *For any binary string s_w the run of s_w on $M_{2,2^\alpha p}$ is accepting if and only if s_w is divisible by $2^\alpha p$*

Proof. We know from our previous work that $M_{2,p}$ accepts a string s_d if and only if the string is divisible by p.
All strings that are divisible by $2^\alpha p$ have the value of $2^\alpha d$ for any s_d divisible by p and because of this we can write any string divisible by $2^\alpha p$ as $s_w = 0^\alpha \circ s_d$ for some s_d divisible by p. If we partially calculate where the run on $M_{2,2^\alpha p}$ will end up at after processing 0^α we get that it ends up at the start state (also the accepting state) of $M_{2,p}$ and then since s_d is divisible by p the run from the start state of $M_{2,p}$ will be accepting and so we have proven the machine for all s_w

□

4.3 Reachability

Lemma 4.2. *For all q_m in the states of $M_2, 2^\alpha p$ there exists a string s_w such that the run of s_w ends at q_m*

Proof. We split q_m into cases based on which part of the DFA it is in, we consider 3 cases.
Case 1: q_m is in the 2^α part of the DFA $0 \leq m \leq \alpha - 1$ then the string is just $s_w = 0^m$ and we know the run of this ends at q_m by definition of the 2^α part of the DFA.
Case 2: $q_m = q_d$ in which case the string is $s_w = 1$ and we can see the run of this ends at q_d
Case 3: q_m is a state in $M_{2,p}$ in which case we can construct s_w by first taking the string 0^α which will take us to the accepting state (also the start state of $M_{2,p}$) and then concatenating it with a string from the equivalence class of q_m and so we can the run of this ends at q_m by definition of $M_{2,p}$ □

4.4 Distinguishability

Lemma 4.3. *For all q_m in the states of $M_{2,2^\alpha p}$ where $q_m \neq q_d$ we have that there exists a string s_w such that running the string w from q_m we get to the accepting state.*

Proof. We split q_m into cases based on which part of the DFA it is in, we consider 2 cases.
Case 1: q_m is in the 2^α part of the dfa $0 \leq m \leq \alpha - 1$, if this is the case we have the trivial string of $s_w = 0^{\alpha - m}$ which we can see will go to the accepting state.
Case 2: q_m is a state in $M_{2,p}$ if this is the case we know that after running the string s_w at the state the value will be $r_2(s_x)w + x$ and so all we need to do is find a string s_w such that $w \equiv -(r_2(s_x))^{-1}x$ which we know we can find as $r_2(s_x)^{-1}$ must exist by 2.2 □

Lemma 4.4. *For all q_m q_n in the states of $M_{2,2^\alpha p}$ $m \neq n$ we have that q_m is distinguishable from Sqn*

Proof. We split q_m and q_n into cases based on which part of the DFA they are in we consider 3 cases for each of them being the dead state q_d, being any other state in the divisibility by 2^α part $q_l, 0 \leq l \leq \alpha - 1$ or being any state in $M_{2,p}$.
Case 1: $q_m \neq q_d$ and $q_n = q_d$. In this case we have that the distinguishing string is the accepting string of q_m which we know exists from 4.3 with this q_m will go to the accepting state and q_n will stay at the dead state and so they are distinguishable.
Case 2: q_m is in the 2^α part of the dfa $0 \leq m \leq \alpha - 1$ and q_n in the states of $M_{2,p}$, in this case for our distinguishing string we first take 1 which will take q_m to q_d and q_n to another state in $M_{2,p}$ as no transitions leave the machine. We then take the rest of the distinguishing string to be the accepting string of the state q_n goes to and then we have that the run on q_n will be accepting

and the run on q_m will go to the dead state and stay there and so they are distinguishable.
Case 3: q_m, q_n are both in the 2^α part of the dfa $0 \leq m, n \leq \alpha - 1$, in this case the distinguishing string is $0^{\alpha - \max m,n}$ which will take the state the closest to the accepting state to the accepting state and leave the other state still in the 2^α part of the DFA and so these states are distinguishable.
Case 4: q_m, q_n are both in $M_{2,p}$ we know these states are all distinguishable from our construction of the machine using equivalence clases.

□