UNIVERSITY

DOCTORAL THESIS

A Language of Polynomials

Author:
Eric UNG

Supervisor:
Dr. xx XX

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Research Group Name
Department or School Name

May 25, 2024

Declaration of Authorship

I, Eric UNG, declare that this thesis titled, "A Language of Polynomials" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism."

Dave Barry

UNIVERSITY

Abstract

Faculty Name
Department or School Name

Doctor of Philosophy

A Language of Polynomials

by Eric UNG

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too...

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor…

Contents

List of Figures

List of Tables

List of Abbreviations

LAH	**L**ist **A**bbreviations **H**ere
WSF	**W**hat (it) **S**tands **F**or

Physical Constants

Speed of Light $c_0 = 2.997\,924\,58 \times 10^{8}\,\mathrm{m\,s^{-1}}$ (exact)

List of Symbols

a	distance	m
P	power	W (J s^{-1})
ω	angular frequency	rad

For/Dedicated to/To my…

Chapter 1

A Language of Polynomials

1.1 Introduction

This paper is on the re-framing of the one way function to a matrix multiplication problem - that of multiplying two 3×3 matrices to form a 6×6 matrix under a locally concatenative property.

1.2 Foundations

There exists a language such that it decides each monomial in the polynomial. In other words, there exists a set of deciders for each monomial in the polynomial where it decides if y is in the monomial. A decider in this term is not of the definition found originally in textbooks but one that is redefined in the below definition.

Given a polynomial
$p(x) = ax^2 + bx + c$
$p(x) = 3x^2 + 4x + 5$
$p(2) = 3(2)^2 + 4(2) + 5$
$p(2) = 12 + 8 + 5$

Let the decider be defined as the following:

Decider is a function $Decider < c \times x^{degree} >\equiv c \times x^{degree} = y$
such that $x_1 \times x_2 \times ... \times x_n$ where n is equal to degree + constant is tested to be equivalent to y and x_1 is the start and *x degree times* is the finish then loop around $x_1 tox\ degree\ times$ until it stops

For each state, x_i, i such that it is between 1 to n, x_i contains a subgroup of size n and for each subgroup, s_i, there exists another subgroup and so on and so forth such that there are n layers starting from x_i to 1. This is the same as saying that it is a rational expression.

A rational expression is a expression that satisfies the following.

$A_n = A_{n-1} \cup \{E^* | E \in \mathcal{E}_{n-1}, (E, 1) = 0\}$

It follows that each state $E \in s_{n-1}$ forms a subgroup.

A rational function is defined as the following:
K[x] and K[[x]]. Let K[[x]] describe a set of deciders. S is an element of K[[x]] meaning S is a decider.
S = $\sum_{n \geq 0} a_n x^n$

Examples
Decider for ax^2 is $Decider < 3(2)^2 > \equiv 3(2)^2 = 12$
Decider for bx is $Decider < 4(2)^1 > \equiv 4(2)^1 = 8$
Decider for c is $Decider < 5(2)^1 > \equiv 5(2)^0 = 5$

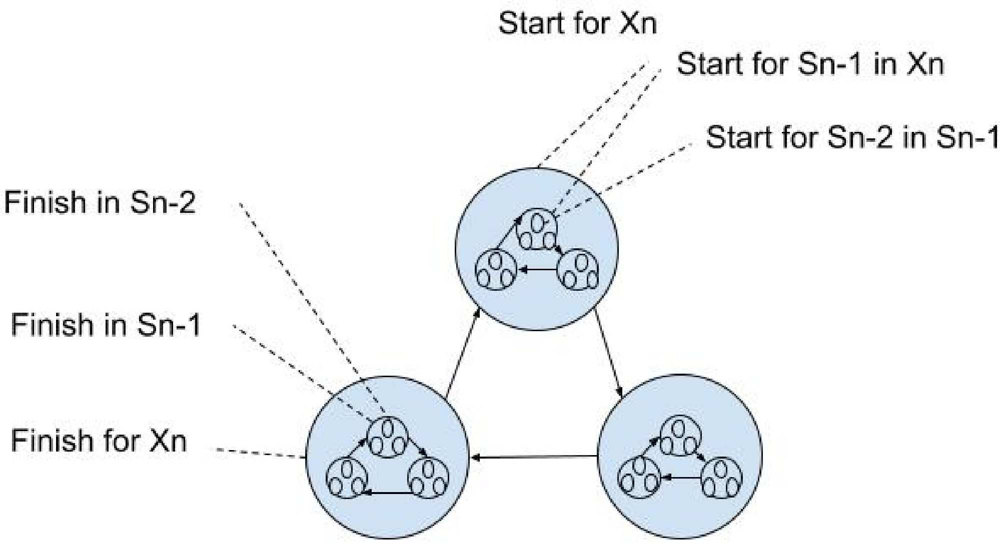

FIGURE 1.1: Decider X to the 3.

Figure 1.1 shows a monomial decider, $Decider < x^n >$ with that represents x^3.

Theorem: A Decider is the equivalent to a cyclic automata

Proof: Remove the lowest level state, S_1 from the bottom then continue removing S_i from i = 2 to n-1 until you get only the states that are at X_n.

Remove the top state down from the S_{n-1} for each layer S_{n-1} to $S_(n-n-1)$. This preserves the start and finish state for the layer x_n. This is a cyclic automaton.

Figure 1.2 shows how the intuition for removing the state top down.

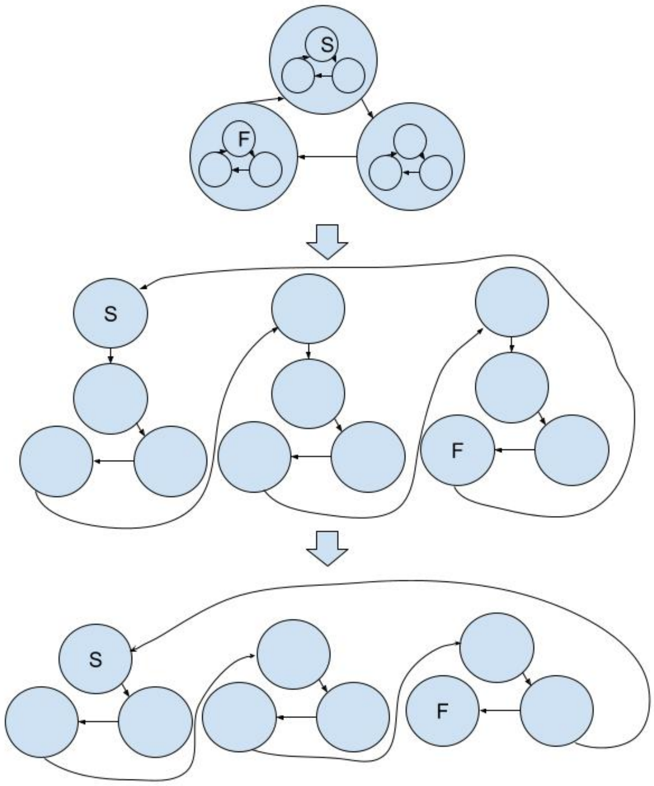

FIGURE 1.2: Top down removal for equivalence of decider and cyclic automata.

1.3 Monomial of One Variable

Given the definition of a decider:

Decider is a function $Decider < cx^n > \equiv cx^n = y$

A decider of at least one degree
$Decider < 3x^4 > \equiv 3x^4 = y$

Contains $Decider < 3x^3 > \equiv 3x^3 = y$
Contains $Decider < 3x^2 > \equiv 3x^2 = y$
Contains $Decider < 3x^1 > \equiv 3x^1 = y$
Contains $Decider < 3x^0 > \equiv 3x^0 = y$
Hence it can be generalized to:
$Decider < cx^n >$ contains the sequence set

$Decider < cx^n >, Decider < cx^{n-1} >, ..., Decider < cx^0 >$ which is equivalent to a rational expression by definition.

There exists a start state and a finish state for each decider.
$\{start, ..., finish\}$

1.4 Addition

Given the first example:

$p(x) = 3x^2 + 4x + 5$
$p(2) = 3(2)^2 + 4(2) + 5$
$p(2) = 12 + 8 + 5$

$m2 = Decider < 3x^2 >= 3x^2$
$m1 = Decider < 4x^1 >= 4x$
$m0 = Decider < 5x^0 >= 5$

Generalized to m_x where x is the degree
Given polynomial functions, p_1 and p_2, they are commutative
$p_1(x) = m_a + ... + m_0$
$p_2(x) = n_b + ... + n_0$
$p_1(x) + p_2(x) = m_a + n_b + (m_{x+1} + n_{y+1}) + ... + (m_x + n_y) + ... + (m_0 + n_0)$ where $x = y$
$Decider < c_x x d_x > + Decider < c_y x d_y >$
$= Decider < c_x + c_y, x, d_x >= Deciders < c_x + c_y, x, d_y >$
$\implies d_x = d_y$

1.5 Product

Given two monomials in the language, a and b, the product of a and b is also in the language.

Given $Decider < c_x x d_x >$ and $Decider < c_y x d_y >$ is in language L
Show that the product $Decider < (c_x + c_y) x d_x \times d_y >$ is in L
$Decider < c_x x d_x > \times Decider < c_y x d_y >$
$= c_x x d_x c_y x d_y$
$= c_x x d_x + d_y$
$= (c_x + c_y) x (d_x + d_y)$ is in L
$= Decider < (c_x + c_y) x (d_x + d_y) >$

1.6 Problem with Matrices

An important problem arising from deciders is representing them as matrices. The problem can be reformulated as the following: given a polynomial p of x, show that the monomial deciders represented in the language can't be contained in a finite matrix after a set number, n, such that x^n.

$[a \times a]\ [b \times b] = [n \times n]$ such that $a \neq b$ and $a, b < n$

1.7 Multivariable Polynomials

A monomial with more than one variable can be treated the same way as handling single variables at different degrees.

Decider(c,xyz,d) = c(xyz)d = Decider(c,x,d) × Decider(c,y,d) × Decider(c,z,d) where c is some constant Decider(c,x,d1) × Decider(c,y,d2) × Decider(c,z,d3) = cxd1 × yd2 × zd3 where c is some constant

Given f(x,y)=3xy2
set x=2,y=3
f(2,3)=3(2)(3)2
f(2,3)=27

1.8 Generalized Monomial Deciders

A monomial decider can be represented in a more general graphic

6x6=Decider 6,x,6 =6 × xstart,x2,x3,x4,x5,xfinish x6=Decider 1,x,6 =1 × xstart,x2,x3,x4,x5,xfinish

In both these examples, xstart is x1 and xfinish is x6.

1.9 Concentric Monomial Deciders

Given a polynomial, p x, with a monomial decider represented as ax n in p x and n in N and a = 1 such that p x = x n, there is special property for these monomial deciders that can be illustrated below.

Decider 1,x,4 = x = xi such that i is in start,2,3,finish and xi contains xi minus 1 where i 1 Decider 1,x,2 = x2 = xi such that i is in start,finish and xi contains xi minus 1 where i 1 In both these examples, xi is a state in the monomial decider. Each state in a one constant monomial decider have a property of being concentric. This means that a state can be defined as, x2 = x1,x0 so going from x1 to x0 then going to the next state x3 or looping back to x1.

1.10 Constants

Given a constant, c, of p or better described in the example: f(x) = 5. Constants are seen as linear directed acyclic graphs.

f(x) = 5

$Decider(5, x, 0) \equiv 5 \equiv start, 2, 3, 4, finish$

There is no state in the decider where it loops back to the start. In other words, there is no x that represents a monomial in a constant.

1.11 Division

Division of monomial deciders

$x^5/x = x^4 \equiv Decider(1,x,5)/Decider(1,x,1)$ $Decider(1,x,4)$ $x^5/x^2 = x^3 \equiv Decider(1,x,5)/Decider(1,x,2)$ $Decider(1,x,3)$ $x^5/x^3 = x^2 \equiv Decider(1,x,5)/Decider(1,x,3)$ $Decider(1,x,$ $x^5/x^4 = x^1 \equiv Decider(1,x,5)/Decider(1,x,4)$ $Decider(1,x,1)$ $x^5/x^5 = 1 \equiv Decider(1,x,5)/Decider(1,x,5)$

here is represented as a special kind of equivalence that we will get to later.

$Decider(1,x,5)/Decider(1,x,1) \equiv sequenceofpermutationsofxi, xjsuchthatthecountofiis4andjis1$
$Decider(1,x,5)/Decider(1,x,2) \equiv sequenceofpermutationsofxi, xjsuchthatthecountofiis3andjis2$
$Decider(1,x,5)/Decider(1,x,3) \equiv sequenceofpermutationsofxi, xjsuchthatthecountofiis2andjis3$

$Decider(1, x, 5)/Decider(1, x, 4) \equiv sequenceofpermutationsofxi, xjsuchthatthecountofiis1andjis4$
$Decider(1, x, 5)/Decider(1, x, 5) \equiv sequenceofpermutationsofxi, xjsuchthatthecountofiis0andjis5Here,$
$xj, meaningxiisofadifferentrepresentationthanxj$

1.12 Multiple Divisions

Given multiple operations of division, this forms an interesting space. x5/x2/x=x5/x)/x2

Decider 1,x,5 /Decider 1,x,2 /Decider 1,x,1 =Decider 1,x,5 /Decider 1,x,1 /Decider 1,x,2 iff ignoring order of operations

1.13 Equivalence

Decider 1,x,6 /Decider 1,x,1 /Decider 1,x,1 Decider 1,x,6 /Decider 1,x,2 iff the order of operations is next to each other

Determining if y is in f x is easy if we are given any monomial decider in the set of the language of polynomials and their representations has the possibility to give different representations if we consider them as representations of the function f x.

Decider 1,x,6 /Decider 1,x,1 /Decider 1,x,1 =sequence of permutations of xi,xj,xk such that the count of i is 4 and j is 1 and k 1

Decider 1,x,6 /Decider 1,x,2 =sequence of permutations of xi,xj such that the count of i is 4 and i is 2

Theorem of Equivalence Decider 1,x,6 /Decider 1,x,1 /Decider 1,x,1 =Decider 1,x,6 /Decider 1,x,2 such that there is some xsuch that the monomial represented by both deciders exists where f x =y

1.14 Theorem of Equivalence

$Decider(1, x, 6)/Decider(1, x, 1)/Decider(1, x, 1)$
$\equiv Decider(1, x, 6)/Decider(1, x, 2)$
such that there is some x
such that the monomial represented by both deciders exists where $f(x) \equiv y$

1.15 Reversing

$Decider(1, x, 6)/Decider(1, x, 1)/Decider(1, x, 1) \equiv sequence\ of\ permutations\ of\ xi, xj, xk\ such\ that\ the$

$Decider(1, x, 6)/Decider(1, x, 2) \equiv sequence\ of\ permutations\ of\ xi, xj\ such\ that\ the\ count\ of\ i\ is\ 4\ an$

Is shown that by the permutation of the order of operations that $Decider(1, x, 6)/Decider(1, x, 1)/D$ is not the same set as $Decider(1, x, 6)/Decider(1, x, 2)$

Theorem of Reversing

Given two deciders x,y in a decider of m(x), x != y iff sequence of all the states are not equivalent, representation-wise, from start to finish or it doesn't represent the same order of operations of the deciders being represented

1.16 Corollary of Reversing

A little more on the theorem of reversing Given a starting point, the paths a monomial decider takes to decide if y is in f x is inherently unique to each representation.

Start at the circle, S, and end at the circle, F. If the circle is white, it is 0. If it is blue, it is 1. As an example let's traverse some of the representations of x 4.

Corollary Given a decider, d, in Decider of m x then there is path, p, that exists for d such that p = Path d = s1,s2,...,si,...,sn where i is count of the states in the decider of m x.

Example: Choose some x such that it is in Path Decider 1,x,6/Decider1,x,2 where p = 001111

1.17 Godel's Theorem

We see that there exists two statements from these theorems

1. x = x from a theorem of equivalence 2. x != x from a theorem of reversal

Example: Given some d1,d2,d3,...,infinity in deciders of m x 1. d1 = d2 = d3 = ... = infinity 2. d1 != d2 != d3 = ... != infinity

The different representations of a monomial through the language of monomial deciders will give us undecidability. This means we can come up with many formal definitions of the mnomial decider and it will not be able to solve the problem of finding a specific representation of a monomial decider without having to guess or apply some sort of probability to it. Relating to the real line, given a real line a,b a $\leq$ b, there is infinite choices between a and b because we canjust make the number smaller. As long as b a $\geq$ 0, there requires some sort of probability of choosing some specific number that is between a and b.

1.18 Reframing The One Way Function

A probability exists to find a certain monomial decider in the set of it's variations. A/B = Probability where A is the monomial decider we want and B is the number of all the variations.

Example: d1,d2,...,d6 in deciders 1,x,6 ,D, such that di are all distinct Choose one of the deciders in D through probability Probability of choosing d in D is 1/6 so 0.16666667 We'll call this picking a function and every time we call this function, the probability is mulltiplied such that it is n k. As an example, if we call the picking function twice using the example above, we have 1/6 1/6 = 1/36 = 6 2.

This is formally known as the one way function.

1.19 Theorem of Infiniteness

Given that, if we can show for any language it abides a theorem of equivalence and a theorem of reversal and they both have infinite representations, we know that we need some sort of probability to choose a specific representation of a monomial decider. We'll call this a theorem of infiniteness because if we can show that some language is infinite, it will require some sort of guess to pick something unique out of all the things it represents, generates, or describes. In other words, you can't map infinity to infinity directly for you must map infinity to something discrete and something discrete to infinity.

Theorem A mapping must be from infinity to discrete representation to discrete representation to infinity.

Given any representation of infinity, Suppose a mapping infinity to infinity exists. Then this map is equivalent to infinity because infinity contains this map. Here, infinity is represented as $*$. $*\ contains * -> *$.

Intuition

Given Decider c,x,d and d is infinity, Decider c,x,d contains Decider c,x,d 1 d 1 is then infinity too.

1.19.1 References

The `biblatex` package is used to format the bibliography and inserts references such as this one (**Reference1**). The options used in the `main.tex` file mean that the in-text citations of references are formatted with the author(s) listed with the date of the publication. Multiple references are separated by semicolons (e.g. (**Reference2**; **Reference1**)) and references with more than three authors only show the first author with *et al.* indicating there are more authors (e.g. (**Reference3**)). This is done automatically for you. To see how you use references, have a look at the `Chapter1.tex` source file. Many reference managers allow you to simply drag the reference into the document as you type.

Scientific references should come *before* the punctuation mark if there is one (such as a comma or period). The same goes for footnotes[1]. You can change this but the most important thing is to keep the convention consistent throughout the thesis. Footnotes themselves should be full, descriptive sentences (beginning with a capital letter and ending with a full stop). The APA6 states: "Footnote numbers should be superscripted, [...], following any punctuation mark except a dash." The Chicago manual of style states: "A note number should be placed at the end of a sentence or clause. The number follows any punctuation mark except the dash, which it precedes. It follows a closing parenthesis."

The bibliography is typeset with references listed in alphabetical order by the first author's last name. This is similar to the APA referencing style. To see how LaTeX typesets the bibliography, have a look at the very end of this document (or just click on the reference number links in in-text citations).

A Note on bibtex

The bibtex backend used in the template by default does not correctly handle unicode character encoding (i.e. "international" characters). You may see a warning about this in the compilation log and, if your references contain unicode characters, they may not show up correctly or at all. The solution to this is to use the biber backend instead of the outdated bibtex backend. This is done by finding this in `main.tex`: *`backend=bibtex`* and changing it to *`backend=biber`*. You will then need to delete all auxiliary BibTeX files and navigate to the template directory in your terminal (command prompt). Once there, simply type `biber main` and biber will compile your bibliography. You can then compile `main.tex` as normal and your bibliography will be updated. An alternative is to set up your LaTeX editor to compile with biber instead of bibtex, see here for how to do this for various editors.

[1] Such as this footnote, here down at the bottom of the page.

Chapter 2

Applications For Monomial Deciders

2.1 Starting With A Theorem Of Infiniteness

This section assumes that both P=NP and P!=NP and the following sections will provide reasoning and examples.

2.2 Mapping Out Representations

This section contains my opinions of what mathematics is.

2.3 Euler's Constant

Euler's constant is an example of P = NP because of it's use of calculus.

To show that it is also in the problem set of P $\neq$ NP, start by using the picking function going into infinity.

$$e = pf(x) = \theta(language\ of\ pf(x))$$

$$\{x = x^2/x = x^3/x^2 = x^4/x^3 = x^5/x^4 = \cdots = x^n/x^{n-1}\}$$

$$1 + 1 + 1/(1+1) + 1/(3+3) + 1/(4 + (4*3 + 4*2) + 4)$$

2.4 Sketching Into Code

There is a technique to develop code from a diagram of a decider. Take into account the degree of the states and that should account for the halting required to break from the loop.

```
bool generalizedMD(int y)
{
    if (y == 0)
    {
        return true;
```

```
    }
    var s = y;
    while (s >= 0)
    {
        for (int i = 0; i < 2; i++)
        {
            for (int j = 0; j < 4; j++)
            {
                if (s == 0 && (j == 0 || j == 2) && i == 0)
                {
                    return true;
                }
                else if (s < 0)
                {
                    return false;
                }
                s -= 1;
            }
        }
    }
    return true;
}
```

2.5 Gather Some Data

Using the generalizedMD function where we make some sketch of the initial algorithm, we can collect some data Generate:
f(x) = y on the first line
Negatives on the second

We notice that we need two finishing states and that all the ven y's end in one state and all the odd y's end on the other.
E: Is it even?

We create a for loop of n^3 that with constant 2 and count how many times it passes through the finishing state.
T: Total amount of times it passes the finishing state.

Insight is gained by noticing that the difference only increases every other time and that it increases by a difference of 2 every it passes a finishing state.
D: Difference of the number of hits to the finishing state between the alst time it is called and the first.

```
// Generates negatives from a general monomial decider represented as an al
// that we can collect data about the negatives
```

```
int[] Generator(int max)
{
    int[] result = new int[max + 1];
    int x = 0;
    int negatives = 0;
    int i = 0;
    while (x < max + 1)
    {
        int num = 2 * (Convert.ToInt32(Math.Pow(x, 2)));
        if (generalizedMD(i))
        {
            // A simple verifier
            if (num == i)
            {
                Console.WriteLine(negatives);
                Console.WriteLine("f(" + x + ") = " + i);
                result[x] = i;

                x++;
                negatives = 0;
            }
            else
            {
                negatives++;
            }
        }
        i++;
    }

    return result;
}
```

2.6 Representing Monomial Deciders As Code

```
// After getting some log results, we can construct a decider
bool MonomialDecider2xx(int y)
{
    var total = 0;
    var hits = 0;
    var diff = 1;
    var isEven = 0;
    var s = 0;

    while (s <= y)
    {
        for (int i = 0; i < 2; i++)
        {
            for (int j = 0; j < 2; j++)
            {
                for (int k = 0; k < 2; k++)
```

```
                {
                    if ((i == 0)&&(j == 0 || j == 1)&&(k == 0))
                    {
                        if (hits == total)
                        {
                            if (s == y)
                            {
                                Console.WriteLine(s + ":␣Hits:␣" + total);
                                return true;
                            }
                            else if (s > y)
                            {
                                return false;
                            }

                            total += diff;
                            isEven++;
                            if (isEven % 3 == 2)
                            {
                                isEven = 0;
                                diff += 2;
                            }
                        }

                        hits++;
                    }
                    s++;
                }
            }
        }
    }

    return false;
}
```

2.7 Negative Numbers

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

2.8 Pi

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

2.9 Fibonacci

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

2.10 Analysis Of Parity In Fibonacci

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

2.11 Redrawing the Fibonacci Sequence

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

2.12 The Fibonacci Decider

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

2.13 The Fibonacci Picking Function

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

Chapter 3

Inferrable Languages

3.1 Applying The Fibonnaci Decider

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.2 Fibonacci DOL Decider Left Hand Side

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.3 Fibonacci DOL Decider Right Hand Side

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.4 The Law of Commutativity and Noncommutativity

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.5 Definition Of Support

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.6 Rationals Of Picking Function

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.7 Support Of Picking Function

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.8 Law Of Strings

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.9 Commutativity Of Addition

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.10 Commutativity Of Multiplication

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.11 Additive Identity

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.12 Multiplicative Identity

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.13 Additive Inverse

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.14 Multiplicative Inverse

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.15 Generalized Operations

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.16 Generalized Communativity

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.17 Associativity Of Addition

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.18 Associativity Of Multiplication

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.19 Distibutivity

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.20 Field

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

Appendix A

Frequently Asked Questions

A.1 How do I change the colors of links?

The color of links can be changed to your liking using:
`\hypersetup{urlcolor=red}`, or
`\hypersetup{citecolor=green}`, or
`\hypersetup{allcolor=blue}`.
If you want to completely hide the links, you can use:
`\hypersetup{allcolors=.}`, or even better:
`\hypersetup{hidelinks}`.
If you want to have obvious links in the PDF but not the printed text, use:
`\hypersetup{colorlinks=false}`.