Generating Similar Images Using Bag Context Picture Grammars

Liemiso Hycinth Mpota

0705433r

Supervised by: Prof. Sigrid Ewert

A Dissertation submitted to the Faculty of Science

in partial fulfilment

of the requirements for the degree of

Master of Science

University of the Witwatersrand, Johannesburg

February 12, 2018

Abstract

Formal language theory was born in the middle of the 20th century as a tool for modeling and investigating syntax of natural languages. It was developed in connection with the handling of programming languages. Bag context grammars are a fairly new grammar class where bag context tree grammars have been defined. Bag context is used to regulate rewriting in tree grammars.

In this dissertation we use bag context to regulate rewriting in picture grammars and thus to generate similar pictures. This work is exploratory work since bag context picture grammars have not been defined. We use examples to show how bag context picture grammars can be used to generate pictures. In this work bag context picture grammars are defined and used to generate similar pictures. Pictures generated by random context picture grammars and three of their sub-classes are selected and bag context picture grammars are used to generate the same pictures to those selected. A lemma is defined that is used to convert the class of random context picture grammars and three of their sub-classes into equivalent bag context picture grammars. For each grammar selected, an equivalent bag context picture grammar is created and used to generate several pictures that are similar to each other. Similarity is defined by noting small differences that are seen in pictures that belong to the same gallery. In this work we generate similar pictures with bag context picture grammars and thus make the discovery that bag context gives a certain level of control in terms of rules applied in a grammar.

Publications

An extended abstract from Chapter 4 of this paper was submitted and presented at the South African Institute for Computer Scientists and Information Technologists (SAICSIT) M and D symposium in September 2016.

Declaration

I, Liemiso Hycinth Mpota (0705433r), hereby declare the contents of this dissertation to be my own work unless otherwise explicitly referenced. This dissertation is submitted for the degree of Master of Science (Dissertation) at the University of the Witwatersrand, Johannesburg. This work has not been submitted to any other university, or for any other degree.

. . . .
Signature

Acknowledgement

I would like to sincerely thank my supervisor, Prof. Sigrid Ewert from the University of the Witwatersrand. The original idea for this project was proposed by my supervisor and for that I am very grateful. I am also thankful to Prof. Sigrid Ewert for guiding and supporting me in writing this dissertation. By being there every step of the way teaching me how to write a scientific paper and how to use correct language and grammar. I applaud her for her patience with me throughout this entire process I have learned a lot from her. If it was not for her diligence, guidance and support I would not be this far with my work.

I would also like to thank the government of Lesotho for their contribution in terms of offering me a scholarship under the National Manpower Development Secretariat Scholarship Fund for assisting in the payment of my tuition fees and living expenses for the years 2015 and 2016. In addition I would also like to thank the the School of Computer Science and Applied Mathematics together with Prof. Sigrid Ewert for funding my studies for the year 2017. I would not have finished this project if it was not for their financial support.

I would also like to thank Dr Abejide Ade-ibijola for his mentorship. He has given me invaluable advice and taught me many skills. I would like to thank him for always availing himself every time I needed his guidance. Dr Ade-ibijola has also let me stand in as an assistant to his robotics presentation at Future Nation Schools. Mostly I would like to thank him for always making me believe I can achieve anything, for giving me exposure in the field of computer science and for being a great inspiration always leading by example.

I would like to thank Mrs Nuru Jingili, a PhD student in the department if Computer Science and Applied Mathematics. She allowed me to use her program to generate my pictures. She wrote a program that takes in bag context picture grammars and draws the picture in python. Nuru was also a great pillar of strength and support often helping me out and availing herself when I had questions.

I would like to thank Miss Precious Shabalala and Miss Kgomotso Monyepote for always being available with any assistance with administration problems. They have both always been so kind and helpful to me. Miss Monyepote has helped me with giving me invigilation and tutoring work that relieved me financially countless times and I will forever be grateful to her. These ladies are both humble souls and I am grateful for their presence in my life.

I would like to thank my parents Mantoa and Phethang Mpota for always believing in me and pushing me. For having patience with me during trying times. For loving me unconditionally and always giving me advice and support. I would not have made it this far in life if it was not for them. My siblings Tsonngoe and Tsongoane Mpota, life would be meaningless without you. I am also thankful for my nephew Tumelo Mpota and my nieces Boitumelo Pitikoe and Mothepane Mpota. They are the reason I want to achieve more in life. I want them to have someone to look up to.

Lastly I would like to thank my friends and there is a lot of them. Nts'eliseng Qopana who has been a pillar of strength to me in many ways like giving me accommodation, being my health and fitness partner, listening, laughing, inspiring me and giving me great advice. Lla Thakholi, Lerato Thakholi, Teboho Khoali, Nompilo Mtshali and Nomonde Khalo, my friends that I dream with, whom I cry with, who struggle and laugh with me. Who inspire me to do great at all times and always believing in me. Thank you ladies for being awesome in all ways possible. Blessing Ogbuokiri, Joshua Choma and Sizwe Mdagana who have helped me countless times with installations on my computer, code and general work advice and who inspire me to be as amazing as they are in this field.

Contents

Abstract **i**

Publications **ii**

Declaration **iii**

Acknowledgment **v**

Contents **vii**

List of Figures **ix**

List of Tables **x**

1 Introduction **1**
- 1.1 General Introduction . . . 1
- 1.2 Bag Context Tree Grammars . . . 2
- 1.3 Random Context Picture Grammars . . . 2
- 1.4 Roadmap . . . 3

2 Literature Review **4**
- 2.1 Definitions and Notation . . . 4
  - 2.1.1 Notation . . . 4
  - 2.1.2 Definitions . . . 5
- 2.2 Discussion and Analysis . . . 14
  - 2.2.1 Generating Perfect Images . . . 15
  - 2.2.2 Generating Similar Images . . . 15
- 2.3 Chapter Summary . . . 17

3 Research Methodology **18**
- 3.1 Introduction . . . 18
- 3.2 Research Hypothesis . . . 18
- 3.3 Research Method . . . 19
- 3.4 Chapter Summary . . . 19

4 Bag Context Picture Grammars **20**
- 4.1 Definitions . . . 20
- 4.2 Relationship between BCPGs and RCPGs . . . 22

4.3 BCPGs, RCPGs and Three Sub-Classes 22
4.3.1 Context Free Picture Grammars 22
4.3.2 Random Permitting Context Picture Grammars 24
4.3.3 Random Forbidding Context Picture Grammars 25
4.3.4 Random Context Picture Grammars 28
4.4 Picture Generation with Bag Context Picture Grammars 31
4.5 Chapter Summary 34

5 Galleries Generated by Bag Context Picture Grammars 37
5.1 Context Free Picture Grammars Case 37
5.2 Random Permitting Context Picture Grammars Case 44
5.3 Random Forbidding Context Picture Grammars Case 53
5.4 Random Context Picture Grammar Case 60
5.5 Chapter Summary 67

6 Conclusion and Future Work 68
6.1 General Conclusion 68
6.2 Future Work 68

Bibliography 71

List of Figures

2.1 Bag context tree grammar rules for binary trees 6
2.2 A production of a random context picture grammar [1] 6
2.3 Pictorial form Π 8
2.4 Random context rules that generate the Sierpiński Carpet 8
2.5 Pictorial forms that show how the picture develops during the application of rules in order to create the Sierpiński Carpet 9
2.6 Sierpiński Carpet where Figure (a) is in first level refinement and Figure (b) is in the second refinement [1] 9
2.7 L system rules that generates the fractal in Figure 2.8 11
2.8 The first to fourth iterations of the Koch snowflake [2] 12
2.9 Grammar that produces the Kolam pattern in Figure 2.10 12
2.10 Anklets of Krishna [3] 12
2.11 Rules for H token figure 14
2.12 H pattern generated by the rules in Figure 2.11 14
2.13 Similar pictures generated of RPCPGs 16
2.14 Rules of Stochastic L system that generates trees in Figure 2.15 17
2.15 Pictures of similar trees generated by SLSs [4] 17

4.1 Production of a bag context picture grammar 21
4.2 Context Free Picture Grammar rules for G_{Free} 23
4.3 Some of the pictures generated by G_{Free} 23
4.4 Bag Context Picture Grammar rules for $G_{\mathsf{Bag\text{-}free}}$ 24
4.5 Random Permitting Context Picture Grammar rules for G_{Permit} 25
4.6 Gallery of pictures generated by G_{Permit} 26
4.7 Bag Context Picture Grammar rules for $G_{\mathsf{Bag\text{-}permit}}$ 26
4.8 Random Forbidding Context Picture Grammar rules for G_{Forbid} 27
4.9 Some of the pictures generated by G_{Forbid} 28
4.10 Bag Context Picture Grammar rules for $G_{\mathsf{Bag\text{-}forbid}}$ 29
4.11 Random Context Picture Grammar rules for G_{Triangle} 30
4.12 Some of the pictures generated by G_{Triangle} [1] 31
4.13 Bag Context Picture Grammar rules for $G_{\mathsf{Bag\text{-}triangle}}$ 32
4.14 Bag Context Picture Grammar rules for $G_{\mathsf{Bag\text{-}free}}$ 33
4.15 The pictorial form after Rule 4.1 is applied 34
4.16 The pictorial form after Rule 4.2 is applied 35
4.17 The pictorial form after Rule 4.2 is applied 35
4.18 Picture generated by Figure 4.14 36
4.19 A possible picture generated by $G_{\mathsf{Bag\text{-}free}}$ 36

5.1 Rules that generate picture in Figure 5.2 38
5.2 Picture generated by G_{Cross} 39
5.3 $G_{\mathsf{Bag\text{-}free}}$ rules that generate pictures in Figure 5.4 39
5.4 Pictures generated by $G_{\mathsf{Bag\text{-}free}}$ 41
5.5 Bag Context Picture Grammar rules for $G_{\mathsf{Color\text{-}crosses}}$ 41
5.6 Some of the pictures generated by $G_{\mathsf{Color\text{-}crosses}}$ with four pink crosses . . . 43
5.7 Bag Context Picture Grammar rules for $G_{\mathsf{Blue\text{-}crosses}}$ 43
5.8 Pictures generated by $G_{\mathsf{Blue\text{-}crosses}}$ with seven blue crosses 45
5.9 Bag Context Picture Grammar rules for $G_{\mathsf{Bag\text{-}permit}}$ 46
5.10 Picture generated by the rules in Figure 5.9 47
5.11 Bag Context Picture Grammar rules for $G_{\mathsf{Permit\text{-}control}}$ 47
5.12 Picture generated by $G_{\mathsf{Permit\text{-}control}}$ 48
5.13 Bag Context Picture Grammar rules for $G_{\mathsf{Permit\text{-}free}}$ 49
5.14 Pictures that are the same to those in Figure 5.4(b) and (c) 50
5.15 Bag Context Picture Grammar rules for $G_{\mathsf{Permit\text{-}color}}$ 51
5.16 Picture with color distribution 52
5.17 Bag Context Picture Grammar rules for $G_{\mathsf{Bag\text{-}forbid}}$ 54
5.18 Two of the Pictures generated by $G_{\mathsf{Bag\text{-}forbid}}$ 55
5.19 Bag Context Picture Grammar rules for G_{Refine} 55
5.20 Some of the pictures generated by G_{Refine} 57
5.21 Bag Context Picture Grammar rules for $G_{\mathsf{Forbid\text{-}color}}$ 57
5.22 Some of the pictures generated by $G_{\mathsf{Forbid\text{-}color}}$ 58
5.23 Bag Context Picture Grammar rules for $G_{\mathsf{Refined\text{-}color}}$ 59
5.24 Some of the pictures generated by $G_{\mathsf{Refined\text{-}color}}$ 61
5.25 Bag Context Picture Grammar rules for $G_{\mathsf{Bag\text{-}triangle}}$ 61
5.26 Pictures that are generated by $G_{\mathsf{Bag\text{-}triangle}}$ 63
5.27 Bag Context Picture Grammar rules for $G_{\mathsf{Hypotenuse}}$ 63
5.28 Some of the pictures generated by $G_{\mathsf{Hypotenuse}}$ 65
5.29 Bag Context Picture Grammar rules for $G_{\mathsf{Triangle\text{-}color}}$ 65
5.30 Some of the pictures generated by $G_{\mathsf{Triangle\text{-}color}}$ 66

List of Tables

4.1 Development of the bag for picture in Figure 4.18 33

5.1 Development of bag for picture in Figure 5.2 38
5.2 Development of bag for one of the pictures in Figure 5.4 39
5.3 Development of the bag for one of the pictures in Figure 5.6 42
5.4 Development of bag for one of the pictures in Figure 5.8 44
5.5 Development of bag for the picture in Figure 5.10 45
5.6 Development of bag for the picture in Figure 5.12 48
5.7 Development of bag for the picture in Figure 5.14(a) 50
5.8 Development of bag for the picture in Figure 5.16 52
5.9 Development of bag for one of the pictures in Figure 5.18 54
5.10 Development of bag for one of the pictures in Figure 5.20 56
5.11 Development of bag for one of the pictures in Figure 5.22 58
5.12 Development of bag for one of the pictures in Figure 5.24 60
5.13 Development of bag for one of the pictures in Figure 5.26 62
5.14 Development of bag for one of the pictures in Figure 5.28 64
5.15 Development of bag for one of the pictures in Figure 5.30 66

List of Abbreviations

CFG	context free grammar
CSG	context sensitive grammar
SLS	stochastic Lindenmayer system
RCG	random context grammar
BCG	bag context grammar
RPCG	random permitting context grammar
RFCG	random forbidding context grammar
CFPG	context-free picture grammar
CSPG	context-sensitive picture grammar
RCPG	random context picture grammar
RPCPG	random permitting context picture grammar
RFCPG	random forbidding context picture grammar
BCPG	bag context picture grammar

In the case where the above abbreviations are in plural form, we add a lower case “s” at the end of the abbreviation, for example RCPGs.

Chapter 1

Introduction

1.1 General Introduction

Regulated rewriting in formal languages is a field that has been explored where there exists various grammar classes such as matrix grammars, random context grammars and bag context grammars to mention a few. Grammars with regulated rewriting are grammars with mechanisms to regulate the application of rules so that certain derivations are avoided. This study uses picture grammars that use regulated rewriting and the syntactic method of picture generation to generate pictures. Picture grammars are a class of grammars that generate pictures where a picture is defined as a pictorial form with all its labels belonging to the set of terminals while a pictorial form is any finite set of non overlapping labelled squares. A set of pictures generated by a grammar is then called that grammar's language or gallery. In the rest of this dissertation, picture languages will often be referred to as galleries in order for the study to be consistent.

The main objective of this study is to define bag context picture grammars (BCPGs) with reference to bag context tree grammars (BCTGs) and random context picture grammars (RCPGs). Because this is exploratory work, different examples of pictures generated by RCPGs and their three sub-classes are used to demonstrate how bag context picture grammars work. A lemma is also defined, that states that any RCPG and their sub-classes can be converted into equivalent BCPGs. The three sub-classes of RCPGs are namely context free picture grammars (CFPGs), random permitting context picture grammars (RPCPGs) and random forbidding context picture grammars (RFCPGs).

BCPGs are used to generate picture galleries that are similar to those of RCPGs. Similarity in the context of this study means that the pictures are generated by the same grammar where they have differences in the levels of refinement or that a certain structure such as a cross may be found in a particular region of one picture while it is found in a different region for another picture that belongs to the same gallery. The pictures are similar but differ in certain ways. Similar pictures are used in different ways, for example, they are used as picture passwords. In order to secure the password and prevent it from being stolen, similar pictures with small differences are used. This is a tactic used to confuse the perpetrator. Scan bar codes are also used to store information and if one examines these bar codes one will find that they are similar with slight differences. In this dissertation similar pictures are generated to demonstrate how bag context allows a certain level of control in picture generation.

As an example, we can use BCPGs in various ways in life. For example we can use them in computer networks. When dealing with computer networks, sometimes the volume or demand to access the networks becomes so high that it becomes essential to assess whether the network is being accessed by a human or machine. In order to determine this, experts have devised a method that checks this by asking the user or machine if they are a robot or not. To determine this a few similar pictures and those that are different will appear on the screen and the user or machine is asked to choose the pictures that are similar. We could use BCPGs to automatically generate similar pictures in this method. We then ask the users to select similar pictures in order to verify that they are in fact a human.

The definition of BCPGs is derived from the definitions of BCTGs and RCPGs. In Section 1.2, BCTGs are defined to give context on how bag context works. Section 1.3 gives an in-depth description of RCPGs. These topics relate to this study in that the definition of BCPGs was derived from them.

1.2 Bag Context Tree Grammars

A BCTG is a regular tree grammar in which the application of the rules is regulated by a vector of integers called the bag [5]. Formally, a bag is a device for regulated rewriting, and it is used to regulate rewriting in tree grammars. The bag evolves with the tree under construction [5]. In addition, a rule can be applied at any given stage if the bag at that stage is within the range defined as part of the rule. This range is specified by an upper and lower limit for the bag . If a rule is applied then not only the tree becomes affected but also the bag by means of a bag adjustment which is also part of the rule [5].

Under BCTGs we have the additive valence grammars and multiplicative valence grammars. Additive valence grammars are BCTGs with only one bag position, and with the upper and lower bounds for each rule being $+\infty$ and $-\infty$, respectively. Multiplicative valence grammars on the other hand generalize additive valence grammars in that the bag may have more than one bag position [5]. An example of a multiplicative valence grammar is given in Chapter 2. We also give the formal definition of BCTGs in Chapter 2. The next section is a discussion of RCPGs.

1.3 Random Context Picture Grammars

RCPGs are used to generate pictures through successive refinement. RCPGs are context free grammars with regulated rewriting. That is to say that a production is context free, but its application may depend on context randomly distributed in the developing picture [1]. Generating a picture with RCPGs involves dividing a square into equal labeled non-overlapping squares to form a pictorial form. The definition of a pictorial form and picture are given in Section 2.1.2. Picture generation with RCPGs begins with any geometric shape, in this case a square. At first the square (shape) contains the start symbol, then it is divided into non-overlapping squares (shapes) each containing a variable or terminal. A variable may be rewritten according to a context free production of the underlying grammars [1]. This can happen in two ways, either the shape containing it is divided into non-overlapping squares (shapes) or the variable is replaced by a variable or terminal. A production may depend on context randomly distributed in the developing picture. Context is classified as either permitting (P) or forbidding (F), where the rewriting in a derivation is regulated by insisting that a rule can only be applied to a non-terminal if every element of P and none of F occur in the rule [1] [5]. In order to visualize the picture every terminal is associated with a color and its shape is filled with the color [1].

The set of pictures generated by a RCPG is called the random context gallery [1]. RCPGs have various sub-classes but in this work, focus is given to three of these sub-classes namely: context free picture grammars (CFPGs), random permitting context picture grammars (RPCPGs) and random forbidding context picture grammars (RFCPGs) [6]. It follows then that the galleries generated by these grammars are the context free galleries, random permitting context galleries and random forbidding context galleries. In Chapter 2 we will look at the formal definition of RCPGs and their sub-classes. We also give an example of the Sierpiński Carpet.

1.4 Roadmap

This paper is divided into six chapters. Chapter 1 is the introduction, where we give a brief discussion of grammar types that use regulated rewriting. We have a brief discussion of RCPGs and BCTGs as we derive the definition of BCPGs from these two grammar types. Chapter 2 is the literature review where we look at related work that has been conducted in this field, we also look at notation that is used throughout the paper and some examples of BCTGs and RCPGs. In Chapter 3 we introduce the hypothesis and methodology used in this paper to get to our results. Chapter 4 focuses on defining BCPGs and the lemma that converts RCPGs into equivalent BCPGs, hence giving the relationship between BCPGs and RCPGs. We also give an example of how a picture is generated using BCPGs. Chapter 5 is dedicated to showing the galleries of pictures we can generate using BCPGs. And lastly Chapter 6 concludes all the work in this paper and includes future work that can be conducted in this field.

Chapter 2

Literature Review

An overview of this research was given in the previous chapter. We now look at the work that has been conducted in terms of generating pictures using different classes of picture grammars. We discuss how these picture grammars have been used to generate perfect and similar pictures. In this study the phrase " perfect pictures " refers to a gallery where the pictures look exactly the same. As an example, if there is a grammar that generates pictures of fractals, then the fractals will have the same level of refinement and they will look exactly the same. The sides will be the same length and the angles will also be the same size. Alternatively, similarity refers to a gallery of pictures where the pictures have different levels of refinement or the structure is different. The structure may be different in that, if there exists a gallery of pictures with exactly one cross, one picture may have a cross in one region while another picture has a cross in a different region to that of the first picture. The two pictures are similar as they both have a cross but different in that the cross is not found in the same region for both pictures belonging to the same gallery. The work that has already been carried out in this field will be related to the work covered in this research project.

2.1 Definitions and Notation

2.1.1 Notation

In this section definitions of the terms that will be used throughout the literature review and the rest of the paper are given. We first start with some basic terms that appear in the different definitions.

- The symbol $\mathbb{Z}$ refers to the set of integers.
- The symbol $\mathbb{N}$ refers to the set of natural numbers.
- The symbol $\mathbb{R}$ refers to the set of real numbers.

A signature is a set Σ of ranked symbols f^k (with $k \geqslant 0$ as the rank) [5], where a ranked alphabet or signature is a couple (F, arity) where F is a finite set and arity is a mapping from F to $\mathbb{N}$. In short a ranked alphabet or signature is an alphabet or signature whose symbols have an associated arity, where arity of a function or operation is the number of arguments or operands the function or operation accepts [7]. T_Σ is the smallest set such that $a \in T_\Sigma$ for all $a^0 \in \Sigma$ and $f[t_1, \ldots, t_k] \in T_\Sigma$ for all $f^k \in \Sigma$ $(k \geqslant 1)$ and $t_1, \ldots, t_k \in T_\Sigma$ [5]. Moreover, $T_{\Sigma \cup N}$ is the class of trees over $\Sigma \cup N$ where N is the set of non-terminals.

The set $\mathbb{Z} \cup \{-\infty, \infty\}$ will be denoted as $\mathbb{Z}_\infty$ [5]. $\mathbb{Z}^I$ is the set of integers with *rank* I. If $I = \{1, \ldots, k\}$ the elements of $\mathbb{Z}^I_\infty$ will be written as k-tuples. An element q of $\mathbb{Z}_\infty$ which occurs in the place of a vector simply denotes the vector of the appropriate size with all components equal to q [5].

2.1.2 Definitions

Bag Context Tree Grammars

In order to define what BCPGs are, it is essential to first define BCTGs. We use the definition of BCTGs given in [5]. BCTGs are defined as follows:

Definition 2.1.1. Bag Context Tree Grammars

A bag context tree grammar is a sex-tuple $G = (N, \Sigma, R, S, I, \beta_0)$ where we have,

- A finite signature N of non-terminals of *rank* 0.
- A finite signature Σ of terminals with $N \cap \Sigma = \emptyset$.
- A finite set R of rules of derivation.
- An initial non-terminal $S \in N$.
- A finite bag index set I; and
- A vector $\beta_0 \in \mathbb{Z}^I$, the initial bag.

A rule in R has the generic form $A \to t(\lambda, \mu; \alpha)$ where $A \in N$, $t \in T_{(\Sigma \cup N)}$, $\lambda, \mu \in \mathbb{Z}^I_\infty$ are the lower and upper limits respectively and $\alpha \in \mathbb{Z}^I$ is the bag adjustment.

In order to demonstrate how BCTGs work, an example adopted from [5] is given. This example shows how to complete a set of all binary trees with BCTGs.

Example 2.1.1. Bag context tree grammars

$G = (\{S, A, B\}, \{f^2, a^0\},\ R,\ S, \{1, 2, 3\}, (1, 0, 0))$, where R is the set of rules in Figure 2.1.

We have the set of non-terminals N. In the beginning of every round (a round starts when we apply the rules to add a level to the tree), the first bag position contains the number of instances of the non-terminal S in the current intermediate tree. The S produces two branches with As at the nodes. Rule 2.1 adds another level to the tree,

$$S \rightarrow f[A, A]\ ((1,0,0),(\infty,\infty,0);(-1,2,0)) \tag{2.1}$$

$$A \rightarrow B\ ((0,1,0),(0,\infty,\infty);(0,-1,1)) \tag{2.2}$$

$$B \rightarrow S\ ((0,0,1),(\infty,0,\infty);(1,0,-1)) \tag{2.3}$$

$$S \rightarrow a\ ((1,-\infty,-\infty),(\infty,0,0);(-1,-1,-1)) \tag{2.4}$$

Figure 2.1: Bag context tree grammar rules for binary trees

while counting the number of As in the second bag position. Rule 2.2 then, turns an A into a B. So now we have Bs at the nodes. Rule 2.3 turns these Bs into Ss again. Because all the Bs are turned into Ss this means another round can start and another level is added to the tree. Rule 2.4 is applicable only at the start of a round, which means the application is terminated at the last rule.

Next, the definitions of RCPGs and three of their sub-classes are given.

Random Context Picture Grammars

Pictures are generated using productions of the form in Figure 2.2, where A is a variable, $m = 1, 2, \ldots,$ $x_{11}, x_{12}, \ldots, x_{mm}$ are variables or terminals and P and F are sets of variables. The interpretation is as follows: if a developing picture contains a square labeled A and if all variables of P and none of F appear as labels of squares in the picture, then the square labeled A may be divided into equal squares with labels $x_{11}, x_{12}, \ldots, x_{mm}$ [1]. In order to cast the formulation into a more linear form, we denote the square with sides parallel to the axes where the lower lefthand corner is at (s, t) and the upper righthand corner at (u, v) by $((s, t), (u, v))$ using lowercase Greek letters for such constructs. Thus (A, θ) denotes a square θ labeled A. Furthermore, if θ is such a square $\theta_{11}, \theta_{12}, \ldots, \theta_{mm}$ denote the equal sub-squares into which θ can be divided, with θ_{11} denoting the left bottom one [6].

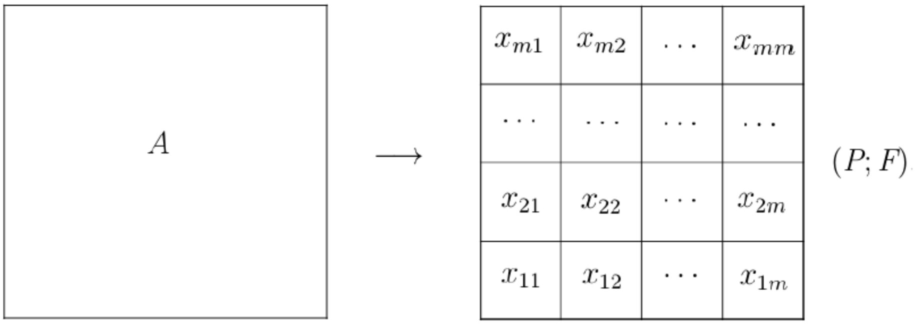


Figure 2.2: A production of a random context picture grammar [1]

According to [6], RCPGs combine the simplicity of context free grammars (CFGs) with the power of context sensitive grammars (CSGs). From [1] we obtain the more formal definition of RCPGs as follows:

Definition 2.1.2. Random Context Picture Grammars

A random context picture grammar is a four-tuple $G = (N,\ T,\ R,\ (S,\theta))$ where we have

- A finite set N of variables.
- A finite set T of terminals where $N \cap T = \emptyset$.
- R is a finite set of rules (productions) of the form $A \rightarrow [x_{11}, x_{12}, \ldots, x_{mm}](P; F)$, $m = 1, 2, 3, \ldots$ where $A \in N$, $x_{11}, x_{12}, ... \in (N \cup T)$ and $P,\ F \subseteq N$.
- An initial labelled square (S,θ) with $S \in N$.

The sub-classes of RCPGs are defined as follows:
If every production in a grammar G has $P = F = \emptyset$, we call G a context free picture grammar (CFPG); if $F = \emptyset$ for every rule, G is a random permitting context picture grammar (RPCPG) and lastly when $P = \emptyset$, G is a random forbidding context picture grammar (RFCPG) [6].

A *pictorial form* is any finite set of non-overlapping labeled squares in the plane. If Π is a pictorial form, we denote by $\ell(\Pi)$ the set of labels used in Π. The size of a pictorial form Π is the number of squares contained in it, that is $|\Pi|$ [1] [6].

For a random context picture grammar G and pictorial forms Π and Γ we write the derivation step $\Pi \Rightarrow \Gamma$ if there is a production $A \rightarrow [x_{11}, x_{12}, \ldots, x_{mm}]\ (P; F)$ in G, Π contains a labeled square (A, α), $\ell(\Pi \backslash\ \{(A,\alpha)\}) \supseteq P$ and $\ell(\Pi \backslash\ \{(A,\alpha)\}) \cap F = \emptyset$, and $\Gamma = (\Pi \setminus \{(A,\ \alpha)\}) \cup \{(x_1,\ \alpha_{11}),\ (x_2,\ \alpha_{12}), \ldots,\ (x_m,\ \alpha_{mm})\}$. As usual, $\Rightarrow^*$ denotes the reflexive transitive closure $\Rightarrow$ [1].

A picture is a pictorial form Π with $\ell(\Pi) \subseteq T$. The *gallery* $\mathcal{G}(G)$ *generated by a grammar* $G = (N,\ T,\ R,\ (S,\ \theta))$ is the set of pictures Π such that $\{(S,\ \theta)\} \Rightarrow^* \Pi$. We call the gallery generated by a random context picture grammar a random context gallery (rcpl) [1].

Example 2.1.2. Consider the pictorial form Π in Figure 2.3 with the set of labeled squares $\ell(\Pi) = \{M\}$, then $\Pi = \{(M, ((0,0), (\frac{1}{2}, \frac{1}{2}))),\ (M, ((\frac{1}{2}, 0), (1, \frac{1}{2})))$,-$(M, ((0, \frac{1}{2}),\ (\frac{1}{2}, 1))),\ (M, ((\frac{1}{2}, \frac{1}{2}), (1, 1)))\}$.

Now that we have defined RCPGs, we give an example of the Sierpiński Carpet which is generated using RCPGs. This example is adopted from [1].

Example 2.1.3. Sierpiński Carpet

The pictures in Figure 2.5 can be generated with an RCPG as follows: $G = (\{S, T, U, F\}, \{w, b\}, R, (S, \theta))$, where R is the set of rules in Figure 2.4.

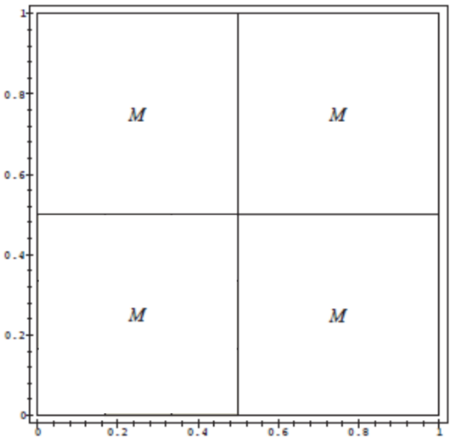


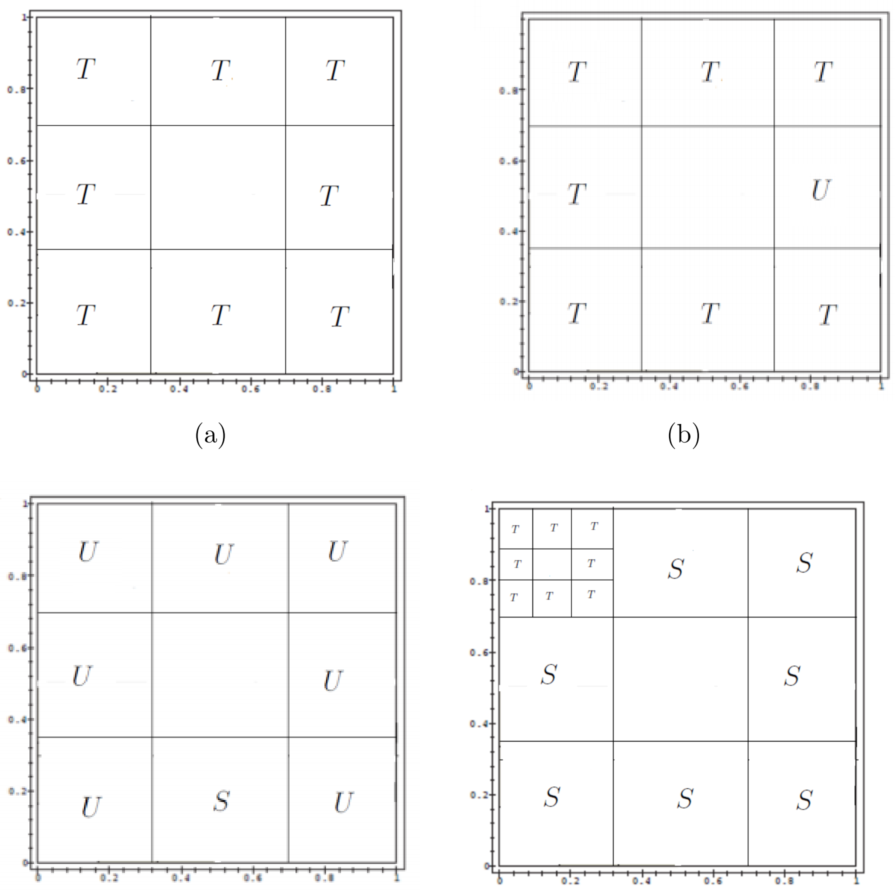
Figure 2.3: Pictorial form Π

$$
\begin{aligned}
S &\rightarrow [T,\ T,\ T,\ T,\ w,\ T,\ T,\ T,\ T]\ (\emptyset; \{U\}) && (2.5)\\
T &\rightarrow U\ (\emptyset; \{S,\ F\})\ | && (2.6)\\
&\quad\ F\ (\emptyset; \{S,\ U,\ F\})\ | && (2.7)\\
&\quad\ b\ (\{F\}; \emptyset) && (2.8)\\
U &\rightarrow S\ (\emptyset; \{T\}) && (2.9)\\
F &\rightarrow b\ (\emptyset\ ;\ \{T\}) && (2.10)
\end{aligned}
$$

Figure 2.4: Random context rules that generate the Sierpiński Carpet

We represent the colours light and dark with the terminals "w" and "b", respectively. The initial square labeled S is divided into nine equal big squares, of which the middle square is labeled w and the other squares are labeled T in Rule 2.5. This is shown in the pictorial form in Figure 2.5 (a). This pictorial form can now derive a picture or a more refined pictorial form. The decision is made by any T. If a T decides to terminate, it produces an F in Rule 2.7. The other Ts on sensing the F, each produce a b by Rule 2.8. Once there are no Ts left in the pictorial form, F also produces a b in Rule 2.10. Alternatively, T produces a U as shown in Figure 2.5 (b), and all other Ts follow suit in Rule 2.6 [1]. Then Rule 2.9 is applied and a U produces an S, all the other Us follow suit and we end up with 8 Ss and the process repeats. Figure 2.5 (c) shows the pictorial form after Rule 2.9 is applied and Figure 2.5 (d) shows the pictorial form after Rule 2.5 is applied. Some pictures in the gallery are shown in Figure 2.6.

We have defined RCPGs and given an example of a gallery generated by RCPGs. Next we look at Lindenmayer systems.



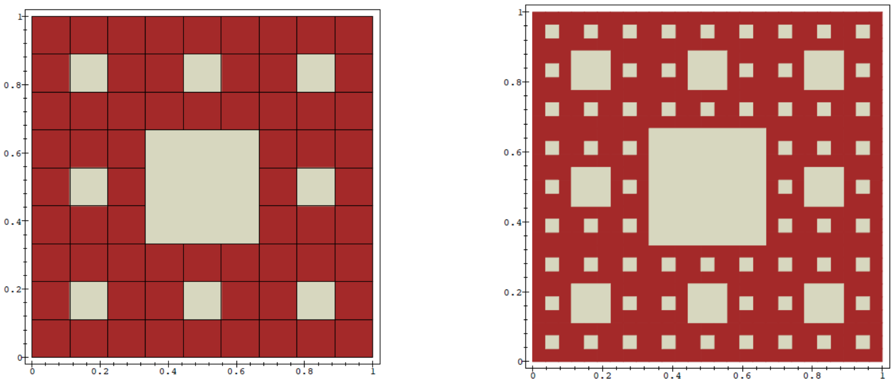
(a)

(b)

(c)

(d)

Figure 2.5: Pictorial forms that show how the picture develops during the application of rules in order to create the Sierpiński Carpet



(a)

(b)

Figure 2.6: Sierpiński Carpet where Figure (a) is in first level refinement and Figure (b) is in the second refinement [1]

Lindenmayer Systems

The Lindenmayer systems (L systems in short) are described to be a mathematical formalism proposed by Aristid Lindenmayer, a biologist, in 1968. The two principal applications of L systems include the generation of fractals and realistic modeling of plants [8]. There are many types of L systems. For example, there are context free, context sensitive and stochastic L systems [4]. Central to L systems is the notion of rewriting where the idea is to define complex structures such as the Kolam patterns by successively replacing parts of a simple object using a set of rewriting rules or productions. The rewriting can be carried out recursively [8]. The Kolam designs are geometric patterns formed by means of interleaving straight or curved lines. Women in villages in the south of India used these Kolam designs to decorate the courtyard in front of their homes. The art of Kolam is of ancient origin and is estimated to have developed as early as 5000 years ago [9]. According to [8], L systems are defined as follows:

Definition 2.1.3. Lindenmayer Systems

A Lindenmayer system is a three-tuple $G = (V, P, \omega)$ where

- Alphabet: the alphabet is a finite set V of formal symbols, usually taken to be letters a, b, c or any other characters.
- Axiom: also called the initiator is a string ω of symbols from V.
- Productions: also called rewriting rules is a mapping of a symbol a $\in V$ to a word $\omega \in V^*$, with notation $p : a \to \omega$.

We then give a more detailed description of the context free L systems as described by [4]:
A context free L system is formally defined as a four-tuple $G = (S, P, d, \alpha)$ where:

- S is the starting symbol also called the axiom.
- P is the set of productions (rules).
- d is the unit length.
- α is the unit angle.

In the L systems we introduce the concept of the turtle. The definition of the turtle is given by [8] as follows:

The state of a turtle is defined as a triplet (x, y, α) where the Cartesian coordinate (x, y) represents the turtle's position and the angle α, is interpreted as the direction in which the turtle is facing. Given the step size d and the angle α, the turtle can respond to the commands represented by the following symbols:

- F: move forward a step of length d. The state of the turtle changes to (x', y', α). A line segment is drawn between the points (x, y) and (x', y').

- f: move forward a step of length d without drawing a line. The state of the turtle changes as above.
- $+$: Turn right by angle δ. The next state of the turtle is $(x, y, \alpha + \delta)$.
- $-$: Turn left by angle δ. The next state of the turtle is $(x, y, \alpha - \delta)$.

Fuethermore still under context free L systems, we now introduce two symbol interpretations by the turtle: [9]

- [Push the current state of the turtle's position and orientation as well as other attributes such as color and width of lines being drawn.

-] Pop a state from the stack and make it the current state of the turtle. No line is drawn although, in general the position of the turtle changes.

We then give examples of L systems below.

Example 2.1.4. Fractal Generation

We present the L system in Figure 2.7 that generates a Koch snowflake taken from [10]. The axiom $F(1) - (120)F(1) - (120)F(1)$ draws an equilateral triangle, with edges of unit length. The parameter s determines the length of the line segment. Production p_1 replaces each line segment with a polygonal shape. Figure 2.7 produces the snowflake fractal that is in Figure 2.8.

$$\begin{aligned} w &: F(1) - (120)F(1) - (120)F(1) \\ p_1 &: F(s) \to F(s \setminus 3) + 60F(s \setminus 3) - (120)F(s \setminus 3) + 60F(s \setminus 3) \end{aligned}$$

Figure 2.7: L system rules that generates the fractal in Figure 2.8

In this chapter reference is made to Kolam patterns and an example is given to demonstrate how to generate Kolam patterns using L systems. According to [9], Kolam patterns can be generated by a variant of array grammars but, L systems with turtle interpretation offer a simpler and more intuitive method for generating these patterns. We give an example of the anklets of Krishna pattern generated using the L systems also adopted from [9].

Example 2.1.5. The Anklets of Krishna

The angle of increment α is 45°and the unit distance d is 5. The picture in Figure 2.10 is produced by the system in Figure 2.9. The rule w draws the inserted polygon which is the little curls and curves in the pattern. The p rule moves the turtle in 45° angles and connects the polygons at this angle [9].

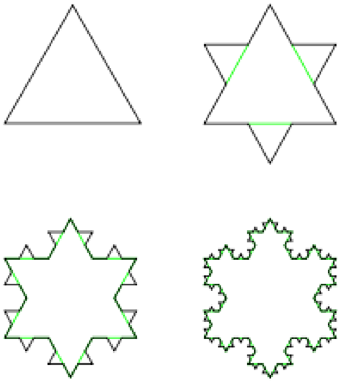


Figure 2.8: The first to fourth iterations of the Koch snowflake [2]

$$\begin{aligned} w &: f+Xf+f+Xf \\ p &: X \rightarrow XfX--XfX \end{aligned}$$

Figure 2.9: Grammar that produces the Kolam pattern in Figure 2.10

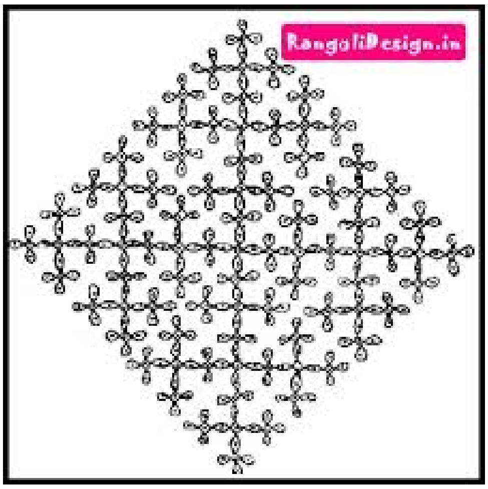


Figure 2.10: Anklets of Krishna [3]

Array Picture Grammars

An array grammar is a two dimensional generation of strings [11]. Among different array grammars for generating pictures of digitized rectangular arrays, the two dimensional (2d) array grammars create an elegant model with two phases of generation, one horizontal phase and the other a vertical phase [12]. The way these arrays work is that: the horizontal phase uses a sequential rewriting of a string grammar, generating strings of intermediate symbols. Each intermediate symbol serves as a start symbol of the vertical phase and generation proceeds in parallel in the vertical direction using right-linear non-terminal rules [12]. We give the definition of two dimensional array grammars which is adopted from the work of [12].

Definition 2.1.4. Array Picture Grammars

A two dimensional array grammar is a two-tuple $G = (G_1, G_2)$ where $G_1 = (H_1, I_1, P_1, S)$ is a regular grammar or a context free grammar.

- H_1 is a finite set of horizontal non-terminals.
- $I_1 = \{S_1, S_2, \ldots, S_k\}$ is a finite set of intermediates with $H_1 \cap I_1 = \emptyset$.
- P_1 is a finite set of rules called horizontal rules.
- S is the start symbol, $S \in H_1$.

$G_2 = (G_{21}, G_{22}, \ldots, G_{2k})$ where $G_{2i} = (V_{2i},\ T,\ P_{2i},\ S_i),\ 1 \leq i \leq k$ are regular or context free 2d array grammars.

- V_{2i} is a finite set of vertical non-terminals, $V_{2i} \cap V_{2j} = i$ where $i \neq j$.
- T is a finite set of terminals.
- P_{2i} is a finite set of right linear rules of the form $X \to aY$ or $X \to a$ where $X, Y \in V_{2i}$, $a \in T$.
- $S_i \in V_{2i}$ is the start symbol of G_{2i}.

Then we say that G is a regular or context free 2d array grammar if G_1 is a regular or context free array grammar [12].

The derivations are defined in the following manner: at first we use horizontal rules to generate a string over I_1 horizontally. Then the rectangular arrays are generated over T vertically and in parallel using the vertical phase non-terminal rules of G_{2i}. The vertical derivation terminates using the terminal rules of the vertical phase [12]. We give an example below from [12].

Example 2.1.6. Generating an H Token Figure

We have an array grammar $G = (G_1, G_2)$ where G_1 and G_2 are as explained above. $T = \{S, V, Z\}$ is the set of non-terminals of the horizontal phase; S_1 and S_2 are the

$$\begin{array}{rcl}
S & \rightarrow & S_2ZC \\
Z & \rightarrow & S_1Z \mid \\
 & \rightarrow & S_1V \\
V & \rightarrow & S_2C \\
S_1 & \rightarrow & xDA \\
C & \rightarrow & C \bullet A \mid \\
 & \rightarrow & \bullet \\
D & \rightarrow & \bullet \mid \\
 & \rightarrow & \bullet D \\
S_2 & \rightarrow & xB \\
A & \rightarrow & x \\
B & \rightarrow & xB \mid \\
 & \rightarrow & x
\end{array}$$

Figure 2.11: Rules for H token figure

$$\begin{array}{cccccc}
\mathrm{x} & \bullet & \bullet & \bullet & \bullet & \mathrm{x} \\
\mathrm{x} & \bullet & \bullet & \bullet & \bullet & \mathrm{x} \\
\mathrm{x} & \bullet & \bullet & \bullet & \bullet & \mathrm{x} \\
\mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} & \\
\mathrm{x} & \bullet & \bullet & \bullet & \bullet & \mathrm{x} \\
\mathrm{x} & \bullet & \bullet & \bullet & \bullet & \mathrm{x}
\end{array}$$

Figure 2.12: H pattern generated by the rules in Figure 2.11

intermediate symbols and $\{\bullet, x\}$ are terminals of the vertical phase. This grammar generates H type pattern (not necessarily with an equal number of rows above and below the middle line of x's) [12]. The set of rules R is found in Figure 2.11.

We have defined grammar classes that will be discussed in this dissertation. The next section will focus on how these different grammars generate both perfect and similar pictures. From this discussion we will show how we can use bag context picture grammars to generate similar pictures using examples of galleries generated by random context picture grammars.

2.2 Discussion and Analysis

One of the main aims of this paper is to define BCPGs. Since this work is exploratory, we will then use BCPGs to demonstrate how we generate similar pictures. In this section, we discuss the work that has been conducted in the different picture grammar classes that have been discussed in the previous section. We will look at picture grammars that generate perfect pictures. We also look at picture grammars that we can manipulate to generate similar pictures. These are pictures that might look the same but have differences such as color or shape.

2.2.1 Generating Perfect Images

There are classes of picture grammars that generate perfect pictures such as array grammars, L systems, radial grammars and radial L systems. Recall the definition of perfect pictures from introduction of this chapter that perfect pictures refers to a grammar that generates pictures that look exactly the same. L systems have been used by various authors to generate perfect pictures such as the Kolam patterns. Authors like *Prusinkiewics et. al* [10], have used L systems to generate pictures such as trees and fractals. In the work of *Oberguggernberger et. al* [8], L systems are used to generate pictures such as the Koch snowflake. For example, looking at the snowflake fractal example given earlier, it is perfect with symmetry and all the sides and angles are the same. Furthermore, grammar classes that generate perfect pictures cannot easily create variations in the pictures without changing the grammar itself. If we wanted to create variations with the Koch snowflake generated earlier, we would have to change the L system's rules. In this dissertation we demonstrate how BCPGs can perform this function without having to change the grammar rules.

Another picture grammar class that generates perfect symmetrical pictures are the radial grammars. These grammars work in the same manner as 2d array grammars. At first a horizontal line of intermediate cells is generated by a phrase structure, context sensitive, context free or regular grammar. Then starting with each intermediate cell a regular language is generated downward in a parallel manner from each cell in the horizontal line [13]. Organisms with bilateral symmetry can be described by these models, which are closed under operations such as translation, reflection and half turn [13]. These grammars generate pictures such as the tortoise shell Kolam. Moreover, it is shown in [12] that we can use these grammars to generate pictures such as the "H" and "I" figures using dots and crosses. An example of how this can be done is given the previous section.

RCPGs are used to generate pictures such as the Sierpiński carpet as shown in Example 2.1.3. Context free L systems can generate the Sierpiński triangles as well. In this dissertation we want to explore the possibility of generating the Sierpiński carpet or triangle with BCPGs. We want to generate these images such that, for example with the Sierpiński carpet, we show how we change the sizes of the squares using BCPGs. In terms of similarity, this dissertation looks at creating pictures that are similar with slight differences. The next section looks at grammar classes which generate similar pictures.

2.2.2 Generating Similar Images

In this section we discuss grammar classes that generate pictures that are similar such as the stochastic L systems. We look at how the different authors have used these grammars to generate similar pictures as this is one of the aims of this research project. We first take

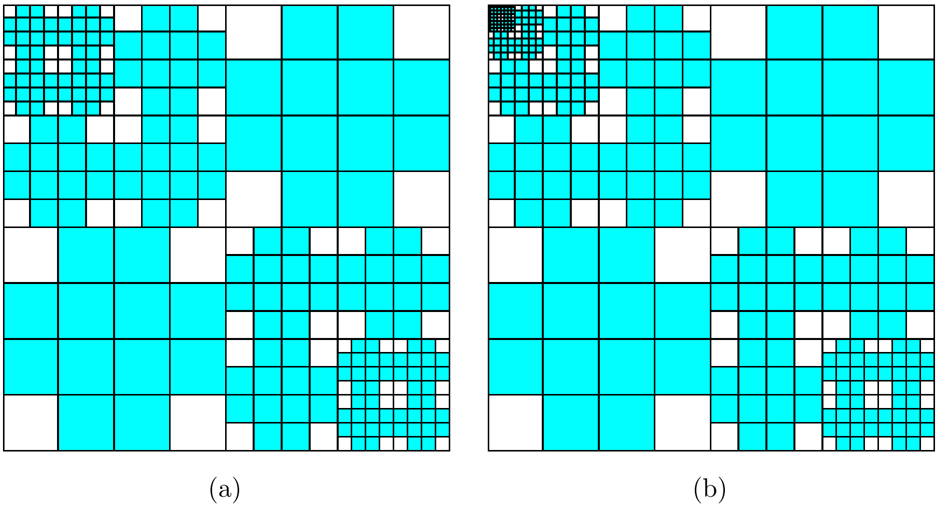


Figure 2.13: Similar pictures generated of RPCPGs

a look at RPCPGs and how we can use them to generate similar pictures. We then take a look at stochastic L systems and how they use probabilities to generate similar pictures.

Earlier in this work the definition of RCPGs was given and it was stated that RPCPGs are a subclass of RCPGs. In the work conducted by [1], an example of a set of pictures generated by RPCPGs was given. In the gallery, we have a cross in the upper right and lower left quarters. The upper left quarter is divided into four and a cross placed in each quarter of the picture. The lower right quarter of the picture must mirror the subdivision of the upper left [1]. With a picture like this, we could generate a similar picture in that we can have one quarter of the picture more refined than the other as shown in Figure 2.13. This creates two images that are similar. We could also use the same image but use different colors for the crosses to those used in the original image and thus generate a similar picture to the original one. We use BCPGs to do this as we will see in Chapter 5.

We now focus on the stochastic L systems (SLSs). These are used to model and generate more realistic or natural looking plants by using probability weights with the different rules. SLSs are almost identical to the L systems previously discussed. They differ in that multiple productions can exist for a single letter; the production applied is determined by a predefined probability [4]. For example, we generate an image of a tree where we give one production the probability of 50 percent, the next production 40 percent and the last production 10 percent. Since different productions may be used at different times, it is possible to generate many different variations of the plant and as a result create a gallery of similar pictures. This allows for the creation of plant classes all sharing similar overall characteristics, but differing in minor details [4]. Figure 2.15 shows a gallery generated by a SLS grammar with its rules shown in Figure 2.14.

As we have already stated, the aim of this project is to use BCPGs to generate similar pictures. We use the property of the bag to control the number of times one rule may be applied. In this manner we end up with similar images generated from the same productions. In the next section we conclude this chapter.

$$\begin{array}{rcl} & & Axiom F \\ & & Angle\ (\delta)\ 22.5^\circ \\ F^{.50} & \longrightarrow & F[+F][-F]F; \\ F^{.30} & \longrightarrow & F[-F]F; \\ F^{.20} & \longrightarrow & F[+F]F; \end{array}$$

Figure 2.14: Rules of Stochastic L system that generates trees in Figure 2.15

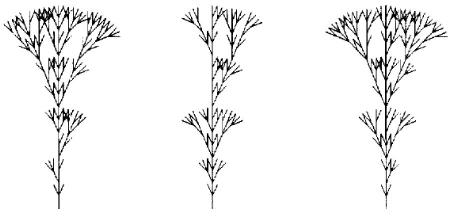


Figure 2.15: Pictures of similar trees generated by SLSs [4]

2.3 Chapter Summary

We discussed how existing picture grammars such as the Lindenmayer systems could be used to generate similar pictures. We also discussed how grammars such as the radial picture grammars generate perfect pictures. One of the main objectives of this research project is to define the bag context picture grammars. In the next chapter, the methodology used to conduct this research is shown given the background work that has already been conducted in this field. Since this work is exploratory, we select a set of pictures which we can generate using random context picture grammars and we generate similar pictures to those ones with bag context picture grammars.

Chapter 3

Research Methodology

3.1 Introduction

In the last chapter the literature related to this research was discussed. Picture grammars such as the array picture grammars, L systems and RCPGs were discussed in context to the objective of this study. For example, [1] gives an example where they used RCPGs to generate the Sierpiński Carpet. Because this dissertation is explanatory, we wanted to see if we could use BCPGs to generate similar pictures. We can use CFPGs to achieve this but CFPGs do not give the control BCPGs give. For example because we can use the bag to apply particular rules and to control the number of times a particular rule can be applied, this gives us control in terms of generating similar pictures.

The motivation for this research project is to explore if we can use BCPGs to generate similar pictures. This research serves three primary functions. The first function is to define BCPGs and the second function is to define a lemma that was adapted from [5] and originally developed for BCTGs. We define a similar lemma for BCPGs and used this lemma to create an equivalent BCPG for every RCPG. Thirdly we explore if we can generate similar pictures using BCPGs. We take pictures that were generated using RCPGs, CFPGs, RPCPGs and RFCPGs and then regenerate these pictures with BCPGs.

The next section presents the research hypothesis, followed by the research methodology. We give lastly, the conclusion of this section.

3.2 Research Hypothesis

In the previous chapter, it was shown that there exist classes of picture grammars that generate perfect pictures and classes that generate similar pictures. It was explained in the previous chapter what is meant by similar pictures and perfect pictures. In this research project we look into defining a picture grammar that we use to generate similar pictures. This leads us to a formulation of the hypothesis for this research which is as follows:

1. It is possible to define BCPGs using BCTGs.

2. It is possible to define a lemma that can convert any RCPG into its equivalent BCPG.

3. It is possible to use BCPGs to generate similar images, where similarity is subjective for the purpose of the research.

3.3 Research Method

We have already discussed the aim of this research project. We now look at the steps that were taken in order to conduct this research.

1. BCPGs were defined using BCTGs.

2. A lemma that is used to convert random context picture grammars into equivalent bag context picture grammars was defined.

3. Pictures that have already been generated using other classes of picture grammars such as RCPGs, RPCPGs, RFCPGs and CFPGs were selected and then BCPGs were used to generate the same and similar images to those selected.

4. The pictures were generated using a program developed by Nuru Jingili, a PhD student at the University of the Witwatersrand in the School of Computer Science and Applied Mathematics. The program was developed in Python and is not yet published. When given a BCPG, the program generates pictures and saves them.

3.4 Chapter Summary

In this chapter we have presented the aim and research methodology of this research project. We have listed the hypothesis statements. Lastly, we presented the different steps required for the methodology and described how they would be carried out. The next chapters will focus on defining BCPGs, stating the lemma and generating pictures using BCPGs.

Chapter 4

Bag Context Picture Grammars

In this chapter we focus on the definition of BCPGs. We formulate this definition based on BCTGs whose definition was given in Chapter 2.

Informally we define BCPGs as picture grammars in which the application of the rules is regulated by a vector of integers called the *bag*, which evolves with the picture under construction. The application of a particular rule is possible at any given stage as long as the bag at that stage is within the range, where the range is specified by the lower and upper limits. If a rule is applied, then a bag adjustment has to be performed as this is also part of the rule. Below we give a more formal definition of BCPGs.

4.1 Definitions

We generate pictures using productions of the form in Figure 4.1 where A is a variable, $m = 1, 2, 3, \ldots,$ $x_{11}, x_{12}, \ldots, x_{mm}$ are variables or terminals, λ and μ are the lower and upper limits respectively and we have the bag adjustment α. If we have a developing picture and that picture contains a square labeled A, we can divide A into equal squares labeled $x_{11}, x_{12}, \ldots, x_{mm}$. We denote the square with the sides parallel to the axes, the lower left vertex at (s, t) and the upper right hand vertex at (u, v) as $((s, t), (u, v))$ using lower case Greek letters for such constructs. Thus (A, θ) denotes a square θ labeled A. In addition, if θ is such a square, then $\theta_{11}, \theta_{12}, \ldots, \theta_{mm}$ denote the equal squares into which θ can be divided, for example θ_{11} denotes the left bottom one and θ_{mm} the top right one [6].

Definition 4.1.1. A bag context picture grammar (BCPG) is a sex-tuple $G = (N,\ \Sigma,\ R,\ (S,\ \theta),\ I,\ \beta_0)$ where we have,

- A finite set N of non-terminals.
- A finite set Σ of terminals with $N \cap \Sigma = \emptyset$.
- A finite set R of rules of derivation.
- An initial labeled square $(S,\ \theta)$ where $S \in N$ and θ is the initial labeled square with coordinates $((0, 0), (1, 1))$.

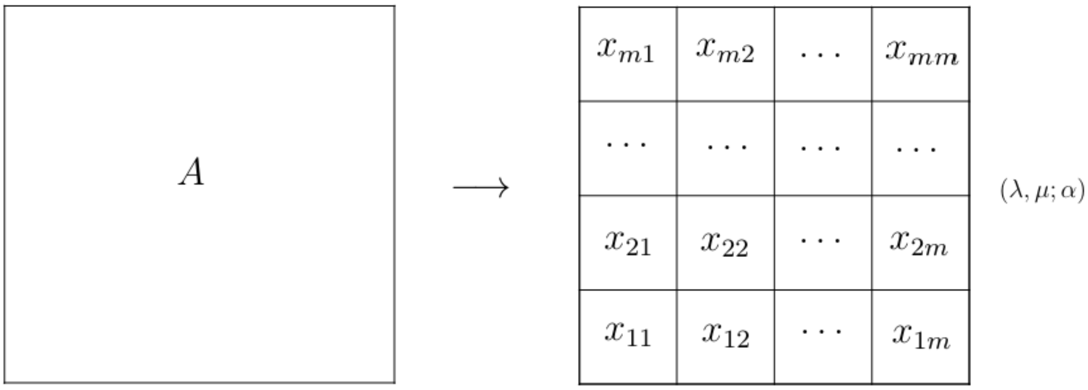


Figure 4.1: Production of a bag context picture grammar

- A finite bag index set I; and
- A vector $\beta_0 \in \mathbb{Z}^I$, the initial bag.

A rule in R has the generic form $A \rightarrow [x_{11},\ x_{12}, \ldots,\ x_{mm}]\ (\lambda, \mu; \alpha)$ where $m = 1, 2, 3, \ldots$, $A \in N$, $\{x_{11},\ x_{12}, \ldots,\ x_{mm}\} \subseteq N \cup \Sigma$, $\lambda,\ \mu \in \mathbb{Z}^I_\infty$ are the lower and upper limits respectively and $\alpha \in \mathbb{Z}^I$ is the bag adjustment.

A *pictorial form* is any finite set of non-overlapping labeled squares in the plane. If Π is a pictorial form, we denote by $\ell(\Pi)$ the set of labels used in Π. The size of a pictorial form Π is the number of squares contained in it, that is $|\Pi|$ [1] [6].

Let G be a BCPG. Let Π be a pictorial form such that any square $(A, \theta) \in \Pi$ while Ω is also a pictorial form such that $\Omega = \Pi \setminus \{(A,\ \theta))\} \cup \{(x_{11},\ \theta_{11}),\ (x_{12},\ \theta_{12}), \ldots, (x_{mm},\ \theta_{mm})\}$. Let $(\Pi,\ \beta)$ where Π and β are the pictorial form and bag at any stage of the picture under construction, then a derivation step from Π to Ω is defined as $(\Pi,\ \beta) \Rightarrow_G (\Omega,\ \beta^{'})$ if there is a production $A \rightarrow [x_{11},\ x_{12}, \ldots,\ x_{mm}]\ (\lambda,\ \mu;\ \alpha)$ in R with, $\lambda\ \leq\ \beta\ \leq\ \mu$ and $\beta^{'} = \beta +\ \alpha$. As usual $\Rightarrow^*_G$ denotes the reflexive transitive closure of $\Rightarrow_G$. If G is clear we will use $\Rightarrow$ and $\Rightarrow^*$ instead of $\Rightarrow_G$ and $\Rightarrow^*_G$.

A *picture* is a pictorial form Π with $\ell(\Pi) \subseteq \Sigma$. The *gallery* $\mathcal{G}(G)$ *generated by a grammar* $G = (N,\ \Sigma,\ R,\ (S,\ \theta),\ I,\ \beta_0)$ is the set of pictures Π such that $((S,\ \theta),\ \beta_0) \Rightarrow^*_G (\Pi,\ \beta),\ \beta \in \mathbb{Z}^I$, in other words $\mathcal{G}(G) = \{\Pi|\ ((S,\ \theta),\ \beta_0) \Rightarrow^*_G (\Pi,\ \beta),\ \beta \in \mathbb{Z}^I\}$. We call the gallery generated by a bag context picture grammar a *bag context gallery.*

We have given the definition of BCPGs and shown the required conditions needed for BCPGs to generate pictures. In the next section we discuss the lemma we use to convert RCPGs into equivalent BCPGs. We then look at a step by step example of how a picture is formed under BCPGs.

4.2 Relationship between BCPGs and RCPGs

In this section we rewrite RCPGs as BCPGs.

Notation

We introduce notation that will be useful in this section. Suppose we are interested in a (bag or random context) picture grammar generating pictures over Σ, using a set $N = \{A_1, \ldots, A_m\}$ of nonterminals (with the order of elements in N arbitrary but fixed). For the right hand side of a rule, say $A \rightarrow [x_{11}, x_{12}, \ldots, x_{mm}]$, we denote by $\mathrm{cnt}([x_{11}, x_{12}, \ldots, x_{mm}])$ the m-tuple $(n_1, \ldots, n_m)$ such that n_i is equal to the number of occurrences of A_i in $[x_{11}, x_{12}, \ldots, x_{mm}]$. In particular, for $A \in N$, $\mathrm{cnt}(A)$ is the m-tuple in which the component corresponding to A is set to 1, whereas all others are set to 0. Then $\mathrm{cnt}([x_{11}, x_{12}, \ldots, x_{mm}]) = \Sigma_{i=1}^{m} \Sigma_{j=1}^{m} \mathrm{cnt}(x_{ij})$.

The lemma discussed in this section is adapted from the lemma developed by [5] for BCTGs.

Lemma 4.2.1. For every random context picture grammar $G = (N,\ \Sigma,\ R,\ (S, \theta))$ there is a bag context picture grammar $G^{'} = (N,\ \Sigma,\ R^{'},\ (S, \theta), I,\ \mathrm{cnt}(S))$.

Proof. Assume without loss of generality that $N = \{A_1, \ldots, A_m\}$. We can then use the bag index set $I = \{1, \ldots, m\}$ to record, at all times during a derivation, the number of occurrences of A_i in the ith position of the bag. In this way, we obtain a bag context picture grammar $G^{'}$ generating the same language as G. In order to implement this, we use the initial bag $\mathrm{cnt}(S)$. For every random context rule $A \rightarrow [x_{11}, x_{12}, \ldots, x_{mm}]$ $(P;\ F)$, the bag context rule becomes $A \rightarrow [x_{11},\ x_{12}, \ldots, x_{mm}]$ $(\lambda,\ \mu;\ \alpha)$, where

- $\lambda = \mathrm{cnt}(A) + \mathrm{cnt}(P)$
- $\mu = \mathrm{cnt}(A) + \infty{\cdot}\mathrm{cnt}(N \backslash F)$ (where by convention $\infty{\cdot}0 = 0$) and
- $\alpha = \mathrm{cnt}([x_{11},\ x_{12}, \ldots,\ x_{mm}])$ - $\mathrm{cnt}(A)$.

Then $\mathcal{G}(G^{'}) = \mathcal{G}(G)$. □

4.3 BCPGs, RCPGs and Three Sub-Classes

In this section we use Lemma 4.2.1 to look at RCPGs, CFPGs, RPCPGs and RFCPGs and see how we can construct equivalent BCPGs.

4.3.1 Context Free Picture Grammars

Let G be a CFPG, then for every rule $P = F = \emptyset$. We calculate λ, μ and α using Lemma 4.2.1 as follows:

- $\lambda = \mathrm{cnt}(A) + \mathrm{cnt}(P) = \mathrm{cnt}(A) + 0 = \mathrm{cnt}(A)$
- $\mu = \mathrm{cnt}(A) + \infty{\cdot}\ \mathrm{cnt}(N \backslash F) = \mathrm{cnt}(A) + \infty{\cdot}\ \mathrm{cnt}(N) = \infty$

$$
\begin{aligned}
S &\longrightarrow [S, S, S, S] \mid && (4.1)\\
& [C_{lb}, C_{rb}, C_{lt}, C_{rt}] && (4.2)\\
C_{lb} &\longrightarrow [w, b, b, b] && (4.3)\\
C_{rb} &\longrightarrow [b, w, b, b] && (4.4)\\
C_{lt} &\longrightarrow [b, b, w, b] && (4.5)\\
C_{rt} &\longrightarrow [b, b, b, w] && (4.6)
\end{aligned}
$$

Figure 4.2: Context Free Picture Grammar rules for G_{Free}

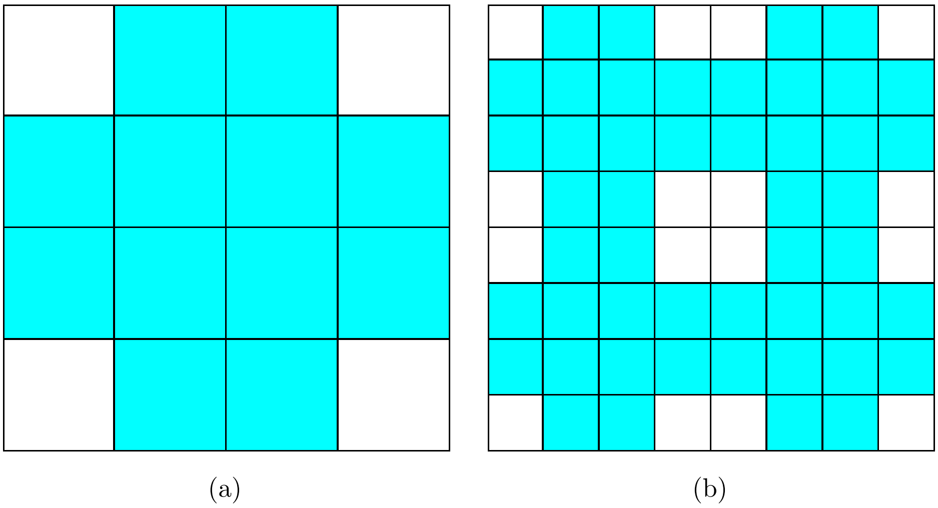

(a) (b)

Figure 4.3: Some of the pictures generated by G_{Free}

- α = cnt($[x_{11},\ x_{12}, \ldots,\ x_{mm}]$) - cnt($A$)

The following grammar is adapted from [1].

Example 4.3.1. We have the grammar $G_{\mathsf{Free}} = (\{S, C_{lb}, C_{rb}, C_{lt}, C_{rt}\}, \{w,\ b\}, R,\ (S, \theta))$ where R is the set of rules in Figure 4.2.

We use the grammar G_{Free} to generate pictures with dark crosses on a light background. The terminals "w" and "b" represent the light and dark colours respectively. We get crosses by dividing the initial square S into four squares (Rule 4.1). Each of these squares can be divided recursively or may be filled with a cross (Rules 4.2–4.6). A cross is generated by quartering a square in Rule 4.2 and then generating the required pattern of light and dark squares in each part (Rules 4.3–4.6) [1]. The gallery of pictures generated by this grammar is in Figure 4.3.

We use the formula given in Lemma 4.2.1 to calculate the lower limit, upper limit and the bag adjustment for each rule in Figure 4.2. We get the BCPG $G_{\mathsf{Bag\text{-}free}}$ $= (\{S, C_{lb}, C_{rb}, C_{lt}, C_{rt}\}, \{w, b\}, R, (S, \theta), \{1, 2, 3, 4, 5\}, (1, 0, 0, 0, 0))$ where R is the set of rules in Figure 4.4:

$$
\begin{array}{rcl}
S & \longrightarrow & [S,S,S,S]\ ((1,0,0,0,0),\infty;(3,0,0,0,0))\ | \\
 & & [C_{lb},C_{rb},C_{lt},C_{rt}]\ ((1,0,0,0,0),\infty;(-1,1,1,1,1)) \\
C_{lb} & \longrightarrow & [w,b,b,b]\ ((0,1,0,0,0),\infty;(0,-1,0,0,0)) \\
C_{rb} & \longrightarrow & [b,w,b,b]\ ((0,0,1,0,0),\infty;(0,0,-1,0,0)) \\
C_{lt} & \longrightarrow & [b,b,w,b]\ ((0,0,0,1,0),\infty;(0,0,0,-1,0)) \\
C_{rt} & \longrightarrow & [b,b,b,w]\ ((0,0,0,0,1),\infty;(0,0,0,0,-1))
\end{array}
$$

Figure 4.4: Bag Context Picture Grammar rules for $G_{\textsf{Bag-free}}$

When we apply Lemma 4.2.1 to the grammar with rules in Figure 4.2, we get the resultant grammar with rules in Figure 4.4.

4.3.2 Random Permitting Context Picture Grammars

Let G be a RPCPG. Then $F = \emptyset$ for every rule in G. We calculate λ, μ and α by Lemma 4.2.1 as follows:

- $\lambda = \text{cnt}(A) + \text{cnt}(P)$
- $\mu = \text{cnt}(A) + \infty\cdot\ \text{cnt}(N\backslash F) = \text{cnt}(A) + \infty \cdot \text{cnt}(N) = \text{cnt}(A) + \infty = \infty$
- $\alpha = \text{cnt}([x_{11},\ x_{12}, \ldots,\ x_{mm}])$ - $\text{cnt}(A)$

The following example was adapted from [1]. This grammar generates pictures that have dark crosses on a light background, where the lower left and upper right quarters of a picture each contains one cross. The upper left quarter may be divided into four and a cross placed in each quarter; this may be recursively repeated for the upper left quarter of the resulting sub-picture. The lower right quarter of the picture must mirror the subdivision of the upper left.

Example 4.3.2. $G_{\textsf{Permit}} = (\{S, A, B, C, A', B', F, C_{lb}, C_{rb}, C_{lt}, C_{rt}\}, \{w, b\}, R, (S, \theta))$ where R is the set of rules in Figure 4.5.

At the initial iteration of the main loop, Rules 4.7–4.13, the pictorial form has an A in the upper left corner and a B in the lower right corner. We can either have pictorial forms ϕ_i or ϕ_{i+1}. We get ϕ_i if the following rules are applied: 4.9, 4.11 and 4.14–4.19 twice each. The decision is made by A by producing an F in Rule 4.9, B produces an F on sensing the F in the pictorial form by Rule 4.11. An F can only generate a cross by Rules 4.14–4.19, which proceeds as described in Example 4.3.1.

Alternatively, a more refined pictorial form ϕ_{i+1} is obtained by applying Rules 4.8, 4.10, 4.12 and 4.13. This happens when the square containing A is quartered and the label A' placed in the upper left corner and a C in every other corner. B will then sense the A' in the pictorial form and mirror the subdivision of A which entails placing

$$\begin{aligned}
S &\longrightarrow [C, B, A, C] && (4.7)\\
A &\longrightarrow [C, C, A', C]\ (\{B\}; \emptyset)\ | && (4.8)\\
& \quad\ F\ (\{B\}; \emptyset) && (4.9)\\
B &\longrightarrow [C, B', C, C]\ (\{A'\}; \emptyset)\ | && (4.10)\\
& \quad\ F\ (\{F\}; \emptyset) && (4.11)\\
A' &\longrightarrow A\ (\{B'\}; \emptyset) && (4.12)\\
B' &\longrightarrow B\ (\{A\}; \emptyset) && (4.13)\\
F &\longrightarrow C && (4.14)\\
C &\longrightarrow [C_{lb}, C_{rb}, C_{lt}, C_{rt}] && (4.15)\\
C_{lb} &\longrightarrow [w, b, b, b] && (4.16)\\
C_{rb} &\longrightarrow [b, w, b, b] && (4.17)\\
C_{lt} &\longrightarrow [b, b, w, b] && (4.18)\\
C_{rt} &\longrightarrow [b, b, b, w] && (4.19)
\end{aligned}$$

Figure 4.5: Random Permitting Context Picture Grammar rules for $G_{\textsf{Permit}}$

a B in the lower right quarter and C's in the other three quarters by Rule 4.10. Once the subdivision of B has taken place, A can be restored by Rule 4.12, which in turn enables B to be restored (Rule 4.13). This completes the main loop. It could happen that this grammar can terminate without generating a picture, if Rule 4.14 is executed immediately after Rule 4.9. Rule 4.11 will never be enabled again. Some of the pictures generated by this grammar are shown in Figure 4.6.

We use Lemma 4.2.1 to calculate the limits and bag adjustment of each rule in Figure 4.5 and we have the equivalent bag context picture grammar as follows: $G_{\textsf{Bag-permit}}$ = $(\{S, A, B, C, A', B', F, C_{lb}, C_{rb}, C_{lt}, C_{rt}\}, \{w, b\}, R, (S, \theta)), \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$,- $(1, 0, \ldots, 0))$ where R is the set of rules in Figure 4.7.

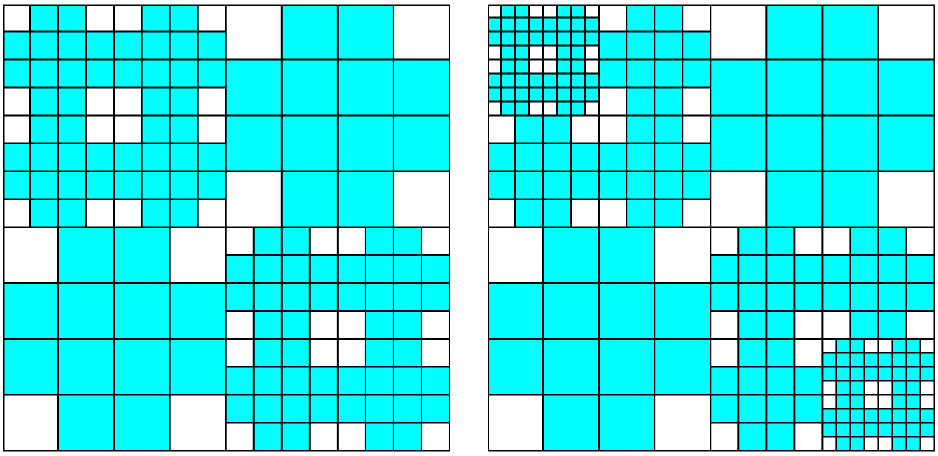
When we apply Lemma 4.2.1 to the grammar with rules in Figure 4.5, we get the resultant grammar with rules in Figure 4.7.

4.3.3 Random Forbidding Context Picture Grammars

Let G be a RFCPG. Then for every rule $P = \emptyset$. We calculate λ, μ and α using Lemma 4.2.1 as follows:

- λ = cnt(A) + cnt(P) = cnt(A) + 0 = cnt(A)
- μ = cnt(A) + $\infty\cdot$ cnt($N\backslash F$)
- α = cnt($[x_{11},\ x_{12}, \ldots,\ x_{mm}]$) - cnt($A$)

Again we want dark crosses on a light background. The grammar $G_{\textsf{Forbid}}$ and its description are adapted from [1]. We want a RFCPG where we have a picture with dark crosses



(a)

(b)

Figure 4.6: Gallery of pictures generated by $G_{\textsf{Permit}}$

$$
\begin{array}{rcl}
S & \longrightarrow & [C, B, A, C]\ ((1, 0, \ldots, 0), \infty; (0, 1, 1, 2, 0, 0, \ldots, 0)) \\
A & \longrightarrow & [C, C, A', C]\ ((0, 1, 1, 0, \ldots, 0), \infty; (0, -1, 0, 3, 1, 0, \ldots, 0))\ | \\
 & & F\ ((0, 1, 1, 0, \ldots, 0), \infty; (0, -1, 0, 0, 0, 0, 1, 0, 0, 0, 0)) \\
B & \longrightarrow & [C, B', C, C]\ ((0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0), \infty; (0, 0, -1, 3, 0, 1, 0, 0, 0, 0, 0))\ | \\
 & & F\ ((0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0), \infty; (0, 0, -1, 0, 0, 0, 1, 0, 0, 0, 0)) \\
A' & \longrightarrow & A\ ((0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0), \infty; (0, 1, 0, 0, -1, 0, 0, 0, 0, 0, 0)) \\
B' & \longrightarrow & B\ ((0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0), \infty; (0, 0, -1, 3, 0, 1, 0, 0, 0, 0, 0)) \\
F & \longrightarrow & C\ ((0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0), \infty; (0, 0, 0, -1, 0, 0, 1, 0, 0, 0, 0)) \\
C & \longrightarrow & [C_{lb}, C_{rb}, C_{lt}, C_{rt}]\ ((0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0), \infty; (0, 0, 0, -1, 0, 0, 0, 1, 1, 1, 1)) \\
C_{lb} & \longrightarrow & [w, b, b, b]\ ((0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0), \infty; (0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0)) \\
C_{rb} & \longrightarrow & [b, w, b, b]\ ((0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0), \infty; (0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0)) \\
C_{lt} & \longrightarrow & [b, b, w, b]\ ((0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0), \infty; (0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0)) \\
C_{rt} & \longrightarrow & [b, b, b, w]\ ((0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1), \infty; (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1))
\end{array}
$$

Figure 4.7: Bag Context Picture Grammar rules for $G_{\textsf{Bag-permit}}$

$$
\begin{aligned}
S &\longrightarrow [T,T,T,T]\ (\emptyset;\{U\}) && (4.20)\\
T &\longrightarrow U\ (\emptyset;\{S,A\})\ | && (4.21)\\
&\ [C,B,A,C]\ (\emptyset;\{S,U\}) && (4.22)\\
U &\longrightarrow S\ (\emptyset;\{T\}) && (4.23)\\
A &\longrightarrow [C,C,A',C]\ (\emptyset;\{T,B',F\})\ | && (4.24)\\
&\ F\ (\emptyset;\{T,B',A'\}) && (4.25)\\
B &\longrightarrow [C,B',C,C]\ (\emptyset;\{A,F\})\ | && (4.26)\\
&\ F\ (\emptyset;\{A,A'\}) && (4.27)\\
A' &\longrightarrow A\ (\emptyset;\{B\}) && (4.28)\\
B' &\longrightarrow B\ (\emptyset;\{A'\}) && (4.29)\\
F &\longrightarrow C\ (\emptyset;\{A,B\}) && (4.30)\\
C &\longrightarrow [C_{lb},C_{rb},C_{lt},C_{rt}] && (4.31)\\
C_{lb} &\longrightarrow [w,b,b,b] && (4.32)\\
C_{rb} &\longrightarrow [b,w,b,b] && (4.33)\\
C_{lt} &\longrightarrow [b,b,w,b] && (4.34)\\
C_{rt} &\longrightarrow [b,b,b,w] && (4.35)
\end{aligned}
$$

Figure 4.8: Random Forbidding Context Picture Grammar rules for G_{Forbid}

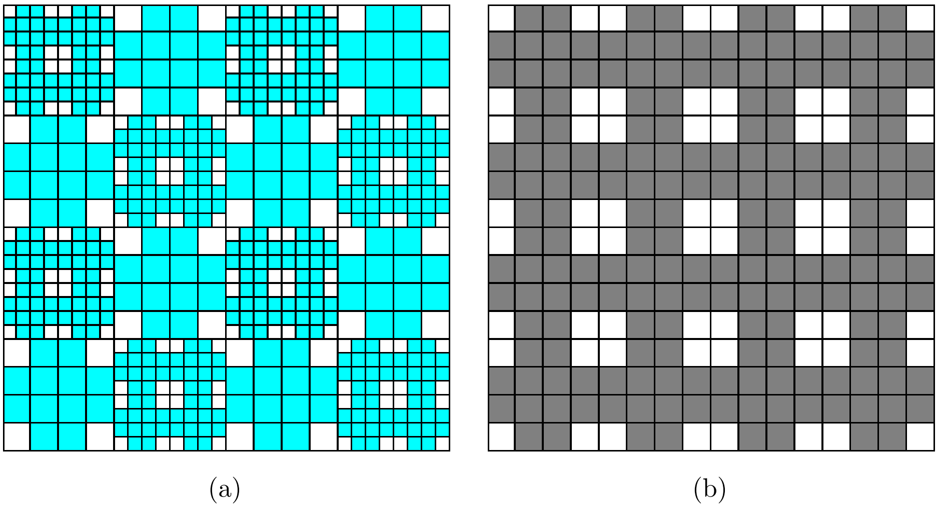
on a light background with the following structure: every picture has 4^i, where $i \geq 1$, identical sub-pictures, each of which is created as described in Example 4.3.2.

Example 4.3.3. $G_{\mathsf{Forbid}} = (\{S,T,U,A,B,C,A',B',F,C_{lb},C_{rb},C_{lt},C_{rt}\},\{w,b\},R,(S,\theta))$ where R is the set of rules in Figure 4.8.

When we start the main loop at the beginning of the ith iteration, with $i \geq 1$, that is, Rules 4.20 to 4.23, we have a pictorial form ϕ_i which is divided into 4^{i-1} equal squares labeled S. Each of these squares divides into four new squares which are labeled T by Rule 4.20. This pictorial form can derive the pictorial form ϕ_{i+1} by Rules 4.21 and 4.23 or a picture with 4^i identical sub-pictures by Rules 4.22, 4.24 to 4.35 where the decision is made by any T.

In the latter case, by Rule 4.22, each T starts to generate one of the 4^i identical sub-pictures by producing an A in the upper left quarter, a B in the lower right and C's in the other two. This pictorial form can lead to a picture or a more refined pictorial form. The decision is made by any A. In the first case A produces an F and this is copied by each A in the pictorial form by Rule 4.25, then by each B in Rule 4.27. An F can only generate a cross by Rules 4.30 to 4.35.

In the former case, the square containing A is quartered and then labeled A' which is placed in the upper left corner, and a C in every other corner; each A in the pictorial form copies this by Rule 4.24. Each B in the pictorial form, on sensing an A', mirrors



(a)

(b)

Figure 4.9: Some of the pictures generated by $G_{\textsf{Forbid}}$

the subdivision of the A's, which entails placing a B in the lower right quarter and C's in the other three (Rule 4.26). Once the subdivision of the B's has taken place, the A's can be restored by Rule 4.28, which in turn enables the B's to be restored by Rule 4.29. Thereby the main loop is completed. Some of the pictures generated by this grammar are shown in Figure 4.9.

We use the formula given in Lemma 4.2.1 to calculate the limits and bag adjustment of Figure 4.10 as follows: $G_{\textsf{Bag-forbid}} = (\{S, T, U, A, B, C, A', B', F, C_{lb}, C_{rb}, C_{lt}, C_{rt}\}, \{w, b\}, R, (S, \theta), \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10,\text{-} 11, 12, 13\}, (1, 0, \ldots, 0))$ where R is the set of rules in Figure 4.10.

When we apply Lemma 4.2.1 to the grammar with rules in Figure 4.8, we get the resultant grammar with rules in Figure 4.10.

4.3.4 Random Context Picture Grammars

Let G be a RCPG. Then for every rule we have sets P and F and we calculate λ, μ and α using Lemma 4.2.1 as follows:

- $\lambda = \text{cnt}(A) + \text{cnt}(P)$
- $\mu = \text{cnt}(A) + \infty\cdot \text{cnt}(N \backslash F)$
- $\alpha = \text{cnt}([x_{11},\ x_{12}, \ldots,\ x_{mm}])$ - $\text{cnt}(A)$

We want to generate the gallery of hollow isosceles right triangle with side 2^i where $i \geq 1$. We adapt this grammar from [1].

Example 4.3.4. We have the RCPG $G_{\textsf{Triangle}} = (\{S,\ T_{lb},\ T_{rb},\ T_t,\ T_{ls},\text{-} T_b,\ T_h,\ T_{lb}',\ T_{rb}',\ T_t',\ T_{ls}',\ T_b',\ T_h',\ F\}, \{w,\ b\}, R,\ (S, \theta))$ where R is the set of rules in Figure 4.11.

$$
\begin{array}{rcl}
S & \longrightarrow & [T,T,T,T]\ ((1,0,\ldots,0),(\infty,\infty,0,\infty,\ldots,\infty);(-1,4,0,\ldots,0)) \\
T & \longrightarrow & U\ ((0,1,0,\ldots,0),(0,\infty,\infty,0,\infty,\ldots,\infty);(0,-1,1,0,\ldots,0))\ | \\
 & & [C,B,A,C]\ ((0,1,\ldots,0),(0,\infty,0,\infty,\ldots,\infty); \\
 & & (0,-1,0,1,1,2,0,0,0,0,0,0,0)) \\
U & \longrightarrow & S\ ((0,0,1,\ldots,0),(\infty,0,\infty,\ldots,\infty);(1,0,-1,0,0,0,0,0,0,0,0,0,0)) \\
A & \longrightarrow & [C,C,A',C]\ ((0,0,0,1,0,\ldots,0),(\infty,0,\infty,\infty,\infty,\infty,\infty,0,0,\infty,\infty,\infty,\infty); \\
 & & (0,0,0,-1,0,3,1,0,0,0,0,0,0))\ | \\
 & & F\ ((0,0,0,1,0,\ldots,0),(\infty,0,\infty,\infty,\infty,\infty,0,0,\infty,\infty,\infty,\infty,\infty); \\
 & & (0,0,0,-1,0,0,0,0,1,0,0,0,0)) \\
B & \longrightarrow & [C,B',C,C]\ ((0,0,0,0,1,0,\ldots,0),(\infty,\ldots,\infty,0,\infty,\infty,\infty,\infty); \\
 & & (0,0,0,0,-1,3,0,1,0,0,0,0,0))\ | \\
 & & F\ ((0,0,0,0,1,0,\ldots,0),(\infty,\infty,\infty,0,\infty,\infty,0,\infty,\ldots,\infty); \\
 & & (0,0,0,0,-1,0,0,0,1,0,0,0,0)) \\
A' & \longrightarrow & A\ ((0,\ldots,0,1,0,\ldots,0),(\infty,\ldots,\infty,0,\infty,\ldots,\infty); \\
 & & (0,0,0,1,0,0,-1,0,0,0,0,0,0)) \\
B' & \longrightarrow & B\ ((0,\ldots,0,1,0,\ldots,0),(\infty,\ldots,\infty,0,\infty,\ldots,\infty); \\
 & & (0,0,0,0,1,0,0,-1,0,0,0,0,0)) \\
F & \longrightarrow & C\ ((0,\ldots,0,1,0,0,0,0),(\infty,\infty,\infty,0,0,\infty,\ldots,\infty); \\
 & & (0,0,0,0,0,1,0,0,-1,0,0,0,0)) \\
C & \longrightarrow & [C_{lb},C_{rb},C_{lt},C_{rt}]((0,0,0,0,0,1,0,\ldots,0),\infty; \\
 & & (0,0,0,0,-1,0,0,0,0,1,1,1,1)) \\
C_{lb} & \longrightarrow & [w,b,b,b]\ ((0,\ldots,0,1,0,0,0),\infty;(0,\ldots,0,-1,0,0,0)) \\
C_{rb} & \longrightarrow & [b,w,b,b]\ ((0,\ldots,0,1,0,0),\infty;(0,\ldots,0,-1,0,0)) \\
C_{lt} & \longrightarrow & [b,b,w,b]\ ((0,\ldots,0,1,0),\infty;(0,\ldots,0,-1,0)) \\
C_{rt} & \longrightarrow & [b,b,b,w]\ ((0,\ldots,0,1),\infty;(0,\ldots,0,-1))
\end{array}
$$

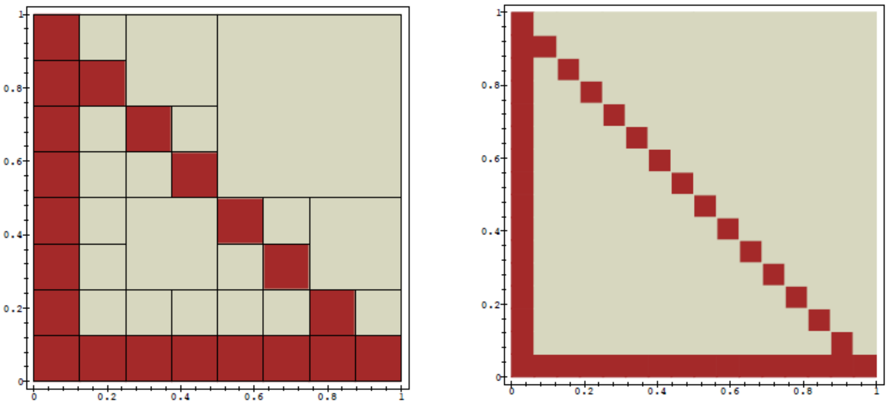
Figure 4.10: Bag Context Picture Grammar rules for $G_{\textsf{Bag-forbid}}$

$$\begin{array}{rcll}
S & \longrightarrow & [T_{lb}, T_{rb}, T_t, w] & (4.36) \\
T_{lb} & \longrightarrow & [{T_{lb}}', {T_b}', {T_{ls}}', w]\ (\emptyset; \{{T_{rb}}', {T_t}', {T_{ls}}', {T_b}', {T_h}'\}) \mid & (4.37) \\
 & & F\ (\emptyset; \{{T_{rb}}', {T_t}', {T_{ls}}', {T_b}', {T_h}'\}) & (4.38) \\
T_{rb} & \longrightarrow & [{T_b}', {T_{rb}}', {T_h}', w]\ (\{{T_{lb}}'\}; \emptyset) \mid & (4.39) \\
 & & b\ (\{F\}; \emptyset) & (4.40) \\
T_t & \longrightarrow & [{T_{ls}}', {T_h}', {T_t}', w]\ (\{{T_{lb}}'\}; \emptyset) \mid & (4.41) \\
 & & b\ (\{F\}; \emptyset) & (4.42) \\
T_{ls} & \longrightarrow & [{T_{ls}}', w, {T_{ls}}', w]\ (\{{T_{lb}}'\}; \emptyset) \mid & (4.43) \\
 & & b\ (\{F\}; \emptyset) & (4.44) \\
T_b & \longrightarrow & [{T_b}', {T_b}', w, w]\ (\{{T_{lb}}'\}; \emptyset) \mid & (4.45) \\
 & & b\ (\{F\}; \emptyset) & (4.46) \\
T_h & \longrightarrow & [w, {T_h}', {T_h}', w]\ (\{{T_{lb}}'\}; \emptyset) \mid & (4.47) \\
 & & b\ (\{F\}; \emptyset) & (4.48) \\
{T_{lb}}' & \longrightarrow & T_{lb}\ (\emptyset; \{T_{rb}, T_t, T_{ls}, T_b, T_h\}) & (4.49) \\
{T_{rb}}' & \longrightarrow & T_{rb}(\{T_{lb}\}; \emptyset) & (4.50) \\
{T_t}' & \longrightarrow & T_t\ (\{T_{lb}\}; \emptyset) & (4.51) \\
{T_{ls}}' & \longrightarrow & T_{ls}\ (\{T_{lb}\}; \emptyset) & (4.52) \\
{T_b}' & \longrightarrow & T_b\ (\{T_{lb}\}; \emptyset) & (4.53) \\
{T_h}' & \longrightarrow & T_h\ (\{T_{lb}\}; \emptyset) & (4.54) \\
F & \longrightarrow & b(\emptyset; \{T_{rb}, T_t, T_{ls}, T_b, T_h\}) & (4.55)
\end{array}$$

Figure 4.11: Random Context Picture Grammar rules for G_{Triangle}

We associate the colours light and dark with the terminals "w" and "b", respectively. At the beginning of the ith iteration of the main loop, $i \geq 1$, Rules 4.37–4.54, the pictorial form σ_i consists of 2^i squares of equal sizes on each side of the triangle. The square on the left bottom is labeled T_{lb}, the one on the lower right T_{rb} and the top left one T_t. All the squares on the left-hand side are T_{ls}, T_b at the bottom and T_h on the hypotenuse. The remaining squares in the pictorial form are labeled w.

σ_i can derive the hollow isosceles right triangle with side length 2^i or the pictorial form σ_{i+1}. The decision is made by T_{lb}. If T_{lb} decides to terminate, it produces an F (Rule 4.38). All other variables, on sensing F, produce a b by Rules 4.40, 4.42, 4.44, 4.46 and 4.48. Once this is done, F is also replaced by a b in Rule 4.55. Alternatively (Rule 4.37), σ_i derives σ_{i+1} by quartering all squares containing variables in such a way that each side contains twice as many variable labeled squares as before in Rules 4.39, 4.41, 4.43, 4.45 and 4.47. For example, each square labeled T_{ls} is divided into four, of which the left two squares are labeled ${T_{ls}}'$ and the other two w (Rule 4.43). Once this has been completed, the original variables are stored in Rules 4.49–4.54 and the main loop thereby ended. The pictures in Figure 4.12 show the third and fourth refinement of the



(a)

(b)

Figure 4.12: Some of the pictures generated by $G_{\textsf{Triangle}}$ [1]

hollow isosceles right triangle in Figure (a) and Figure (b) respectively.

We use the formula given in Lemma 4.2.1 to calculate the limits and bag adjustment of Figure 4.13. We have the equivalent BCPG as follows: $G_{\textsf{Bag-triangle}} = (\{S, T_{lb}, T_{rb}, T_t, T_{ls}, T_b, T_h, T_{lb}', T_{rb}', T_t', T_{ls}', T_b', T_h', F\}, \{w, b\}, R, (S, \theta), \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}, (1, 0, \ldots, 0))$ where R is the set of rules in Figure 4.13.

We have reached the end of this section. We have shown using examples how we can use Lemma 4.2.1 to convert any RCPG into an equivalent BCPG. In the next section we show how a picture is generated using BCPGs.

4.4 Picture Generation with Bag Context Picture Grammars

In this section we give an example of the CFPG in Section 4.3.1 where we used Lemma 4.2.1. We use the BCPG in Figure 4.14 to show how the picture develops in each stage after a rule is applied. We give Table 4.1 to show how the bag develops with the application of every rule. We then show the different pictorial forms at each stage. The CFPG used in this section is adapted from [1].

Example 4.4.1. We have the grammar $G_{\textsf{Bag-free}} = (\{S, C_{lb}, C_{rb}, C_{lt}, C_{rt}\}, \{w, b\}, R, (S, \theta)), \{1, 2, 3, 4, 5\}, (1, 0, 0, 0, 0))$ where R is the set of rules in Figure 4.14.

We first have the initial bag (1, 0, 0, 0, 0). Rule 4.1 is applied and the resultant pictorial form is in Figure 4.15. The pictorial form Π described in this figure is $\{(S, ((0, 0), (\frac{1}{2}, \frac{1}{2}))), (S, ((\frac{1}{2}, 0), (1, \frac{1}{2}))), (S, ((0, \frac{1}{2}), (\frac{1}{2}, 1))), (S, ((\frac{1}{2}, \frac{1}{2}), (1, 1)))\}$, then $\ell(\Pi) = \{S\}$. We then get the resultant bag in Table 4.1. Rule 4.2 is then applied to any of the S variables and the resultant pictorial form Ω is in Figure 4.16. The pictorial form Ω described in this figure is

$$
\begin{array}{rcl}
S & \longrightarrow & [T_{lb}, T_{rb}, T_t, w]\ ((1,0,\ldots,0), \infty; (-1,1,1,1,0,\ldots,0)) \\
T_{lb} & \longrightarrow & [{T_{lb}}', {T_b}', {T_{ls}}', w]\ ((0,1,0,\ldots,0), (\infty,\ldots,\infty,0,0,0,0,0,0); \\
 & & (0,-1,0,0,0,0,0,1,0,0,1,1,0,0))\ | \\
 & & F\ ((0,1,0,\ldots,0), (\infty,\infty,\infty,\infty,\infty,\infty,\infty,\infty,0,0,0,0,0,\infty); \\
 & & (0,-1,0,0,0,0,0,0,0,0,0,0,0,1)) \\
T_{rb} & \longrightarrow & [{T_b}', {T_{rb}}', {T_h}', w]\ ((0,0,1,0,0,0,0,1,0,0,0,0,0,0), \infty; \\
 & & (0,0,-1,0,0,0,0,0,1,0,0,1,1,0))\ | \\
 & & b\ ((0,0,1,0,0,0,0,1,0,0,0,0,0,1), \infty; (0,0,0,0,0,0,0,0,0,0,0,0,0,-1)) \\
T_t & \longrightarrow & [{T_{ls}}', {T_h}', {T_t}', w]\ ((0,0,0,1,0,0,0,1,0,0,0,0,0,0), \infty; \\
 & & (0,0,0,0,0,0,0,-1,0,1,1,0,1,0))\ | \\
 & & b\ ((0,0,0,1,0,0,0,0,0,0,0,0,0,1), \infty; (0,0,0,0,0,0,0,0,0,0,0,0,0,-1)) \\
T_{ls} & \longrightarrow & [{T_{ls}}', w, {T_{ls}}', w]\ ((0,0,0,0,1,0,0,1,0,0,0,0,0,0), \infty; \\
 & & (0,0,0,0,-1,0,0,0,0,0,1,0,0,0))\ | \\
 & & b\ ((0,0,0,0,1,0,0,0,0,0,0,0,0,1), \infty; (0,0,0,0,-1,0,0,0,0,0,0,0,0,0)) \\
T_b & \longrightarrow & [{T_b}', {T_b}', w, w]\ ((0,0,0,0,0,1,0,1,0,0,0,0,0,0), \infty; \\
 & & (0,0,0,0,0,-1,0,0,0,0,0,1,0,0))\ | \\
 & & b\ ((0,0,0,0,0,1,0,0,0,0,0,0,0,1), \infty; (0,0,0,0,0,-1,0,0,0,0,0,0,0,0)) \\
T_h & \longrightarrow & [w, {T_h}', {T_h}', w]\ ((0,0,0,0,0,0,1,1,0,0,0,0,0,0), \infty; \\
 & & (0,0,0,0,0,-1,1,0,0,0,0,0,0,0))\ | \\
 & & b\ ((0,0,0,0,0,0,1,0,0,0,0,0,0,1), \infty; (0,0,0,0,0,0,0,0,0,0,0,0,0,-1)) \\
{T_{lb}}' & \longrightarrow & T_{lb}\ ((0,0,0,0,0,0,0,1,0,0,0,0,0,0), (\infty,\infty,0,0,0,0,0,\infty,\ldots,\infty); \\
 & & (0,1,0,0,0,0,0,-1,0,0,0,0,0,0)) \\
{T_{rb}}' & \longrightarrow & T_{rb}\ ((0,1,0,0,0,0,0,0,1,0,0,0,0,0), \infty; (0,0,1,0,0,0,0,-1,0,0,0,0,0,0)) \\
{T_t}' & \longrightarrow & T_t\ ((0,1,0,0,0,0,0,0,0,1,0,0,0,0), \infty; (0,0,0,1,0,0,0,0,0,-1,0,0,0,0)) \\
{T_{ls}}' & \longrightarrow & T_{ls}\ ((0,1,0,0,0,0,0,0,0,0,1,0,0,0), \infty; (0,0,0,0,1,0,0,0,0,0,-1,0,0,0)) \\
{T_b}' & \longrightarrow & T_b\ ((0,1,0,0,0,0,0,0,0,0,0,1,0,0), \infty; (0,0,0,0,0,1,0,0,0,0,0,-1,0,0)) \\
{T_h}' & \longrightarrow & T_h\ ((0,1,0,0,0,0,0,0,0,0,0,1,0,0), \infty; (0,0,0,0,0,0,1,0,0,0,0,0,-1,0)) \\
F & \longrightarrow & b\ ((0,\ldots,0,1), (\infty,\infty,0,0,0,0,0,\infty,\infty,\infty,\infty,\infty,\infty,\infty); (0,\ldots,0,-1))
\end{array}
$$

Figure 4.13: Bag Context Picture Grammar rules for $G_{\textsf{Bag-triangle}}$

$$
\begin{aligned}
S &\longrightarrow [S,S,S,S]\ ((1,0,0,0,0),\infty;(3,0,0,0,0))| && (4.1)\\
&\qquad [C_{lb},C_{rb},C_{lt},C_{rt}]\ ((1,0,0,0,0),\infty;(-1,1,1,1,1)) && (4.2)\\
C_{lb} &\longrightarrow [w,b,b,b]\ ((0,1,0,0,0),\infty;(0,-1,0,0,0)) && (4.3)\\
C_{rb} &\longrightarrow [b,w,b,b]\ ((0,0,1,0,0),\infty;(0,0,-1,0,0)) && (4.4)\\
C_{lt} &\longrightarrow [b,b,w,b]\ ((0,0,0,1,0),\infty;(0,0,0,-1,0)) && (4.5)\\
C_{rt} &\longrightarrow [b,b,b,w]\ ((0,0,0,0,1),\infty;(0,0,0,0,-1)) && (4.6)
\end{aligned}
$$

Figure 4.14: Bag Context Picture Grammar rules for $G_{\textsf{Bag-free}}$

rule	λ	μ	α	$\beta' = \beta + \alpha$
4.1	(1, 0, 0, 0, 0)	∞	(3, 0, 0, 0, 0)	(4, 0, 0, 0, 0)
4.2	(1, 0, 0, 0, 0)	∞	(-1, 1, 1, 1, 1)	(3, 1, 1, 1, 1)
4.2	(1, 0, 0, 0, 0)	∞	(-1, 1, 1, 1, 1)	(2, 2, 2, 2, 2)
4.2	(1, 0, 0, 0, 0)	∞	(-1, 1, 1, 1, 1)	(1, 3, 3, 3, 3)
4.2	(1, 0, 0, 0, 0)	∞	(-1, 1, 1, 1, 1)	(0, 4, 4, 4, 4)
4.4	(0, 0, 1, 0, 0)	∞	(0, 0, -1, 0, 0)	(0, 4, 3, 4, 4)
4.3	(0, 1, 0, 0, 0)	∞	(0, -1, 0, 0, 0)	(0, 3, 3, 4, 4)
4.3	(0, 1, 0, 0, 0)	∞	(0, -1, 0, 0, 0)	(0, 2, 3, 4, 4)
4.3	(0, 1, 0, 0, 0)	∞	(0, -1, 0, 0, 0)	(0, 1, 3, 4, 4)
4.5	(0, 0, 0, 1, 0)	∞	(0, 0, 0, -1, 0)	(0, 1, 3, 3, 4)
4.6	(0, 0, 0, 0, 1)	∞	(0, 0, 0, 0, -1)	(0, 1, 3, 3, 3)
4.4	(0, 0, 1, 0, 0)	∞	(0, 0, -1, 0, 0)	(0, 1, 2, 3, 3)
4.5	(0, 0, 0, 1, 0)	∞	(0, 0, 0, -1, 0)	(0, 1, 2, 2, 3)
4.3	(0, 1, 0, 0, 0)	∞	(0, -1, 0, 0, 0)	(0, 0, 2, 2, 3)
4.5	(0, 0, 0, 1, 0)	∞	(0, 0, 0, -1, 0)	(0, 0, 2, 1, 3)
4.6	(0, 0, 0, 0, 1)	∞	(0, 0, 0, 0, -1)	(0, 0, 2, 1, 2)
4.5	(0, 0, 0, 1, 0)	∞	(0, 0, 0, -1, 0)	(0, 0, 2, 0, 2)
4.6	(0, 0, 0, 0, 1)	∞	(0, 0, 0, 0, -1)	(0, 0, 2, 0, 1)
4.6	(0, 0, 0, 0, 1)	∞	(0, 0, 0, 0, -1)	(0, 0, 2, 0, 0)
4.4	(0, 0, 1, 0, 0)	∞	(0, 0, -1, 0, 0)	(0, 0, 1, 0, 0)
4.4	(0, 0, 1, 0, 0)	∞	(0, 0, -1, 0, 0)	(0, 0, 0, 0, 0)

Table 4.1: Development of the bag for picture in Figure 4.18

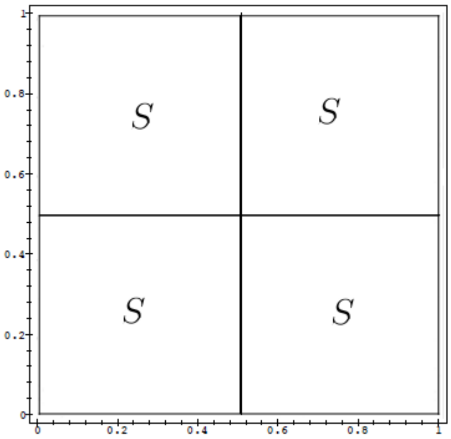


Figure 4.15: The pictorial form after Rule 4.1 is applied

$\{(C_{lb}, ((0,0),(\frac{1}{4},\frac{1}{4}))), (C_{rb}, ((\frac{1}{4},0),(\frac{1}{2},\frac{1}{4}))), (C_{lt}, ((0,\frac{1}{4}),(\frac{1}{4},\frac{1}{2}))), (C_{rt}, ((\frac{1}{4},\frac{1}{4}),(\frac{1}{2},\frac{1}{2}))), (S, ((\frac{1}{2},0),(1,\frac{1}{2}))), (S, ((0,\frac{1}{2}),(\frac{1}{2},1))), (S, ((\frac{1}{2},\frac{1}{2}),(1,1)))\}$, then $\ell(\Omega) = \{C_{lb}, C_{rb}, C_{lt}, C_{rt}, S\}$.

A look at Table 4.1 shows that Rule 4.2 was applied four times with the resultant bags shown in the table. Rule 4.2 is applied to any of the remaining S variables so that each S is divided into four squares with the labels C_{lb}, C_{rb}, C_{lt} and C_{rt} to get the pictorial form in Figure 4.17. Rules 4.3–4.6 are applied in any order with the bags shown in the table. We get the picture in Figure 4.18. It should be noted that this grammar may generate a variety of pictures. We used the above description to show one way in which the picture may develop. This grammar also generates the picture in Figure 4.19 amongst others.

4.5 Chapter Summary

This brings us to the end of this chapter. We have defined BCPGs and used them to generate pictures with dark crosses on a light background and hollow isosceles right-angled triangles. We also defined a lemma that converts RCPGs into equivalent BCPGs. In the next chapter we give examples on how we use BCPGs to generate pictures that are the same as or similar to pictures that were generated by grammars in this chapter.

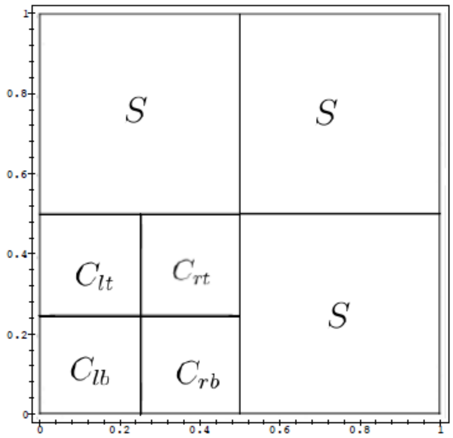


Figure 4.16: The pictorial form after Rule 4.2 is applied

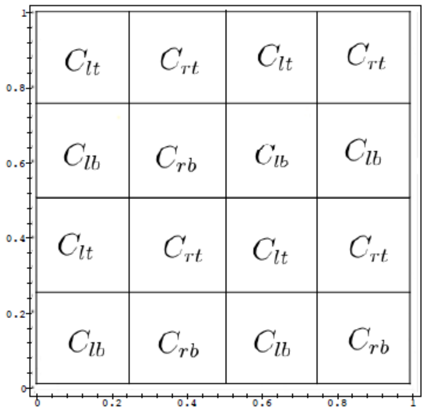


Figure 4.17: The pictorial form after Rule 4.2 is applied

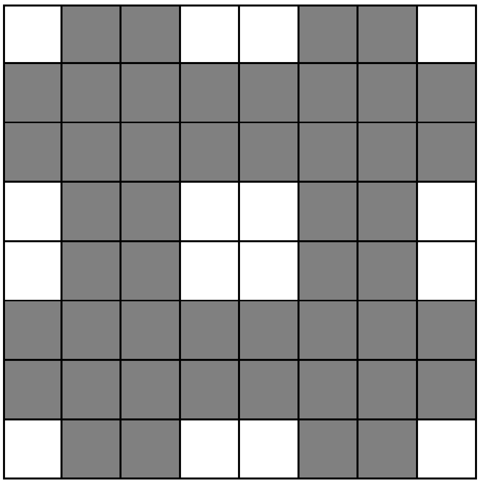


Figure 4.18: Picture generated by Figure 4.14

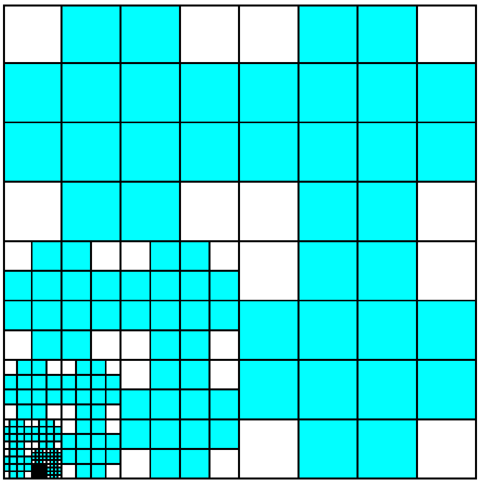


Figure 4.19: A possible picture generated by $G_{\textsf{Bag-free}}$

Chapter 5

Galleries Generated by Bag Context Picture Grammars

In this chapter we showed how we use BCPGs to generate similar pictures. We translated the grammars in Chapter 4 into BCPGs with Lemma 4.2.1 from the previous chapter and used them as examples to show how BCPGs are used to generate pictures that are the same and those that are similar to the ones given in Chapter 4. We looked at how picture sets change when we change the lower limits and upper limits. Some of the rules in the grammars were not applied at all. This is because when we want certain sets of pictures it is not necessary to apply these rules. In addition we looked at colour distribution in pictures. We looked at how we control application of rules to have a certain amount of colour in a picture, for example, we showed how we count the number of pink crosses we want in a picture. We modified some of the grammars by adding new nonterminals and terminals. The added nonterminals and terminals helped us to create colour variation in the pictures.

5.1 Context Free Picture Grammars Case

In this section we look at the grammar G_{Free} that generates a gallery of pictures with crosses as shown in Section 4.3.1. In this instance we change the lower and upper limits to control how the rules are applied. The following example is a variation of the BCPG $G_{\mathsf{Bag\text{-}free}}$ given in Section 4.3.1, which is the equivalent BCPG for G_{Free}.

Example 5.1.1. Consider a BCPG $G_{\mathsf{Cross}} = (\{S, C_{lb}, C_{rb}, C_{lt}, C_{rt}\}, \{w, b\}, R, (S, \theta),$-$\{1, 2, 3, 4, 5\}, (1, 0, 0, 0, 0))$ where R is the set of rules in Figure 5.1.

In this example the bag is used to control how rules are applied. In this instance we prohibit the application of Rule 5.1. This grammar only generates one picture. We associate the non-terminals $S, C_{lb}, C_{rb}, C_{lt}$ and C_{rt} with the bag positions 1 to 5 respectively. We use the bag in this example to restrict Rule 5.1 from being applied first. This is done by setting the lower limit to 2. When the picture generation process starts, the inequality does not hold for this rule and therefore the rule is not applied. However, Rule 5.2 can be applied and the resultant nonterminals are C_{lb}, C_{rb}, C_{lt} and C_{rt}. This rule can only be applied once due to the nature of the grammar since, only one S produces C_{lb}, C_{rb}, C_{lt}

$$
\begin{aligned}
S &\longrightarrow [S,S,S,S]\ (2,2;(3,0,0,0,0))\ | && (5.1)\\
&\ [C_{lb},C_{rb},C_{lt},C_{rt}]\ ((1,0,0,0,0),5;(-1,1,1,1,1)) && (5.2)\\
C_{lb} &\longrightarrow [b,b,b,w]\ (-5,5;(0,-1,0,0,0)) && (5.3)\\
C_{rb} &\longrightarrow [b,b,w,b]\ (-5,5;(0,0,-1,0,0)) && (5.4)\\
C_{lt} &\longrightarrow [b,w,b,b]\ (-5,5;(0,0,0,-1,0)) && (5.5)\\
C_{rt} &\longrightarrow [w,b,b,b]\ (-5,5;(0,0,0,0,-1) && (5.6)
\end{aligned}
$$

Figure 5.1: Rules that generate picture in Figure 5.2

rule	λ	μ	α	$\beta' = \beta + \alpha$
5.2	(1,0,0,0,0)	5	(-1,1,1,1,1)	(0, 1, 1, 1, 1)
5.3	-5	5	(0,-1,0,0,0)	(0, 0, 1, 1, 1)
5.4	-5	5	(0,0,-1,0,0)	(0, 0, 0, 1, 1)
5.4	-5	5	(0,0,-1,0,0)	(0, 0, -1, 1, 1)
5.6	-5	5	(0,0,0,0,-1)	(0, 0, -1, 1, 0)
5.3	-5	5	(0,-1,0,0,0)	(0, -1, -1, 1, 0)
5.5	-5	5	(0,0,0,-1,0)	(0, -1, -1, 0, 0)
5.4	-5	5	(0,0,-1,0,0)	(0, -1, -2, 0, 0)
5.6	-5	5	(0,0,0,0,-1)	(0, -1, -2, 0, -1)
5.6	-5	5	(0,0,0,0,-1)	(0, -1, -2, 0, -2)
5.3	-5	5	(0,-1,0,0,0)	(0, -2, -2, 0, -2)
5.5	-5	5	(0,0,0,-1,0)	(0, -2, -2, -1, -2)
5.6	-5	5	(0,0,0,0,-1)	(0, -2, -2, -1, -3)
5.4	-5	5	(0,0,-1,0,0)	(0, -2, -3, -1, -3)
5.5	-5	5	(0,0,0,-1,0)	(0, -2, -3, -2, -3)
5.3	-5	5	(0,-1,0,0,0)	(0, -3, -3, -2, -3)
5.5	-5	5	(0,0,0,-1,0)	(0, -3, -3, -3, -3)

Table 5.1: Development of bag for picture in Figure 5.2

and C_{rt}. Rules 5.3–5.6 are then applied in any order until the picture generation process ends. We get the picture in Figure 5.2. Table 5.1 shows the development of the bag. The bag may or may not develop in the same way every time the picture is generated so Table 5.1 shows a possible way in which the bag develops after the application of each rule.

In the next example we use the limits and bag adjustment to count the number of S non-terminals in the pictorial form. In these pictures we want to have exactly one quarter of the picture to be more refined than the rest of the picture. Other rules are prohibited from being applied until there are exactly 7 S variables in the pictorial form. This grammar is another variation of G_{Free} from the previous chapter.

Example 5.1.2. The following BCPG gives us the pictures in Figure 5.4. $G_{\mathsf{Bag\text{-}free}}$ $= (\{S, C_{lb}, C_{rb}, C_{lt}, C_{rt}\}, \{w, b\}, R, (S, \theta), \{1, 2, 3, 4, 5\}, (1, 0, 0, 0, 0))$ where R is the set of rules in Figure 5.3.

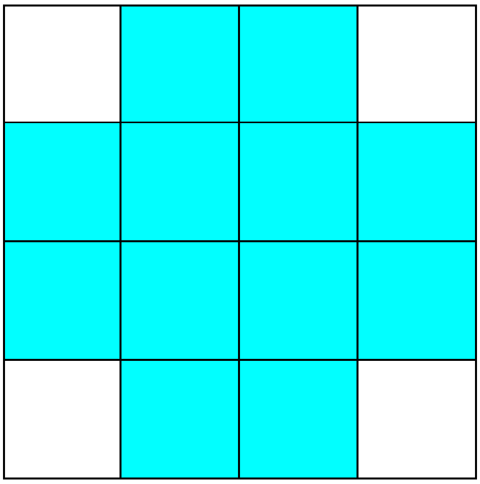


Figure 5.2: Picture generated by G_{Cross}

$$
\begin{aligned}
S &\longrightarrow [S,S,S,S]\ ((1,0,0,0,0),(6,0,0,0,0);(3,0,0,0,0))\ | && (5.7)\\
&\ [C_{lb},C_{rb},C_{lt},C_{rt}]\ ((6,0,0,0,0),\infty;(7,14,14,14,14)) && (5.8)\\
C_{lb} &\longrightarrow [b,b,b,w]\ (14,\infty;0) && (5.9)\\
C_{rb} &\longrightarrow [b,b,w,b]\ (14,\infty;0) && (5.10)\\
C_{lt} &\longrightarrow [b,w,b,b]\ (14,\infty;0) && (5.11)\\
C_{rt} &\longrightarrow [w,b,b,b]\ (14,\infty;0) && (5.12)
\end{aligned}
$$

Figure 5.3: $G_{\mathsf{Bag\text{-}free}}$ rules that generate pictures in Figure 5.4

rule	λ	μ	α	$\beta' = \beta + \alpha$
5.7 (×2)	(1, 0, 0, 0, 0)	(6, 0, 0, 0, 0)	(3, 0, 0, 0, 0)	(4, 0, 0, 0, 0)
5.8 (×7)	(7, 0, 0, 0, 0)	∞	(7, 14, 14, 14, 14)	14
5.9	14	∞	0	14
5.10	14	∞	0	14
5.11	14	∞	0	14
5.12	14	∞	0	14

Table 5.2: Development of bag for one of the pictures in Figure 5.4

We associate the variables $S, C_{lb}, C_{rb}, C_{lt}$ and C_{rt} with the bag positions 1 to 5. Initially, Rule 5.7 is applied because the lower limit is set to (1, 0, 0, 0, 0). Then the first bag position is 4. We apply Rule 5.7 again because we have set the lower limit to (6, 0, 0, 0, 0) in Rule 5.8 and 14 in Rules 5.9–5.12. We have done this so that we are forced to apply Rule 5.7 again. This results in any S variable being divided into four more S variables and resulting in one quarter of the picture being more refined than the other parts. At this point the inequality in this rule no longer holds ($(7, 0, 0, 0, 0) \geqslant (6, 0, 0, 0, 0)$).

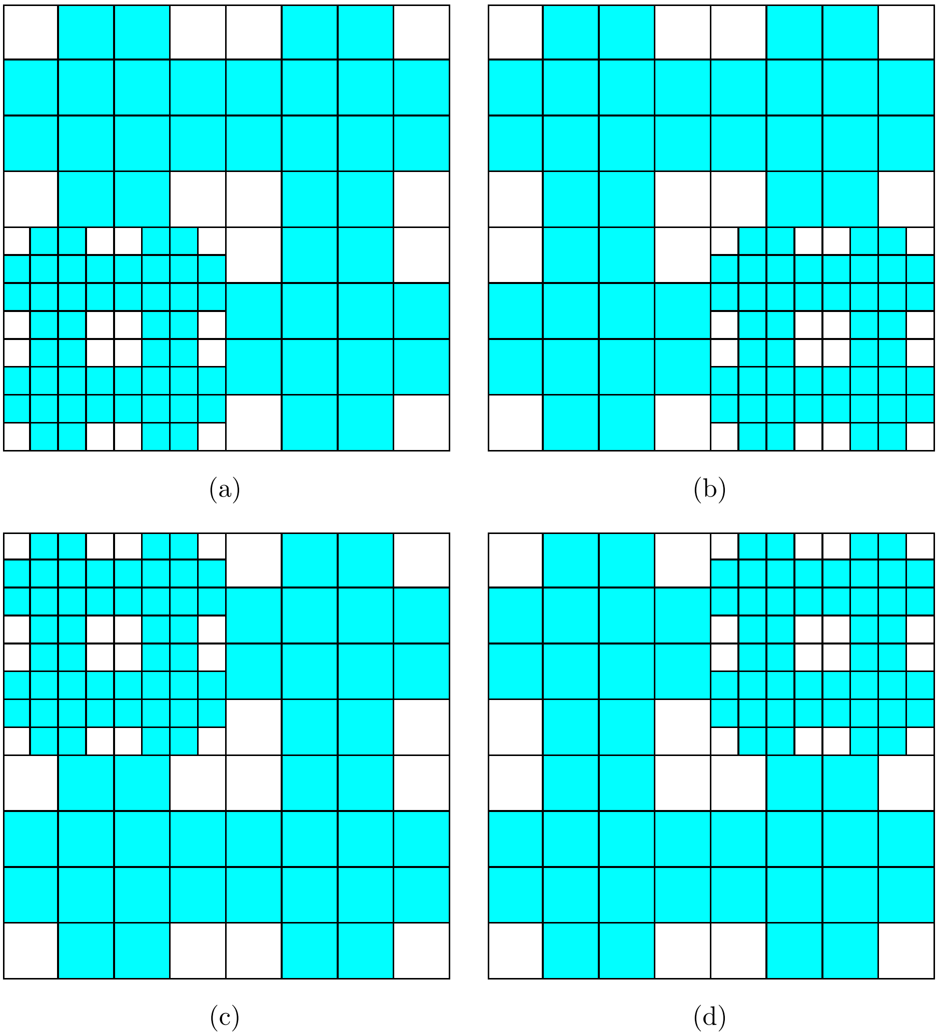
Rule 5.8 is applied next as it has lower limit (6, 0, 0, 0, 0). This rule is applied until the bag becomes 14. The bag adjustment α is set to 0 (Rules 5.9–5.12) so that the bag remains the same as the rules are applied. Rules 5.9–5.12 are then applied in any order until we obtain one of the pictures in Figure 5.4. Figure 5.4 shows four similar pictures that can be generated with this grammar. We see that each of the four pictures has one quarter refined one level more than the other three quarters in the picture. In this case the refined quarter is in the second level of refinement while the rest of the picture is in the first level. Rule 5.8 is applied seven times to convert the S variables into the variables C_{lb}, C_{rb}, C_{lt} and C_{rt}. In order to avoid a table that is too long we add the bag adjustment to the bag six times because the rule is applied seven times. The bag remains the same while the rest of the rules are applied because the bag adjustment for Rules 5.9–5.12 is 0. Table 5.2 shows a possible way in which the bag develops with the application of each rule in the picture generation process.

In the next two examples we look at the color distribution cases. New variables are added to G_{Free} that result in crosses that have a different color from the original blue crosses. This is done to show color distribution in the pictures. The variables, B_{lb}, B_{rb}, B_{lt} and B_{rt} produce terminal "p" which represents the color pink and results in pink crosses while the terminal "b" represents the color blue which results in blue crosses.

Example 5.1.3. Consider the BCPG $G_{\mathsf{Color\text{-}crosses}} = (\{S, B_{lb}, B_{rb}, B_{lt}, B_{rt}, C_{lb}, C_{rb}, C_{lt}, C_{rt}\}, \{w, b, p\}, R, (S, \theta), \{1, 2, 3, 4, 5, 6, 7, 8, 9\}, (1, 0, 0, 0, 0, 0, 0, 0, 0))$ where R is the set of rules in Figure 5.5.

We associate the non-terminals $S, B_{lb}, B_{rb}, B_{lt}, B_{rt}, C_{lb}, C_{rb}, C_{lt}$ and C_{rt} with the bag positions 1 to 9 respectively. In this example we want to generate pictures that have four pink crosses. The crosses can be at any position in the picture and may be of any size. We demonstrate how the bag develops in Table 5.3. Rule 5.13 is applied four times to get thirteen S variables. This is enforced by the bag. Rule 5.14 is applied next, also four times so that any four S variables each divides into B_{lb}, B_{rb}, B_{lt} and B_{rt}. This will eventually result in 4 pink crosses.

The rest of the S variables will become the blue crosses. When the bag gets to the same value as the lower limit in Rule 5.15, then this rule is activated. We then apply Rules 5.15–5.23 until the process terminates so that we get one of the pictures in Figure 5.6. We have four pictures generated by $G_{\mathsf{Color\text{-}crosses}}$ where one sees that the pink crosses in Figure 5.6(a), (b) and (c) have different sizes. When one examines Figure 5.6(c), one finds that each pink cross is in a different level of refinement, with these levels ranging



(a) (b)

(c) (d)

Figure 5.4: Pictures generated by $G_{\textsf{Bag-free}}$

$$
\begin{aligned}
S &\longrightarrow [S, S, S, S]\ ((1, 0, \ldots, 0), (1, 3, \ldots, 3); (0, 1, \ldots, 1))\ | \quad (5.13)\\
&\ [B_{lb}, B_{rb}, B_{lt}, B_{rt}]\ ((1, 4, \ldots, 4), (4, 7, \ldots, 7); (0, 1, \ldots, 1))\ | \quad (5.14)\\
&\ [C_{lb}, C_{rb}, C_{lt}, C_{rt}]\ ((5, 8, \ldots, 8), \infty; 0) \quad (5.15)\\
B_{lb} &\longrightarrow [p, p, p, w]\ ((5, 8, \ldots, 8), \infty; 0) \quad (5.16)\\
B_{rb} &\longrightarrow [p, p, w, p]\ ((5, 8, \ldots, 8), \infty; 0) \quad (5.17)\\
B_{lt} &\longrightarrow [p, w, p, p]\ ((5, 8, \ldots, 8), \infty; 0) \quad (5.18)\\
B_{rt} &\longrightarrow [w, p, p, p]\ ((5, 8, \ldots, 8), \infty; 0) \quad (5.19)\\
C_{lb} &\longrightarrow [b, b, b, w]\ ((5, 8, \ldots, 8), \infty; 0) \quad (5.20)\\
C_{rb} &\longrightarrow [b, b, w, b]\ ((5, 8, \ldots, 8), \infty; 0) \quad (5.21)\\
C_{lt} &\longrightarrow [b, w, b, b]\ ((5, 8, \ldots, 8), \infty; 0) \quad (5.22)\\
C_{rt} &\longrightarrow [w, b, b, b]\ ((5, 8, \ldots, 8), \infty; 0) \quad (5.23)
\end{aligned}
$$

Figure 5.5: Bag Context Picture Grammar rules for $G_{\textsf{Color-crosses}}$

rule	λ	μ	α	$\beta' = \beta + \alpha$
5.13 (×4)	(1, 0, . . . , 0)	(1, 3, . . . , 3)	(0, 1, . . . , 1)	(1, 4, . . . , 4)
5.14 (×4)	(1, 4, . . . , 4)	(4, 7, . . . , 7)	1	(5, 8, . . . , 8)
5.15	(5, 8, . . . , 8)	∞	0	(5, 8, . . . , 8)
5.16	(5, 8, . . . , 8)	∞	0	(5, 8, . . . , 8)
5.17	(5, 8, . . . , 8)	∞	0	(5, 8, . . . , 8)
5.18	(5, 8, . . . , 8)	∞	0	(5, 8, . . . , 8)
5.19	(5, 8, . . . , 8)	∞	0	(5, 8, . . . , 8)
5.20	(5, 8, . . . , 8)	∞	0	(5, 8, . . . , 8)
5.21	(5, 8, . . . , 8)	∞	0	(5, 8, . . . , 8)
5.22	(5, 8, . . . , 8)	∞	0	(5, 8, . . . , 8)
5.23	(5, 8, . . . , 8)	∞	0	(5, 8, . . . , 8)

Table 5.3: Development of the bag for one of the pictures in Figure 5.6

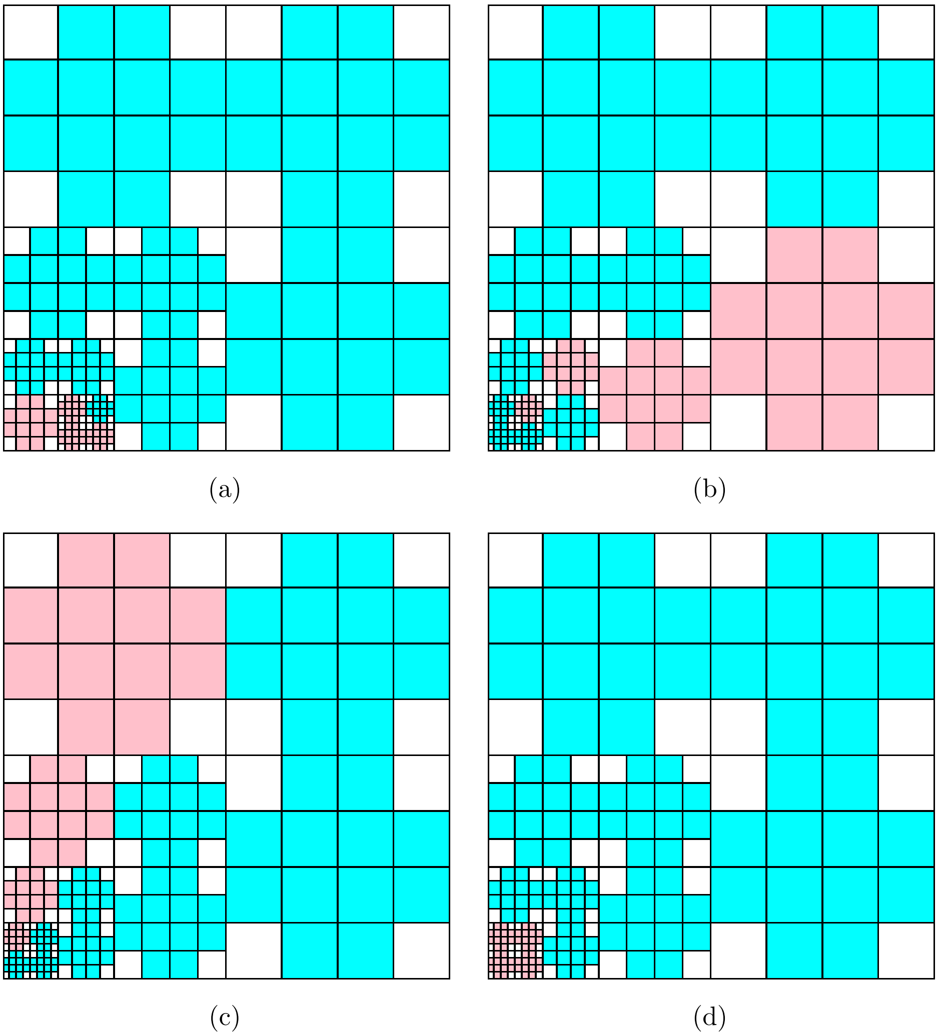
from the first level to the fourth level. The picture in Figure 5.6(d) on the other hand has four pink crosses with the same size and in the same region and levels of refinements.

Table 5.3 shows a possible way the bag develops during the derivation process. For Rules 5.16–5.23 the bag adjustment value is 0. This is because at this point, we have the four pink crosses we wanted. The rules can be applied in any order from this point until the process terminates.

The next example is similar but in every picture, we require exactly seven blue crosses while the rest of the crosses in a picture are pink crosses. We note that the crosses may be at any position in the picture and may be of any size. The important thing is that we control the number of blue crosses in the picture.

Example 5.1.4. Let $G_{\mathsf{Blue\text{-}crosses}} = (\{S, B_{lb}, B_{rb}, B_{lt}, B_{rt}, C_{lb}, C_{rb}, C_{lt}, C_{rt}\}, \{w, b, p\}, R,$-$(S, \theta), \{1, 2, 3, 4, 5, 6, 7, 8, 9\}, (1, 0, 0, 0, 0, 0, 0, 0, 0))$ where R is the set of rules in Figure 5.7.

We associate the non-terminals $S, B_{lb}, B_{rb}, B_{lt}, B_{rt}, C_{lb}, C_{rb}, C_{lt}$ and C_{rt} with the bag positions 1 to 9 respectively. In this example we want to generate pictures that have exactly seven blue crosses. We demonstrate how the bag develops in Table 5.4. At first Rule 5.24 is applied. This rule can be applied a maximum of five times since the upper limit is (1, 4, 4, 4, 4). Rule 5.26 is then applied seven times so that any seven S variables each divides into C_{lb}, C_{rb}, C_{lt} and C_{rt} in order for the pictures to have exactly seven blue crosses. The rest of the S variables will become B_{lb}, B_{rb}, B_{lt} and B_{rt} and these variables produce the pink crosses. Rules 5.25, 5.27–5.34 are applied until the process terminates and we get one of the pictures in Figure 5.8. The picture in Figure 5.8(a) is more refined than the picture in Figure 5.8(b). This is because in Figure 5.8(a), Rule 5.24 is applied five times while it was applied four times in Figure 5.8(b). These pictures are similar in shape or structure but different in refinement. Figure 5.8(a) is in the fifth level of refinement while Figure 5.8(b) is in the fourth level. Finally, in each picture there are



(a) (b) (c) (d)

Figure 5.6: Some of the pictures generated by $G_{\textsf{Color-crosses}}$ with four pink crosses

$$\begin{array}{rcll}
S & \longrightarrow & [S,S,S,S]\ ((1,0,\ldots,0),(1,4,\ldots,4);(0,1,\ldots,1))\ | & (5.24)\\
 & & [B_{lb},B_{rb},B_{lt},B_{rt}]\ ((8,11,\ldots,11),\infty;0)\ | & (5.25)\\
 & & [C_{lb},C_{rb},C_{lt},C_{rt}]\ ((1,4,\ldots,4),(7,10,\ldots,10);1) & (5.26)\\
B_{lb} & \longrightarrow & [p,p,p,w]\ ((8,11,\ldots,11),\infty;0) & (5.27)\\
B_{rb} & \longrightarrow & [p,p,w,p]\ ((8,11,\ldots,11),\infty;0) & (5.28)\\
B_{lt} & \longrightarrow & [p,w,p,p]\ ((8,11,\ldots,11),\infty;0) & (5.29)\\
B_{rt} & \longrightarrow & [w,p,p,p]\ ((8,11,\ldots,11),\infty;0) & (5.30)\\
C_{lb} & \longrightarrow & [b,b,b,w]\ ((8,11,\ldots,11),\infty;0) & (5.31)\\
C_{rb} & \longrightarrow & [b,b,w,b]\ ((8,11,\ldots,11),\infty;0) & (5.32)\\
C_{lt} & \longrightarrow & [b,w,b,b]\ ((8,11,\ldots,11),\infty;0) & (5.33)\\
C_{rt} & \longrightarrow & [w,b,b,b]\ ((8,11,\ldots,11),\infty;0) & (5.34)
\end{array}$$

Figure 5.7: Bag Context Picture Grammar rules for $G_{\textsf{Blue-crosses}}$

rule	λ	μ	α	$\beta' = \beta + \alpha$
5.24(×4)	(1, 0, . . . , 0)	(1, 4, . . . , 4)	(0, 1, . . . , 1)	(1, 4, . . . , 4)
5.26 (×7)	(1, 4, . . . , 4)	(7, 10, . . . , 10)	1	(8, 11, . . . , 11)
5.25	(8, 11, . . . , 11)	∞	0	(8, 11, . . . , 11)
5.27	(8, 11, . . . , 11)	∞	0	(8, 11, . . . , 11)
5.28	(8, 11, . . . , 11)	∞	0	(8, 11, . . . , 11)
5.29	(8, 11, . . . , 11)	∞	0	(8, 11, . . . , 11)
5.30	(8, 11, . . . , 11)	∞	0	(8, 11, . . . , 11)
5.31	(8, 11, . . . , 11)	∞	0	(8, 11, . . . , 11)
5.32	(8, 11, . . . , 11)	∞	0	(8, 11, . . . , 11)
5.33	(8, 11, . . . , 11)	∞	0	(8, 11, . . . , 11)
5.34	(8, 11, . . . , 11)	∞	0	(8, 11, . . . , 11)

Table 5.4: Development of bag for one of the pictures in Figure 5.8

exactly seven blue crosses. In addition, the blue crosses in Figure 5.8(b),(c) and (d) are bigger in size than those in Figure 5.8(a).

In the next section we look at the RPCPG case. Again we look at how the shape changes from one picture to the next and how the color is distributed.

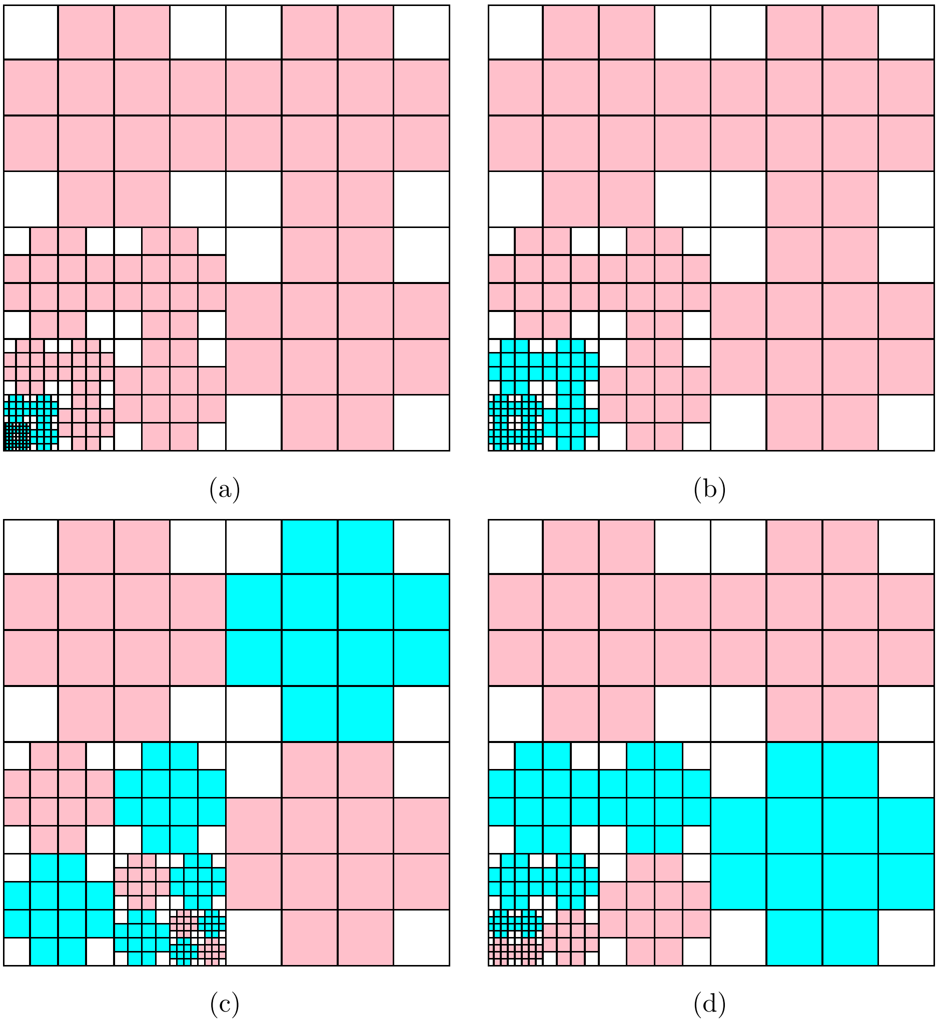
5.2 Random Permitting Context Picture Grammars Case

In the previous chapter, we discussed how the RPCPG in Example 4.3.2 generated pictures with crosses in the four quarters of the picture where, the left top and right bottom parts of the picture are more refined than the rest of the picture. We use the same grammar and show how we generate similar pictures. We show similarity in terms of shape and color. In this section we give four examples.

Example 5.2.1. Let $G_{\textsf{Bag-permit}} = (\{S, A, B, C, A', B', F, C_{lb}, C_{rb}, C_{lt}, C_{rt}\}, \{w, b\}, R, (S,$-$\theta), \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\},\ (1, 0, \ldots, 0))$ where R is the set of rules in Figure 5.9.

Table 5.5 shows how the bag develops with the application of each rule.

We want to generate a picture that has the left top part more refined than the bottom right part of the picture. This grammar is controlled in such a way that it generates one picture. This is done by applying Rules 5.1, 5.4, 5.7 and 5.5 first and in this order. To do this, the lower limits are set up in increasing order according to the sequence we want the rules to be applied in. Each rule is applied once. This sequence results in the right bottom cross in the second level of refinement while the left bottom and right top are in the first level of refinement. Rules 5.2, 5.6, 5.3, 5.8–5.13 are then applied next. These Rules 5.2 and 5.6 are applied twice each. We enforce this by setting up the lower and upper limit in the manner we did in Figure 5.9. The rest of the rules are applied until the process terminates so that the left top part of the picture is in the third level of


(a)

(b)

(c)

(d)

Figure 5.8: Pictures generated by $G_{\mathsf{Blue\text{-}crosses}}$ with seven blue crosses

rule	λ	μ	α	$\beta' = \beta + \alpha$
5.1	(1, 0, . . . , 0)	(1, 0, . . . , 0)	(0, 1,. . . , 1)	1
5.4	1	1	1	2
5.7	2	2	1	3
5.5	3	3	1	4
5.2 (×2)	4	6	1	6
5.6 (×2)	5	7	1	7
5.3	7	∞	1	8
5.8	8	∞	0	8
5.9	8	∞	0	8
5.10	8	∞	0	8
5.11	8	∞	0	8
5.12	8	∞	0	8
5.13	8	∞	0	8

Table 5.5: Development of bag for the picture in Figure 5.10

$$\begin{array}{rcll}
S & \longrightarrow & [C, B, A, C]\ ((1, 0, \ldots, 0), (1, 0, \ldots, 0); (0, 1, \ldots, 1)) & (5.1)\\
A & \longrightarrow & [C, C, A', C]\ (4, 6; 1)\ | & (5.2)\\
 & & F(7, \infty; 0) & (5.3)\\
B & \longrightarrow & [C, B', C, C]\ (1, 1; 1)\ | & (5.4)\\
 & & F(3, 3; 1) & (5.5)\\
A' & \longrightarrow & A\ (5, 7; 1) & (5.6)\\
B' & \longrightarrow & B\ (2, 2; 1) & (5.7)\\
F & \longrightarrow & C(8, \infty; 0) & (5.8)\\
C & \longrightarrow & [C_{lb}, C_{rb}, C_{lt}, C_{rt}]\ (8, \infty; 0) & (5.9)\\
C_{lb} & \longrightarrow & [w, b, b, b]\ (8, \infty; 0) & (5.10)\\
C_{rb} & \longrightarrow & [b, w, b, b]\ (8, \infty; 0) & (5.11)\\
C_{lt} & \longrightarrow & [b, b, w, b]\ (8, \infty; 0) & (5.12)\\
C_{rt} & \longrightarrow & [b, b, b, w]\ (8, \infty; 0) & (5.13)
\end{array}$$

Figure 5.9: Bag Context Picture Grammar rules for $G_{\textsf{Bag-permit}}$

refinement. This process results in the picture in Figure 5.10. This grammar generates one picture.

In the next example we generate a picture where the right bottom part of the picture is more refined than the other parts.

Example 5.2.2. Let $G_{\textsf{Permit-control}} = (\{S, A, B, C, A', B', F, C_{lb}, C_{rb}, C_{lt}, C_{rt}\}, \{w, b\}, R, (S,\text{-}\theta), \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}, (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0))$ where R is the set of rules in Figure 5.11.

Table 5.6 shows how the bag develops with the application of each rule.

In this example we want to show the level of control BCPGs gives us in terms of having a picture refined to an exact level of refinement. Here we want the top left part of the picture to have a cross in the third level of refinement and the right bottom part of the picture to have a cross in the fifth level of refinement. The grammar limits are set up in a way that is very strict in terms of how the rules are applied in this example, hence the grammar produces only one picture. Rules 5.14, 5.15, 5.19 and 5.16 are applied first in order of a sequence such that the result is the left top part of the picture in the third level of refinement. This is done by setting the lower bound, upper bound and bag adjustment as shown in Figure 5.11. Rules 5.17 and 5.20 are then applied four times by setting the lower limits to 7 and 8 respectively. We set the upper limit to 14 to enforce a sequence in which these rules are applied four times each until the bag becomes 15. We then apply Rule 5.18 once and Rules 5.21–5.26 until the picture is developed. We have the resultant picture in Figure 5.12. Other pictures can be generated with this grammar by changing the limits.

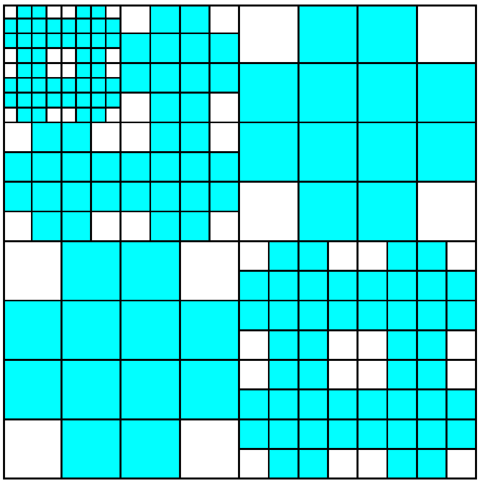


Figure 5.10: Picture generated by the rules in Figure 5.9

$$
\begin{array}{rcll}
S & \longrightarrow & [C, B, A, C]\ ((1,0,\ldots,0),(1,0,\ldots,0);(1,2,\ldots,2)) & (5.14)\\
A & \longrightarrow & [C, C, A', C]\ (2,5;1)\ | & (5.15)\\
 & & F\ (6,6;1) & (5.16)\\
B & \longrightarrow & [C, B', C, C]\ (7,14;1)\ | & (5.17)\\
 & & F\ (15,15;1) & (5.18)\\
A' & \longrightarrow & A\ (3,5;1) & (5.19)\\
B' & \longrightarrow & B\ (8,14;1) & (5.20)\\
F & \longrightarrow & C\ (16,\infty;1) & (5.21)\\
C & \longrightarrow & [C_{lb}, C_{rb}, C_{lt}, C_{rt}]\ (16,\infty;0) & (5.22)\\
C_{lb} & \longrightarrow & [w,b,b,b]\ (16,\infty;0) & (5.23)\\
C_{rb} & \longrightarrow & [b,w,b,b]\ (16,\infty;0) & (5.24)\\
C_{lt} & \longrightarrow & [b,b,w,b]\ (16,\infty;0) & (5.25)\\
C_{rt} & \longrightarrow & [b,b,b,w]\ (16,\infty;0) & (5.26)
\end{array}
$$

Figure 5.11: Bag Context Picture Grammar rules for $G_{\textsf{Permit-control}}$

rule	λ	μ	α	$\beta' = \beta + \alpha$
5.14	(1, 0, ...,0)	(1, 0, ...,0)	(1, 2, ...,2)	2
5.15 (×2)	2	5	1	5
5.19 (×2)	3	5	1	6
5.16 (×1)	6	6	1	7
5.17 (×4)	7	14	1	14
5.20 (×4)	8	14	1	15
5.18 (×1)	15	15	1	16
5.21 (×2)	16	∞	1	18
5.22	16	∞	0	18
5.23	16	∞	0	18
5.24	16	∞	0	18
5.25	16	∞	0	18
5.26	16	∞	0	18

Table 5.6: Development of bag for the picture in Figure 5.12

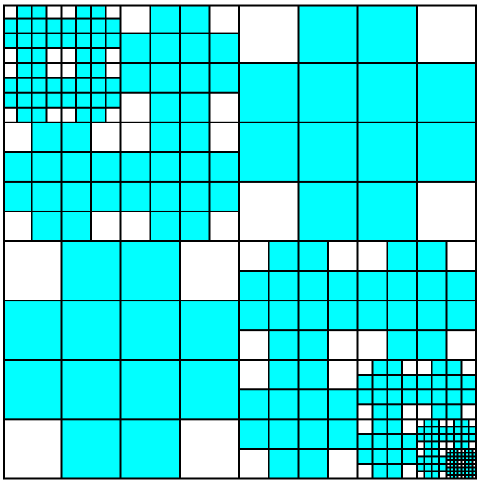


Figure 5.12: Picture generated by $G_{\mathsf{Permit\text{-}control}}$

$$
\begin{align}
S &\longrightarrow [C, B, A, C]\ (0, \infty; (0, 1, \ldots, 1)) \tag{5.27}\\
A &\longrightarrow [C, C, A', C]\ (1, 1; 1)\ | \tag{5.28}\\
&\ F\ (3, 3; 1) \tag{5.29}\\
B &\longrightarrow [C, B', C, C]\ (\infty, \infty; 1)\ | \tag{5.30}\\
&\ F\ (4, 4; 1) \tag{5.31}\\
A' &\longrightarrow A\ (2, 2; 1) \tag{5.32}\\
B' &\longrightarrow B\ (\infty, \infty; 1) \tag{5.33}\\
F &\longrightarrow C\ (5, \infty; 0) \tag{5.34}\\
C &\longrightarrow [C_{lb}, C_{rb}, C_{lt}, C_{rt}]\ (5, \infty; 0) \tag{5.35}\\
C_{lb} &\longrightarrow [w, b, b, b]\ (5, \infty; 0) \tag{5.36}\\
C_{rb} &\longrightarrow [b, w, b, b]\ (5, \infty; 0) \tag{5.37}\\
C_{lt} &\longrightarrow [b, b, w, b]\ (5, \infty; 0) \tag{5.38}\\
C_{rt} &\longrightarrow [b, b, b, w]\ (5, \infty; 0) \tag{5.39}\\
& \tag{5.40}
\end{align}
$$

Figure 5.13: Bag Context Picture Grammar rules for $G_{\mathsf{Permit\text{-}free}}$

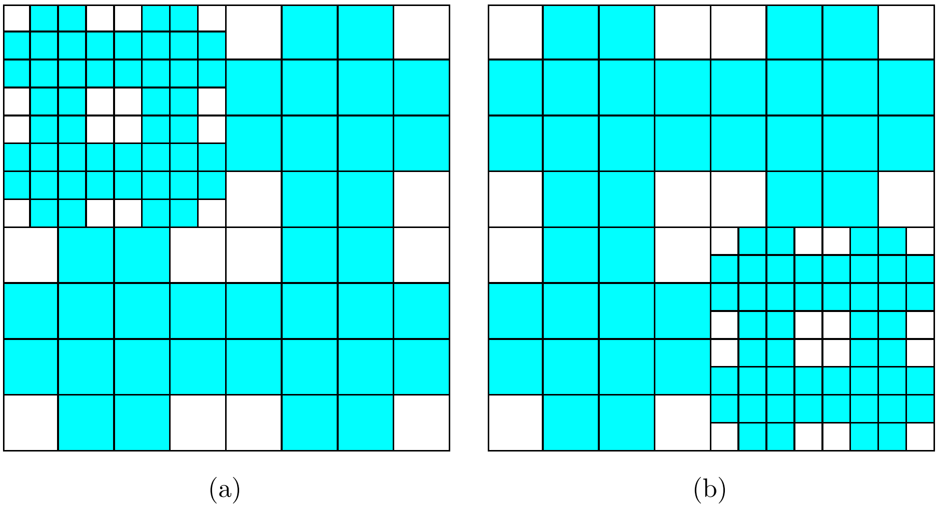
In the next example we create a picture that is exactly the same as the picture in Figure 5.4(c). We do this to show that we can generate the same picture using different grammars with different limits. We can also generate the picture in Figure 5.4(b) using the same method below. Figure 5.4(a) and (d) cannot be generated in the same manner because the grammar assigns C variables in the left bottom and right top regions of the pictures respectively.

Example 5.2.3. Consider $G_{\mathsf{Permit\text{-}free}} = (\{S, A, B, C, A', B', F, C_{lb}, C_{rb}, C_{lt}, C_{rt}, \}, \{w, b\}$,-$R, (S, \theta), \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}, (1, 0, \ldots, 0))$ where R is the set of rules in Figure 5.13.

In this example we show that the picture in Figure 5.4(c) can be generated with this grammar. At first Rules 5.27, 5.28, 5.32 and 5.29 are applied in order and only once. This sequence is forced by the way the lower and upper limit are set up in the rules in Figure 5.13. This process results in the left top part of the picture being refined one more level than the rest of the picture (see Figure 5.14(a)). Next, Rules 5.31, 5.34–5.39 are applied until the process terminates. Rules 5.30 and 5.33 are not applied in this example and this is done by setting the upper and lower limits of these rules to infinity. This is done so that the bottom part of the picture is not refined and as a result for the grammar to generate the picture in Figure 5.4(c). The picture in Figure 5.4(b) can be generated in a similar way by swapping the limits of Rules 5.30 and 5.33 with those of Rules 5.28 and 5.32. However, the pictures in Figure 5.4(a) and (d) cannot be generated by this grammar because of the nature of the grammar. We have Table 5.7, where we show how the bag develops with the application of each rule.

rule	λ	μ	α	$\beta' = \beta + \alpha$
5.27	0	∞	(0, 1, . . . ,1)	1
5.28	1	1	1	2
5.32	2	2	1	3
5.29	3	3	1	4
5.31	4	4	1	5
5.34	5	∞	0	5
5.35	5	∞	0	5
5.36	5	∞	0	5
5.37	5	∞	0	5
5.38	5	∞	0	5
5.39	5	∞	0	5

Table 5.7: Development of bag for the picture in Figure 5.14(a)


(a)

(b)

Figure 5.14: Pictures that are the same to those in Figure 5.4(b) and (c)

$$
\begin{array}{rcll}
S & \longrightarrow & [C, B, A, C]\ ((1, 0, \ldots, 0), (1, 0, \ldots, 0); (1, 2, \ldots, 2)) & (5.41) \\
A & \longrightarrow & [C, C, A', C]\ (\infty, \infty; 1)| & (5.42) \\
 & & F\ (2, 2; 1) & (5.43) \\
B & \longrightarrow & [C, B', C, C]\ (\infty, \infty; 1)| & (5.44) \\
 & & F\ (3, 3; 1) & (5.45) \\
A^{'} & \longrightarrow & A\ (\infty, \infty; 1) & (5.46) \\
B^{'} & \longrightarrow & B\ (\infty, \infty; 1) & (5.47) \\
F & \longrightarrow & [F_{lb}, F_{rb}, F_{lt}, F_{rt}]\ (4, 6; 1) & (5.48) \\
C & \longrightarrow & [C_{lb}, C_{rb}, C_{lt}, C_{rt}]\ (5, \infty; 0) & (5.49) \\
C_{lb} & \longrightarrow & [w, b, b, b]\ (6, \infty; 0) & (5.50) \\
C_{rb} & \longrightarrow & [b, w, b, b]\ (6, \infty; 0) & (5.51) \\
C_{lt} & \longrightarrow & [b, b, w, b]\ (6, \infty; 0) & (5.52) \\
C_{rt} & \longrightarrow & [b, b, b, w]\ (6, \infty; 0) & (5.53) \\
F_{lb} & \longrightarrow & [w, p, p, p]\ (6, \infty; 0) & (5.54) \\
F_{rb} & \longrightarrow & [p, w, p, p]\ (6, \infty; 0) & (5.55) \\
F_{lt} & \longrightarrow & [p, p, w, p](6, \infty; 0) & (5.56) \\
F_{rt} & \longrightarrow & [p, p, p, w]\ (6, \infty; 1) & (5.57)
\end{array}
$$

Figure 5.15: Bag Context Picture Grammar rules for $G_{\mathsf{Permit\text{-}color}}$

In the next example we show how we can use BCPGs to generate pictures that have color distribution. Recall the terminal $'p'$ that represents the color pink, and we add the nonterminals F_{lb}, F_{rb}, F_{lt} and F_{rt} to the grammar and these variables produce the terminal $'p'$ which represents the pink colour.

Example 5.2.4. Let $G_{\mathsf{Permit\text{-}color}} = (\{S, A, B, C, A', B', F, C_{lb}, C_{rb}, C_{lt}, C_{rt}, F_{lb}, F_{rb}, F_{lt}, F_{rt}\},$-$\{w, b, p\}, R, (S, \theta), \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}, (1, 0, \ldots, 0))$ where R is the set of rules in Figure 5.15.

The bag positions 1 to 15 correspond to the non-terminals $S, A, B, C, A', B', F, C_{lb},$-$C_{rb}, C_{lt}, C_{rt}, F_{lb}, F_{rb}, F_{lt}$ and F_{rt} respectively. We want to generate a picture with an equal distribution of blue and pink crosses. We do this by preventing the application of Rules 5.42, 5.44, 5.46 and 5.47 by setting the upper and lower bounds for these rules to infinity. Rule 5.43 and Rule 5.45 are applied only once to get the pink crosses. Then C variables become blue crosses. We end up with the picture in Figure 5.16. The bag develops as shown in Table 5.8. It should be noted that this grammar generates one picture due to the nature of the grammar.

In the next section we look at the RFCPG case. Again we look at how the shape changes from one picture to the next and how the color is distributed.

rule	λ	μ	α	$\beta' = \beta + \alpha$
5.41	(1, 0, . . . ,0)	(1, 0, . . . ,0)	(1, 2, . . . ,2)	2
5.43	2	2	1	3
5.45	3	3	1	4
5.48 (×2)	4	6	1	6
5.49	5	∞	0	6
5.50	6	∞	0	6
5.51	6	∞	0	6
5.52	6	∞	0	6
5.53	6	∞	0	6
5.54	6	∞	0	6
5.55	6	∞	0	6
5.56	6	∞	0	6
5.57	6	∞	0	6

Table 5.8: Development of bag for the picture in Figure 5.16

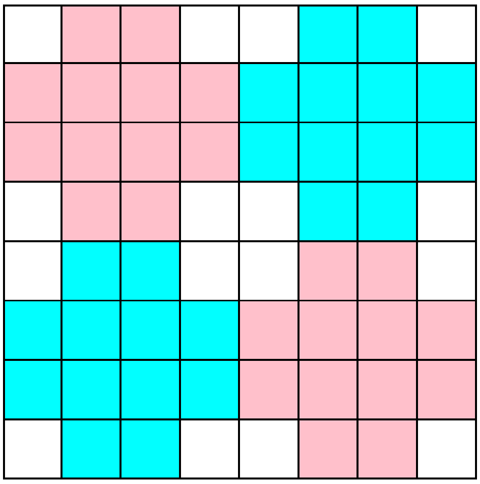


Figure 5.16: Picture with color distribution

5.3 Random Forbidding Context Picture Grammars Case

We use the BCPGs for the forbidding context to show how we generate similar pictures by changing the limits and bag adjustments. In this section we generate pictures where we have dark crosses on a light background with the following structure: every picture has 4^i, where $i \geq 1$, sub-pictures, each of which is created as described in Example 4.3.2 in Chapter 4.

Example 5.3.1. Let $G_{\textsf{Bag-forbid}} = (\{S, T, U, A, B, C, A', B', F, C_{lb}, C_{rb}, C_{lt}, C_{rt}\}, \{w, b\}$,-$R, (S, \theta), \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}, (1, 0, \ldots, 0))$ where R is the set of rules in Figure 5.17.

We associate the variables $S, T, U, A, B, C, A', B', F, C_{lb}, C_{rb}, C_{lt}$ and C_{rt} with the bag positions 1 to 13 respectively. In this example we want to generate pictures where we have two parts of the picture in the third level of refinement while the rest of the crosses are in the second level of refinement. Initially Rule 5.1 and then Rule 5.3 is applied next. The upper limit for Rule 5.3 is set to 6 and this rule is applied four times until all there are no T variables in the pictorial form. Rules 5.2 and 5.4 are prevented from being applied by setting their upper and lower limits to infinity. The limits are set in this way so that this rule is applied after Rule 5.3 only once. The same is done for Rule 5.7 with lower and upper bound at 7. After these rules have been applied then Rules 5.6–5.16 can be applied in any order, until the process terminates. Some of the pictures generated by this grammar are shown in Figure 5.18. We have Table 5.9, where we show how the bag develops with the application of each rule. When we look at Table 5.9 we see that the bag adjustment is 0 from Rules 5.8–5.16 so the bag remains as 8 from this point until the picture generation process ends. The table shows a possible way in which the bag develops for one of the pictures in Figure 5.18.

The next example shows pictures with different levels of refinement in the corners of the sub-pictures.

Example 5.3.2. Consider $G_{\textsf{Refine}} = (\{S, T, U, A, B, C, A', B', F, C_{lb}, C_{rb}, C_{lt}, C_{rt}\}, \{w, b\}, R$,-$(S, \theta), \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}, (1, 0, \ldots, 0))$ where R is the set of rules in Figure 5.19.

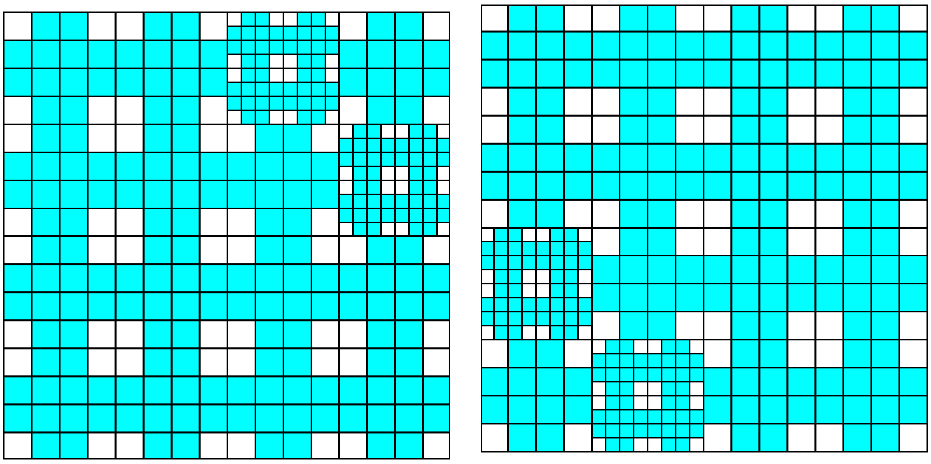
We associate the variables $S, T, U, A, B, C, A', B', F, C_{lb}, C_{rb}, C_{lt}$ and C_{rt} with the bag positions 1 to 13 respectively. In this example we want the corners of some of the four sub-pictures in the main picture as described in Example 4.3.2 in Chapter 4, to have different levels of refinement. This is done first by applying Rule 5.17 four times so that there are four sub-pictures. Then Rule 5.19 is applied next. Rules 5.21 and 5.25 are applied one after the other thirteen times until the bag becomes 19. Then, Rule 5.22 is applied four times so that the top left corners of the sub-pictures are refined. Rules 5.23, 5.24 5.26–5.32, are applied until the process terminates and we get some of the pictures in Figure 5.20. These pictures have different levels of refinement in the corners. Some crosses are in their fourth level of refinement while some crosses are in the first, second

$$
\begin{aligned}
S &\longrightarrow [T,T,T,T]\ (0,\infty;(1,2,\ldots,2)) && (5.1)\\
T &\longrightarrow U\ (\infty,\infty;0)\ | && (5.2)\\
&\ [C,B,A,C]\ (2,6;1) && (5.3)\\
U &\longrightarrow S\ (\infty,\infty;0) && (5.4)\\
A &\longrightarrow [C,C,A',C]\ (6,6;1)\ | && (5.5)\\
&\ F\ (8,\infty;1) && (5.6)\\
B &\longrightarrow [C,B',C,C]\ (7,7;1)\ | && (5.7)\\
&\ F\ (8,\infty;0) && (5.8)\\
A' &\longrightarrow A\ (8,\infty;0) && (5.9)\\
B' &\longrightarrow B\ (8,\infty;0) && (5.10)\\
F &\longrightarrow C\ (8,\infty;0) && (5.11)\\
C &\longrightarrow [C_{lb},C_{rb},C_{lt},C_{rt}]\ (8,\infty;0) && (5.12)\\
C_{lb} &\longrightarrow [w,b,b,b]\ (8,\infty;0) && (5.13)\\
C_{rb} &\longrightarrow [b,w,b,b]\ (8,\infty;0) && (5.14)\\
C_{lt} &\longrightarrow [b,b,w,b]\ (8,\infty;0) && (5.15)\\
C_{rt} &\longrightarrow [b,b,b,w]\ (8,\infty;0) && (5.16)
\end{aligned}
$$

Figure 5.17: Bag Context Picture Grammar rules for $G_{\textsf{Bag-forbid}}$

rule	λ	μ	α	$\beta' = \beta + \alpha$
5.1	0	∞	(1, 2, ...,2)	2
5.3 (×4)	2	6	1	6
5.5	6	6	1	7
5.7	7	7	1	8
5.6	8	∞	1	9
5.8	8	∞	0	9
5.9	8	∞	0	9
5.10	8	∞	0	9
5.11	8	∞	0	9
5.12	8	∞	0	9
5.13	8	∞	0	9
5.14	8	∞	0	9
5.15	8	∞	0	9
5.16	8	∞	0	9

Table 5.9: Development of bag for one of the pictures in Figure 5.18



(a) (b)

Figure 5.18: Two of the Pictures generated by $G_{\textsf{Bag-forbid}}$

$$S \longrightarrow [T, T, T, T]\ (0, \infty; (0, 1, \ldots, 1)) \tag{5.17}$$

$$T \longrightarrow U\ (\infty, \infty; 0)| \tag{5.18}$$

$$[C, B, A, C]\ (1, 4; 1) \tag{5.19}$$

$$U \longrightarrow S\ (\infty, \infty; 0) \tag{5.20}$$

$$A \longrightarrow [C, C, A', C]\ (5, 18; 1)| \tag{5.21}$$

$$F\ (18, 22; 1) \tag{5.22}$$

$$B \longrightarrow [C, B', C, C]\ (23, \infty; 0)| \tag{5.23}$$

$$F\ (23, \infty; 0) \tag{5.24}$$

$$A' \longrightarrow A\ (6, 18; 1) \tag{5.25}$$

$$B' \longrightarrow B\ (23, \infty; 0) \tag{5.26}$$

$$F \longrightarrow C\ (23, \infty; 0) \tag{5.27}$$

$$C \longrightarrow [C_{lb}, C_{rb}, C_{lt}, C_{rt}]\ (23, \infty; 0) \tag{5.28}$$

$$C_{lb} \longrightarrow [w, b, b, b]\ (23, \infty; 0) \tag{5.29}$$

$$C_{rb} \longrightarrow [b, w, b, b]\ (23, \infty; 0) \tag{5.30}$$

$$C_{lt} \longrightarrow [b, b, w, b]\ (23, \infty; 0) \tag{5.31}$$

$$C_{rt} \longrightarrow [b, b, b, w]\ (23, \infty; 0) \tag{5.32}$$

Figure 5.19: Bag Context Picture Grammar rules for $G_{\textsf{Refine}}$

rule	λ	μ	α	$\beta' = \beta + \alpha$
5.17	0	∞	(0, 1, ...,1)	1
5.19 (×4)	1	4	1	5
5.21 (×13)	5	18	1	18
5.25 (×13)	6	18	1	19
5.22 (×4)	19	22	1	23
5.23	23	∞	0	23
5.26	23	∞	0	23
5.24	23	∞	0	23
5.27	23	∞	0	23
5.28	23	∞	0	23
5.29	23	∞	0	23
5.30	23	∞	0	23
5.31	23	∞	0	23
5.32	23	∞	0	23

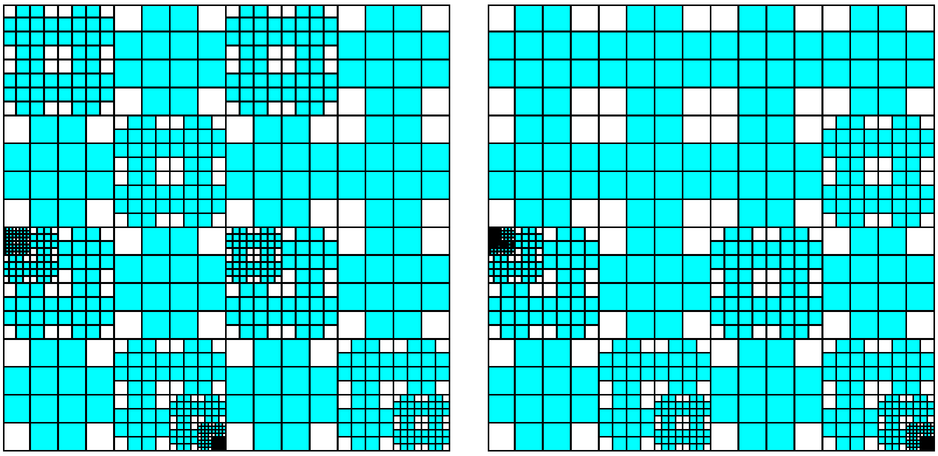
Table 5.10: Development of bag for one of the pictures in Figure 5.20

or third level of refinement. The corners of the pictures can have levels of refinement like fifth or even eighth level of refinement. Table 5.10 shows a possible way in which the bag develops with the application of each rule for one of the pictures Figure 5.20.

In the next two examples, focus is given to color distribution. Recall the terminal $'p'$ which is associated to the colour pink. We also add the variables F_{lb}, F_{rb}, F_{lt} and F_{rt} to the original grammar from the previous chapter.

Example 5.3.3. Let $G_{\textsf{Forbid-color}} = (\{S, T, U, A, B, C, A', B', F, C_{lb}, C_{rb}, C_{lt}, C_{rt}, F_{lb}, F_{rb},$-$F_{lt}, F_{rt}\}, \{w, b, p\}, R, (S, \theta), \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17\}, (1, 0, \ldots, 0))$ where R is the set of rules in Figure 5.21.

We associate the variables $S, T, U, A, B, C, A', B', F, C_{lb}, C_{rb}, C_{lt}, C_{rt}, F_{lb}, F_{rb}, F_{lt}$ and F_{rt} with the bag positions 1 to 17 respectively. In this example, pictures with color distribution are generated. There are two pink crosses and six blue crosses that are in the second level of refinement while the rest of the crosses are in the first level of refinement. To ensure that two pink crosses are in the second level of refinement, Rule 5.37 is applied twice. The lower limit and upper limit are set to 9 and 10 respectively. Rule 5.41 is applied when the bag becomes 11. This is done to make sure that an A variable becomes A' by making sure that all the other rules cannot be applied at this point. Rules 5.34, 5.36, 5.39 and 5.42 are prohibited from being applied by setting their lower limits to infinity. By not applying Rules 5.39 and 5.42, this ensures that the pink crosses produced by the variable B are not refined, as a result some pink crosses are in the second level of refinement while others are in the first level of refinement. This results in the pictures in Figure 5.22 and other pictures generated by this grammar. Table 5.11 shows how the bag develops with the application of each rule. It shows a possible way in which the bag develops for this grammar.



(a) (b)

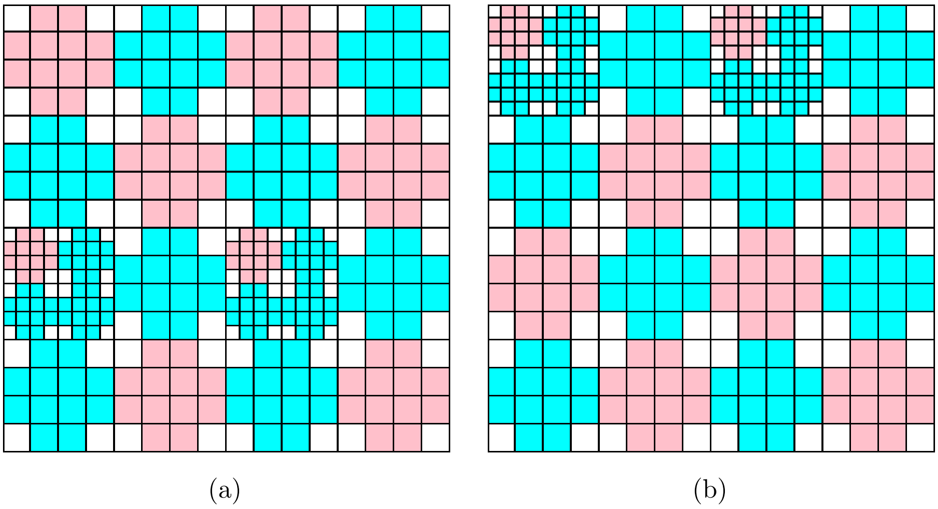
Figure 5.20: Some of the pictures generated by G_{Refine}

$$
\begin{aligned}
S &\longrightarrow [T,T,T,T]\ (0,\infty;(0,1,\ldots,1)) && (5.33)\\
T &\longrightarrow U\ (\infty,\infty;1)\ | && (5.34)\\
&\ [C,B,A,C]\ (1,4;1) && (5.35)\\
U &\longrightarrow S\ (\infty,\infty;1) && (5.36)\\
A &\longrightarrow [C,C,A',C]\ (9,10;1)\ | && (5.37)\\
&\ F\ (13,17;1) && (5.38)\\
B &\longrightarrow [C,B^{'},C,C]\ (\infty,\infty;1)\ | && (5.39)\\
&\ F\ (5,8;1) && (5.40)\\
A' &\longrightarrow A\ (11,12;1) && (5.41)\\
B' &\longrightarrow B\ (\infty,\infty;1) && (5.42)\\
F &\longrightarrow [F_{lb},F_{rb},F_{lt},F_{rt}]\ (17,\infty;0) && (5.43)\\
C &\longrightarrow [C_{lb},C_{rb},C_{lt},C_{rt}]\ (17,\infty;0) && (5.44)\\
C_{lb} &\longrightarrow [w,b,b,b]\ (17,\infty;0) && (5.45)\\
C_{rb} &\longrightarrow [b,w,b,b]\ (17,\infty;0) && (5.46)\\
C_{lt} &\longrightarrow [b,b,w,b]\ (17,\infty;0) && (5.47)\\
C_{rt} &\longrightarrow [b,b,b,w]\ (17,\infty;0) && (5.48)\\
F_{lb} &\longrightarrow [w,p,p,p]\ (17,\infty;0) && (5.49)\\
F_{rb} &\longrightarrow [p,w,p,p]\ (17,\infty;0) && (5.50)\\
F_{lt} &\longrightarrow [p,p,w,p]\ (17,\infty;0) && (5.51)\\
F_{rt} &\longrightarrow [p,p,p,w]\ (17,\infty;0) && (5.52)
\end{aligned}
$$

Figure 5.21: Bag Context Picture Grammar rules for $G_{\mathsf{Forbid\text{-}color}}$

rule	λ	μ	α	$\beta' = \beta + \alpha$
5.33	0	∞	(0, 1, . . . , 1)	1
5.35	1	4	1	5
5.40	5	8	1	9
5.37 (×2)	9	12	1	11
5.41 (×2)	11	12	1	13
5.38 (×4)	13	17	1	17
5.43	17	∞	0	17
5.44	17	∞	0	17
5.45	17	∞	0	17
5.46	17	∞	0	17
5.47	17	∞	0	17
5.48	24	∞	0	17
5.49	24	∞	0	17
5.50	24	∞	0	17
5.51	24	∞	0	17
5.52	24	∞	0	17

Table 5.11: Development of bag for one of the pictures in Figure 5.22


(a)

(b)

Figure 5.22: Some of the pictures generated by $G_{\mathsf{Forbid\text{-}color}}$

$$S \longrightarrow [T,T,T,T]\ (0,\infty;(0,1,\ldots,1)) \tag{5.53}$$

$$T \longrightarrow U\ (\infty,\infty;1)\ | \tag{5.54}$$

$$[C,B,A,C]\ (1,4;1) \tag{5.55}$$

$$U \longrightarrow S\ (\infty,\infty;1) \tag{5.56}$$

$$A \longrightarrow [C,C,A',C]\ (5,9;1)\ | \tag{5.57}$$

$$F\ (13,17;1) \tag{5.58}$$

$$B \longrightarrow [C,B',C,C]\ (17,\infty;0)\ | \tag{5.59}$$

$$F\ (17,\infty;0) \tag{5.60}$$

$$A' \longrightarrow A\ (9,13;1) \tag{5.61}$$

$$B' \longrightarrow B\ (17,\infty;0) \tag{5.62}$$

$$F \longrightarrow [F_{lb},F_{rb},F_{lt},F_{rt}]\ (17,\infty;0) \tag{5.63}$$

$$C \longrightarrow [C_{lb},C_{rb},C_{lt},C_{rt}]\ (17,\infty;0) \tag{5.64}$$

$$C_{lb} \longrightarrow [w,b,b,b]\ (17,\infty;0) \tag{5.65}$$

$$C_{rb} \longrightarrow [b,w,b,b]\ (17,\infty;0) \tag{5.66}$$

$$C_{lt} \longrightarrow [b,b,w,b]\ (17,\infty;0) \tag{5.67}$$

$$C_{rt} \longrightarrow [b,b,b,w]\ (17,\infty;0) \tag{5.68}$$

$$F_{lb} \longrightarrow [w,p,p,p]\ (17,\infty;0) \tag{5.69}$$

$$F_{rb} \longrightarrow [p,w,p,p]\ (17,\infty;0) \tag{5.70}$$

$$F_{lt} \longrightarrow [p,p,w,p]\ (17,\infty;0) \tag{5.71}$$

$$F_{rt} \longrightarrow [p,p,p,w]\ (17,\infty;0) \tag{5.72}$$

Figure 5.23: Bag Context Picture Grammar rules for $G_{\mathsf{Refined\text{-}color}}$

The next example shows the pink crosses in different levels of refinement.

Example 5.3.4. Consider $G_{\mathsf{Refined\text{-}color}} = (\{S,T,U,A,B,C,A',B',F,C_{lb},C_{rb},C_{lt},C_{rt},F_{lb},F_{rb},F_{lt},F_{rt}\},\{w,b,p\},R,(S,\theta),\{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17\},(1,0,\ldots,0))$ where R is the set of rules in Figure 5.23.

We associate the variables $S,T,U,A,B,C,A',B',F,C_{lb},C_{rb},C_{lt},C_{rt},F_{lb},F_{rb},F_{lt}$ and F_{rt} with the bag positions 1 to 17 respectively. The purpose of this example is to generate pictures with pink crosses that have different levels of refinement from the first level to the fourth level of refinement. This is done first by applying Rule 5.53 so that there are four sub-pictures. Rule 5.55 is applied four times so that there are no T variables left in the pictorial form. Rule 5.57 is then applied four times until the bag becomes 9. Rule 5.61 is also applied four times until the bag becomes 13. Next, Rule 5.58 is applied four times so we are done refining the top left corners of the sub-pictures. When one examines the pictures in Figure 5.24, one will find that all the left top pink crosses are in the second level of refinement. Rules 5.59, 5.60, 5.62–5.72 have their upper limits set to infinity and they are applied until the process terminates. This results in the crosses having

rule	λ	μ	α	$\beta' = \beta + \alpha$
5.53	0	∞	(0, 1, . . . , 1)	1
5.55 (×4)	1	4	1	5
5.57 (×4)	5	9	1	9
5.61 (×4)	9	13	1	13
5.58 (×4)	13	17	1	17
5.59	17	∞	0	17
5.62	17	∞	0	17
5.60	17	∞	0	17
5.63	17	∞	0	17
5.64	17	∞	0	17
5.65	17	∞	0	17
5.66	17	∞	0	17
5.67	17	∞	0	17
5.68	17	∞	0	17
5.69	17	∞	0	17
5.70	17	∞	0	17
5.71	17	∞	0	17
5.72	17	∞	0	17

Table 5.12: Development of bag for one of the pictures in Figure 5.24

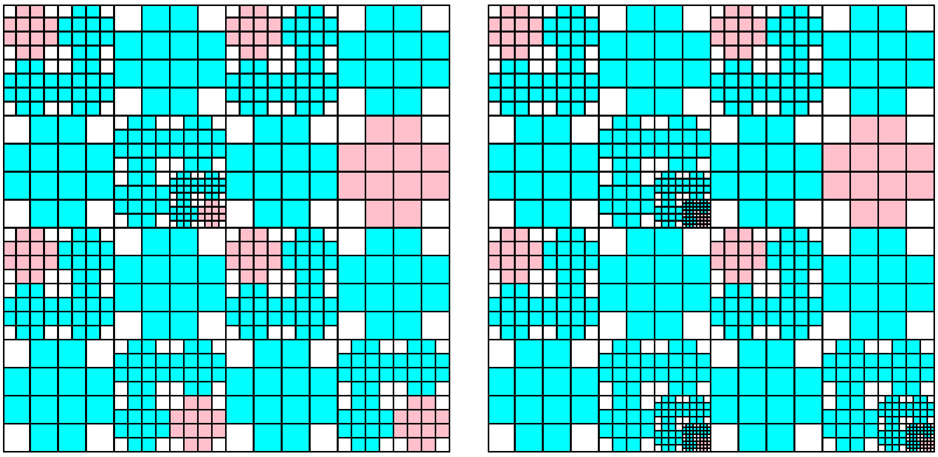
different levels of refinement. Figure 5.24 shows some of the pictures that are generated by this grammar. We see that Figure 5.24(b) has some of its crosses in the fourth level of refinement while other crosses are in the first level. Figure 5.24(a) on the other hand has crosses in first level of refinement while others are in the third level of refinement. The crosses in these pictures have different levels of refinement and hence different sizes. Table 5.12 shows a possible way in which the bag develops with the application of each rule.

5.4 Random Context Picture Grammar Case

In this section we use BCPGs to generate similar pictures with a hollow isosceles right triangle with side 2^i where $i \geq 1$ from Section 4.3.4. In the first two examples we look at how we can change the shape of certain parts of the triangle like the hypotenuse. The last example shows color distribution in these pictures.

Example 5.4.1. Let $G_{\textsf{Bag-triangle}} = (\{S, T_{lb}, T_{rb}, T_t, T_{ls}, T_b, T_h, T_{lb}', T_{rb}', T_t', T_{ls}', T_b', T_h', F\}$,-$\{w, b\}, R, (S, \theta), \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}, (1, 0, \ldots, 0))$ where R is the set of rules in Figure 5.25.

We associate the variables $S, T_{lb}, T_{rb}, T_t, T_{ls}, T_b, T_h, T_{lb}', T_{rb}', T_t', T_{ls}', T_b', T_h'$ and F with the bag positions 1 to 14. In this example, we want to generate pictures with a hollow isosceles right triangle where the left bottom corner is refined to the third or fourth level of refinement. The explanation given on how the picture is generated with this grammar



(a)

(b)

Figure 5.24: Some of the pictures generated by $G_{\textsf{Refined-color}}$

$$
\begin{aligned}
S &\longrightarrow [T_{lb}, T_{rb}, T_t, w](0, \infty; (0, 1, \ldots,)) && (5.1)\\
T_{lb} &\longrightarrow [T_{lb}', T_b', T_{ls}', w]\ (1, 6; 1)\ | && (5.2)\\
 & \quad\ F\ (5, 7; 1) && (5.3)\\
T_{rb} &\longrightarrow [T_b', T_{rb}', T_h', w]\ (6, 6; 1)\ | && (5.4)\\
 & \quad\ b\ (8, 8; 1) && (5.5)\\
T_t &\longrightarrow [T_{ls}', T_h', T_t', w]\ (9, 9; 1)\ | && (5.6)\\
 & \quad\ b\ (11, 11; 1) && (5.7)\\
T_{ls} &\longrightarrow [T_{ls}', w, T_{ls}', w]\ (\infty, \infty; 1)\ | && (5.8)\\
 & \quad\ b\ (15, 17; 1) && (5.9)\\
T_b &\longrightarrow [T_b', T_b', w, w]\ (\infty, \infty; 1)\ | && (5.10)\\
 & \quad\ b\ (21, 24; 1) && (5.11)\\
T_h &\longrightarrow [w, T_h', T_h', w]\ (\infty, \infty; 1)\ | && (5.12)\\
 & \quad\ b\ (26, 27; 1) && (5.13)\\
T_{lb}' &\longrightarrow T_{lb}\ (2, 6; 1) && (5.14)\\
T_{rb}' &\longrightarrow T_{rb}\ (7, 7; 1) && (5.15)\\
T_t' &\longrightarrow T_t\ (10, 10; 1) && (5.16)\\
T_{ls}' &\longrightarrow T_{ls}\ (12, 14; 1) && (5.17)\\
T_b' &\longrightarrow T_b\ (18, 20; 1) && (5.18)\\
T_h' &\longrightarrow T_h\ (24, 25; 1) && (5.19)\\
F &\longrightarrow b\ (27, 27; 1) && (5.20)
\end{aligned}
$$

Figure 5.25: Bag Context Picture Grammar rules for $G_{\textsf{Bag-triangle}}$

rule	λ	μ	α	$\beta' = \beta + \alpha$
5.1	0	∞	(0, 1, . . . , 1)	1
5.2 (×3)	1	6	1	4
5.14(×3)	2	6	1	5
5.3	5	7	1	6
5.4	6	6	1	7
5.15	7	7	1	8
5.5	8	8	1	9
5.6	9	9	1	10
5.16	10	10	1	11
5.7	11	11	1	12
5.17 (×3)	12	14	1	15
5.9 (×3)	15	17	1	18
5.18 (×3)	18	20	1	21
5.11 (×2)	21	22	1	23
5.19 (×2)	23	24	1	25
5.13 (×2)	25	26	1	27
5.20	27	27	1	28

Table 5.13: Development of bag for one of the pictures in Figure 5.26

is in relation with the picture in Figure 5.26 (a). First, Rule 5.1 is applied. Then Rules 5.2 and 5.14 are applied one after another until the bag becomes 5. At this point Rule 5.3 is applied in order for the corner to be in its third level of refinement. The rest of the triangle is in its second level of refinement and this is done by applying the rest of the rules either once or twice as shown in Table 5.13. Rules 5.8, 5.10 and 5.12 have lower limits as infinity so that these rules are never applied as a result the rest, the triangle is in the second level of refinement. We see how the bag also develops in this picture in Table 5.13. The result is the picture in Figure 5.26(a). Due to the upper limits of Rules 5.2 and 5.14 being 6, this gives allowance for the corner to be refined to the fourth level as shown in Figure 5.26(b). It should be noted then that sometimes Rule 5.3 may be applied after Rule 5.14 when the bag is 5 and the other rules applied thereafter resulting in the picture generation process to not be completed since there will still be variable T_{lb}' in the pictorial form.

The next example shows pictures of hollow isosceles right triangle that have jagged edged hypotenuses where some squares in the hypotenuse are big and some are small.

Example 5.4.2. Consider $G_{\mathsf{Hypotenuse}} = (\{S, T_{lb}, T_{rb}, T_t, T_{ls}, T_b, T_h, T_{lb}', T_{rb}', T_t', T_{ls}', T_b',$-$T_h', F\}, \{w, b\}, R, (S, \theta), \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}, (1, 0, \dots, 0))$ where R is the set of rules in Figure 5.27.

We associate the variables $S, T_{lb}, T_{rb}, T_t, T_{ls}, T_b, T_h, T_{lb}', T_{rb}', T_t', T_{ls}', T_b', T_h'$ and F with the bag positions 1 to 14. In this picture we want to generate a hollow isosceles right triangle with a jagged hypotenuse while the rest of the triangle is in the second level of refinement. The rules are applied in the order shown in Table 5.14. The upper and lower bounds in the grammar are set in such a way that Rules 5.21–5.27, 5.34, 5.35, 5.36 and

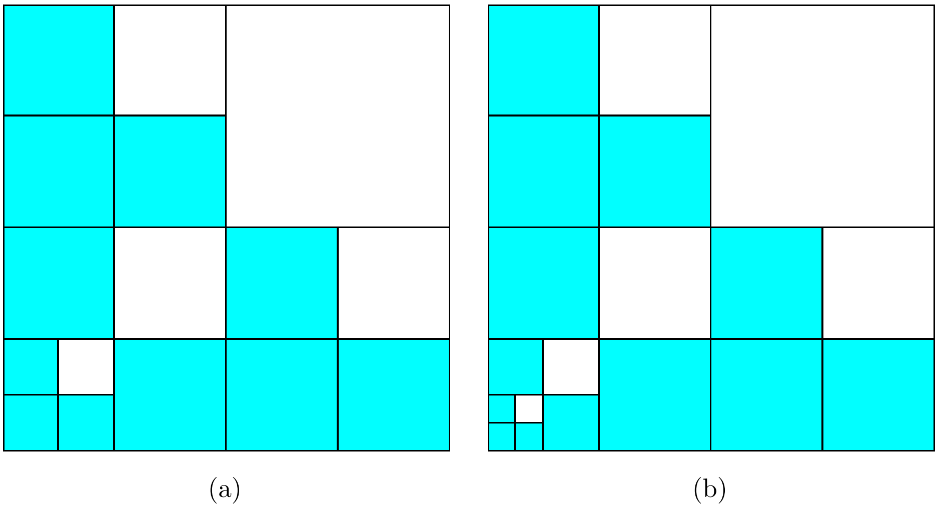

(a) (b)

Figure 5.26: Pictures that are generated by $G_{\textsf{Bag-triangle}}$

$$
\begin{array}{rcll}
S & \longrightarrow & [T_{lb}, T_{rb}, T_t, w]\ (0, \infty; (0, 1, \ldots,)) & (5.21)\\
T_{lb} & \longrightarrow & [T_{lb}', T_b', T_{ls}', w]\ (1, 1; 1)\ | & (5.22)\\
 & & F\ (3, 3; 1) & (5.23)\\
T_{rb} & \longrightarrow & [T_b', T_{rb}', T_h', w]\ (4, 4; 1)\ | & (5.24)\\
 & & b\ (6, 6; 1) & (5.25)\\
T_t & \longrightarrow & [T_{ls}', T_h', T_t', w]\ (7, 7; 1)\ | & (5.26)\\
 & & b\ (9, 9; 1) & (5.27)\\
T_{ls} & \longrightarrow & [T_{ls}', w, T_{ls}', w]\ (\infty, \infty; 1)\ | & (5.28)\\
 & & b\ (12, 13; 1) & (5.29)\\
T_b & \longrightarrow & [T_b', T_b', w, w]\ (\infty, \infty; 1)\ | & (5.30)\\
 & & b\ (17, 18; 1) & (5.31)\\
T_h & \longrightarrow & [w, T_h', T_h', w]\ (19, \infty; 0)\ | & (5.32)\\
 & & b\ (19, \infty; 0) & (5.33)\\
T_{lb}' & \longrightarrow & T_{lb}\ (2, 2; 1) & (5.34)\\
T_{rb}' & \longrightarrow & T_{rb}\ (5, 5; 1) & (5.35)\\
T_t' & \longrightarrow & T_t\ (8, 8; 1) & (5.36)\\
T_{ls}' & \longrightarrow & T_{ls}\ (10, 11; 1) & (5.37)\\
T_b' & \longrightarrow & T_b\ (14, 15; 1) & (5.38)\\
T_h' & \longrightarrow & T_h\ (19, \infty; 0) & (5.39)\\
F & \longrightarrow & b\ (16, 16, 1) & (5.40)
\end{array}
$$

Figure 5.27: Bag Context Picture Grammar rules for $G_{\textsf{Hypotenuse}}$

rule	λ	μ	α	$\beta' = \beta + \alpha$
5.21	0	∞	(0, 1, . . . , 1)	1
5.22	1	1	1	2
5.34	2	2	1	3
5.23	3	3	1	4
5.24	4	4	1	5
5.35	5	5	1	6
5.25	6	6	1	7
5.26	7	7	1	8
5.36	8	8	1	9
5.27	9	9	1	10
5.37 (×2)	10	11	1	12
5.29 (×2)	12	13	1	14
5.38 (×2)	14	15	1	16
5.40	16	16	1	17
5.31 (×2)	17	18	1	19
5.39	19	∞	0	19
5.32	19	∞	0	19
5.33	19	∞	0	19

Table 5.14: Development of bag for one of the pictures in Figure 5.28

5.40 are applied once while Rules 5.29, 5.31, 5.37 and 5.38 are applied twice. Rules 5.32, 5.33 and 5.39 have upper limit set to infinity so that they are applied until the process terminates. We get the pictures with jagged hypotenuses in Figure 5.28.

The next example is focused on color distribution and jagged edges.

Example 5.4.3. Consider $G_{\textsf{Triangle-color}} = (\{S, T_{lb}, T_{rb}, T_t, T_{ls}, T_b, T_h, T_{lb}', T_{rb}', T_t', T_{ls}', T_b',$-$T_h', F\}, \{w, b, p\}, R, (S, \theta), \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}, (1, 0, \ldots, 0))$ where R is the set of rules in Figure 5.29.

We associate the variables $S, T_{lb}, T_{rb}, T_t, T_{ls}, T_b, T_h, T_{lb}', T_{rb}', T_t', T_{ls}', T_b', T_h'$ and F with the bag positions 1 to 14. In this picture we want to generate a hollow isosceles right triangle with color distribution and jagged edges. The bottom part and hypotenuse of all the triangles generated by this grammar has colour pink while the left side and all corners of the triangle are blue. We have all three corners of the pictures in the third level of refinement while parts of the bottom wall are in the second and third level of refinement. The hypotenuse and left side wall of the triangles have jagged edges. When one examines the left side wall of the triangle in Figure 5.30(a), one finds that due to the different sized squares, the triangle's shape looks a little distorted. So the wider the range of levels of refinement the triangle has, the more distorted the shape looks. Table 5.15 shows a possible way the bag developments for the pictures when each rule is applied. We get the pictures in Figure 5.30.

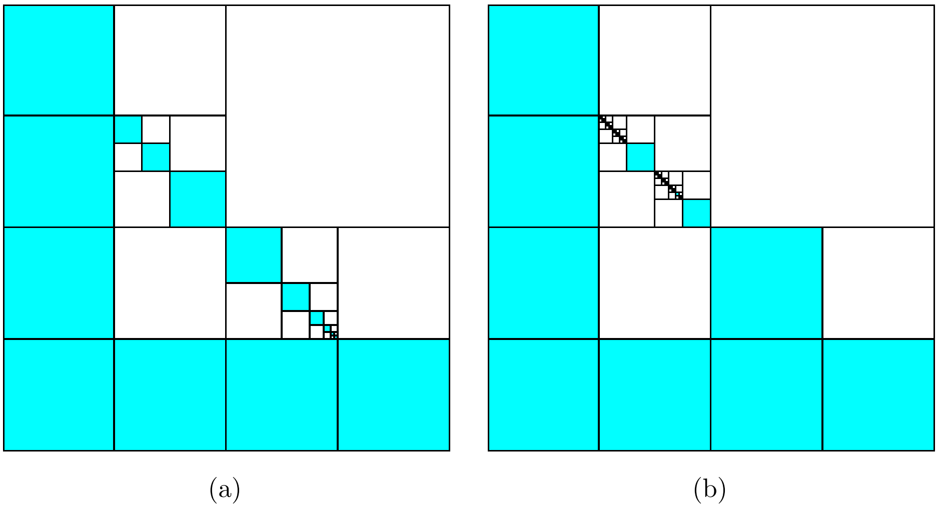


Figure 5.28: Some of the pictures generated by $G_{\mathsf{Hypotenuse}}$

$$S \longrightarrow [T_{lb}, T_{rb}, T_t, w]\ (0, \infty; (0, 1, \ldots,)) \tag{5.41}$$

$$T_{lb} \longrightarrow [T_{lb}', T_b', T_{ls}', w]\ (1, 4; 1)\ | \tag{5.42}$$

$$F\ (5, 5; 1) \tag{5.43}$$

$$T_{rb} \longrightarrow [T_b', T_{rb}', T_h', w]\ (7, 10; 1)\ | \tag{5.44}$$

$$b\ (11, 11; 1) \tag{5.45}$$

$$T_t \longrightarrow [T_{ls}', T_h', T_t', w]\ (12, 15; 1)\ | \tag{5.46}$$

$$p\ (16, 16; 1) \tag{5.47}$$

$$T_{ls} \longrightarrow [T_{ls}', w, T_{ls}', w]\ (20, \infty; 0)\ | \tag{5.48}$$

$$p\ (20, \infty; 0) \tag{5.49}$$

$$T_b \longrightarrow [T_b', T_b', w, w]\ (\infty, \infty; 1)\ | \tag{5.50}$$

$$b(17, 20; 1) \tag{5.51}$$

$$T_h \longrightarrow [w, T_h', T_h', w]\ (20, \infty; 0)\ | \tag{5.52}$$

$$p\ (20, \infty; 0) \tag{5.53}$$

$$T_{lb}' \longrightarrow T_{lb}\ (2, 4; 1) \tag{5.54}$$

$$T_{rb}' \longrightarrow T_{rb}\ (8, 10; 1) \tag{5.55}$$

$$T_t' \longrightarrow T_t\ (13, 15; 1) \tag{5.56}$$

$$T_{ls}' \longrightarrow T_{ls}\ (20, \infty; 0) \tag{5.57}$$

$$T_b' \longrightarrow T_b\ (20, \infty; 0) \tag{5.58}$$

$$T_h' \longrightarrow T_h\ (20, \infty; 0) \tag{5.59}$$

$$F \longrightarrow b\ (6, 6, 1) \tag{5.60}$$

Figure 5.29: Bag Context Picture Grammar rules for $G_{\mathsf{Triangle\text{-}color}}$

rule	λ	μ	α	$\beta' = \beta + \alpha$
5.41	0	∞	(0, 1, ..., 1)	1
5.42 (×4)	1	4	1	4
5.54	2	4	1	5
5.43	5	5	1	6
5.60	6	6	1	7
5.44 (×3)	7	10	1	10
5.55	8	10	1	11
5.45	11	11	1	12
5.46 (×3)	12	15	1	15
5.56 (×2)	13	15	1	16
5.47	16	16	1	17
5.58 (×4)	17	∞	1	21
5.51	20	∞	0	21
5.57	20	∞	0	21
5.48	20	∞	0	21
5.49	20	∞	0	21
5.59	20	∞	0	21
5.52	20	∞	0	21
5.53	20	∞	0	21

Table 5.15: Development of bag for one of the pictures in Figure 5.30

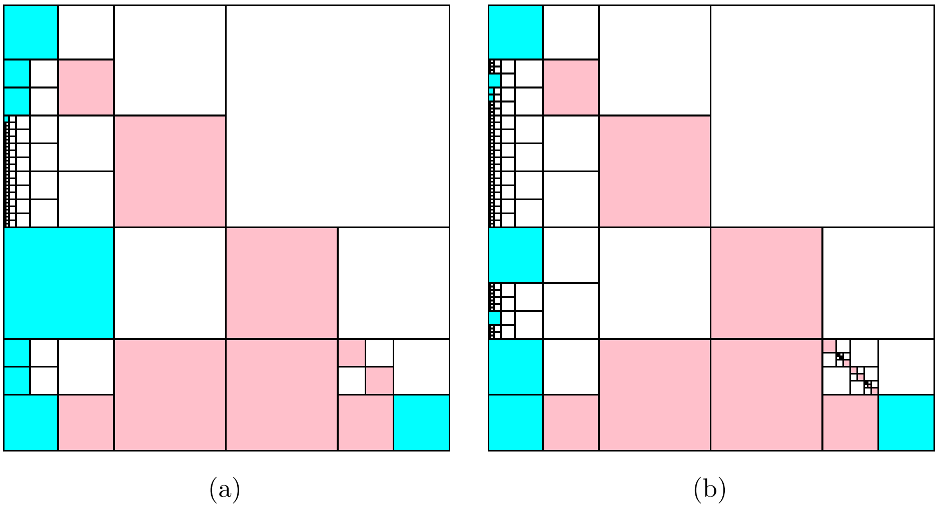


Figure 5.30: Some of the pictures generated by $G_{\textsf{Triangle-color}}$

5.5 Chapter Summary

We have reached the end of this chapter. We have shown that BCPGs can be used to generate similar pictures. We have generated galleries of pictures using four grammars adapted from [1]. We used limits and bag adjustments to show how we can use one grammar to generate different pictures. In some cases we included color in order to drive a point across. For example, we used color to show how we could count the number of crosses in a picture with a certain color.

Bag context helps to generate similar pictures in that it gives us control on the rules to be applied. We may apply the rules at random or force the rules to be applied in a certain order to force a certain sequence. One way to force the rules to be applied in sequence is to set up the lower limits in ascending order to determine which rule gets applied after the other. In this manner we have a certain level of control of how the pictures will turn out. We can also prohibit some rules from being applied by setting the lower limit high enough so that the rule is never applied.

We have shown we can distribute color in pictures. We count the number of variables that produce a particular color by counting the number of times the rule that produces those variables is applied. We also create different levels of refinement in pictures in the same way by counting the number of times particular rules are applied. In the next chapter we give a general conclusion of this study and we discuss future work that may be conducted in this field concerning bag context.

Chapter 6

Conclusion and Future Work

6.1 General Conclusion

The main objectives of this research were to define BCPGs and use them to generate similar images using picture sets generated by RCPGs and their three subclasses. We also defined a lemma that converts RCPGs into equivalent BCPGs.

We defined BCPGs based on the definition of BCTGs and RCPGs. We gave an introduction in Chapter 1 which summarised the work to be conducted in this paper. Chapter 2 was on related work that has already been conducted in this field. We also gave definitions and notation that were used in the rest of this paper. Chapter 3 presented the methodology used to conduct this research. In Chapter 4 we gave the definition of BCPGs. We gave a lemma that converts RCPGs into equivalent BCPGs and showed how a picture is generated using BCPGs. In Chapter 5 we used BCPGs to generate similar pictures using four different grammars from [1]. For each grammar we generated picture sets, of which some galleries had pictures which had different levels of refinement and the other galleries showed color distribution. We saw that we can generate similar pictures with BCPGs.

Bag context was defined as a device for regulated rewriting and we saw how we used this device to control rule application in grammars in order to generate pictures such that they are similar but not identical. We achieved the main objectives of the research and next we discuss future work.

6.2 Future Work

This section looks at future work that could be conducted in terms of bag context in general.

- Explore defining bag context string grammars and how they can be used.
- Use bag context with a wider range of picture grammars such as the L-Systems discussed in Chapter 2 and see how we can regulate rewriting in other picture grammars with bag context.

- Measure similarity of pictures using a similarity measures such as the Content Based Image Retrieval system to check for similarity in pictures generated using BCPGs.
- In this dissertation we used the lemma to translate a RCPG into a BCPG . As it was observed in Chapter 5, some rules were not applied when generating some of the picture sets. We also had instances where there were too many variables which were not used due to some rules not being applied. We can create specific rules to generate BCPGs with fewer variables.

We conclude this research. We have defined BCPGs and used them to generate similar images.

Bibliography

[1] S. Ewert, “Random Context Picture Grammars,” Ph.D. dissertation, University of Stellenbosch, 1999.

[2] Wikipedia, “Koch snowflake,” 2015, accessed: 2015-10-24. [Online]. Available: https://en.wikipedia.org/wiki/Koch_snowflake

[3] R. Designs, “Simple dots rangoli for festival season.” [Online]. Available: http://rangolidesign.in/simple-dots-rangoli-festival-seasonnote={Accessed:2015-10-24}

[4] A. Samal, B. Peterson, and D. J. Holliday, “Recognizing plants using stochastic L-systems,” in *Internation Conference of Image Processing-94*, vol. 1. IEEE, 1994, pp. 183–187.

[5] F. Drewes, C. du Toit, S. Ewert, B. van der Merwe, and A. van der Walt, “Bag context tree grammars,” *Fundamenta Informaticae*, vol. 86, no. 4, pp. 459–480, 2006.

[6] A. van der Walt and S. Ewert, “A property of random context picture grammars,” *Theoretical Computer Science*, vol. 301, no. 1, pp. 313–320, 2003.

[7] H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi, “Tree automata techniques and applications,” Available on: http://www.grappa.univ-lille3.fr/tata, accessed: 2015-12-6.

[8] M. Oberguggenberger and A. Ostermann, “Fractals and L-systems,” in *Analysis for Computer Scientists.* Springer, 2011, pp. 23–47.

[9] P. Prusinkiewicz and J. Hanan, *Lindenmayer Systems, Fractals, and Plants.* Springer Science & Business Media, 2013, vol. 79.

[10] P. Prusinkiewicz, J. Hanan, M. Hammel, and R. Mech, “L-systems: from the theory to visual models of plants,” in *Proceedings of the 2nd CSIRO Symposium on Computational Challenges in Life Sciences*, vol. 3. Citeseer, 1996, pp. 1–32.

[11] C. R. Cook and P. S.-P. Wang, “A Chomsky hierarchy of isotonic array grammars and languages,” *Computer Graphics and Image Processing*, vol. 8, no. 1, pp. 144–152, 1978.

[12] K. Subramanian, S. Hemalatha, and C. S. H. Nagore, “On image generation by sequential/parallel array grammars,” in *2007 International Conference on Signal Processing, Communications and Networking*, 2007.

[13] G. Siromoney and R. Siromoney, "Radial grammars and radial L-systems," *Computer Graphics and Image Processing*, vol. 4, no. 4, pp. 361–374, 1975.