The Combinatorial RNA Design Problem for Binary Trees

by

Tara Joyce Petrie

B.Sc., University of Regina, 2014

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Mathematics
Faculty of Science

© Tara Joyce Petrie 2017
SIMON FRASER UNIVERSITY
Spring 2017

All rights reserved.
However, in accordance with the *Copyright Act of Canada*, this work may be reproduced without authorization under the conditions for "Fair Dealing." Therefore, limited reproduction of this work for the purposes of private study, research, education, satire, parody, criticism, review and news reporting is likely to be in accordance with the law, particularly if cited appropriately.

Approval

Name:	**Tara Joyce Petrie**
Degree:	**Master of Science (Mathematics)**
Title:	***The Combinatorial RNA Design Problem for Binary Trees***
Examining Committee:	**Chair:** Nathan Ilten Assistant Professor

Jonathan Jedwab
Senior Supervisor
Professor _____

Marni Mishna
Supervisor
Associate Professor _____

Ladislav Stacho
Internal Examiner
Associate Professor _____

Date Defended: 13 April 2017

Abstract

The nucleotides adenine, uracil, guanine, and cytosine are the building blocks of ribonucleic acid (RNA). Certain nucleotides can pair, creating folds in the RNA sequence known as the secondary structure. The stability of the secondary structure increases with the number of pairings, but there are typically many foldings that achieve the maximum number of pairings. The combinatorial RNA design problem is to find a design for a target secondary structure (a sequence which can achieve its maximum number of pairings only by folding into this structure), or else to show that no design exists for this structure. A design is known for a class of secondary structures in which all nucleotides are paired, but a structure in which even one nucleotide is unpaired need not admit a design. We prove constructively that there is an infinite class of secondary structures containing unpaired nucleotides and admitting a design.

Keywords: Secondary structure; combinatorial RNA design problem; nucleotide; binary tree

Acknowledgements

First I would like to acknowledge that my thesis was researched and written on the unceded Aboriginal territories of the Coast Salish people, including the Musqueam, Tsleil-Waututh, and Squamish First Nations.

Thank you to my family back in Saskatchewan and my little Vancouver family, as well as my many other dear friends scattered around the world. Thank you to my favourite baristas at Sciué.

I also want to thank Dr. Ladislav Stacho for introducing me to this problem, Dr. Yann Ponty for passing along some much-needed clarifications (especially on the biological background), and Dr. Marni Mishna for providing valuable advice on writing mathematics. Thank you to (future Dr.) Sam Simon for numerous great ideas during our group meetings and (future Dr.) Stefan Hannie for wonderful ideas about unsaturated trees. A special thank you to my supervisor, Dr. Jonathan Jedwab, who has put up with my cringe-worthy grammar mistakes and supported my learning throughout the last two and a half years.

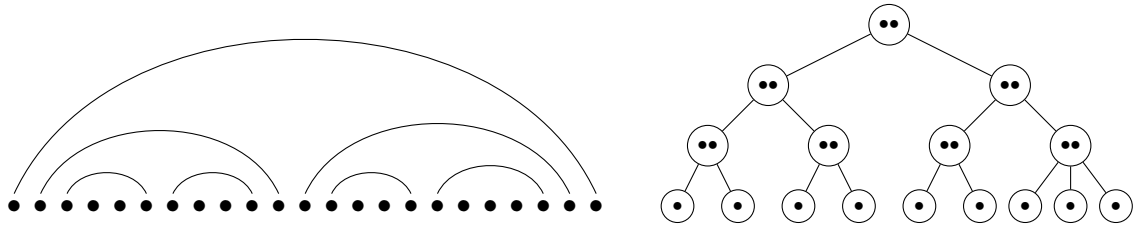
Thank you to NSERC and SFU Department of Mathematics for funding my research.

Table of Contents

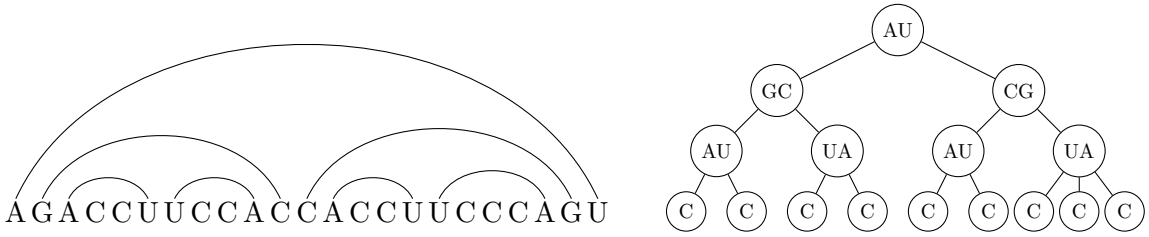
List of Figures

Outline of Main Problem

Step 1: Input a target secondary structure M or its corresponding tree T:



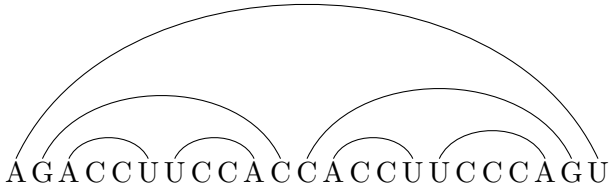
Step 2: Strategically label the secondary structure, or equivalently label the tree with nucleotides to obtain τ:



Step 3: Retain only the sequence S of nucleotides given by the labelling:

A G A C C U U C C A C C A C C U U C C C A G U

Step 4: Test the labelling: does this sequence have a unique maximum-size arc set which equals M?



Prior result: When the tree T in Step 1 is saturated, each vertex having at most three children, there is a labelling in Step 2 which passes the test in Step 4 [14].

Main problem: What if the tree in Step 1 is unsaturated?

Chapter 1

Introduction

A strength of the combinatorial approach is the ability to identify the structure of collections of objects. This way of studying can actually provide us with a deeper understanding of the natural world, and even very complex biological objects. In particular, objects from genomics are well suited to combinatorial analysis as their large scale structure is not immediately evident.

Small scale versions of these objects might appear simple; however, they do not occur at such small scales in nature. Ribonucleic acid (RNA) is made up of many thousands of nucleotides, all of which can interact with one another with some probability. The benefit of using combinatorics to study biological objects is that it provides insight into simplified biological problems which can then be extended to better model the real world. In this thesis we will use properties of trees to establish results on the structure of synthetic RNA.

1.1 RNA background

Deoxyribonucleic acid (DNA) is the storehouse for genetic information, while ribonucleic acid (RNA) is the connection between genetic information and proteins. RNA reads the information stored in DNA and is responsible for protein synthesis. Nucleotides, or nucleic acids, are the building blocks of both DNA and RNA. The nucleotides of DNA are formed from the nitrogenous bases adenine, thymine, guanine, and cytosine, whereas those of RNA are formed from adenine (A), uracil (U), guanine (G), and cytosine (C) [14].

DNA has a double helix structure and is found only in the nucleus of the cell, whereas RNA is a single strand and is found almost everywhere in the cell. Associated with each RNA nucleotide (A, U, G, C) is a sugar and a phosphate [14]. The backbone of the RNA sequence is an alternating string of sugars and phosphates to each of which is attached one nucleotide, as demonstrated in Figure 1.1.

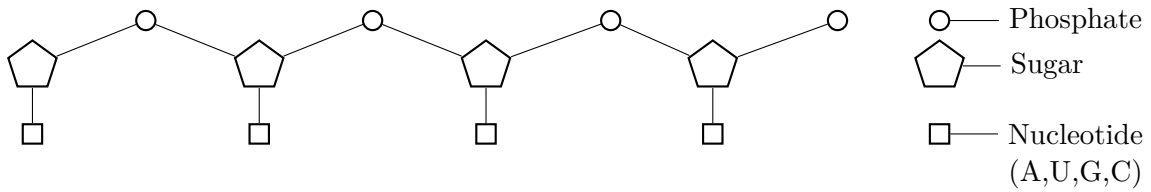


Figure 1.1: RNA primary structure

We represent an RNA sequence with four different structures, each of which provides a different level of understanding. The first is the *primary structure*, a sequence of nucleotides each covalently bonded to the next. A nucleotide can also form hydrogen bonds with another nucleotide further along the sequence. Such a pair of nucleotides is called a *base pair*. More specifically, G can form a base pair with C, and A with U. There are some instances where G can also pair with U, and this is considered in Section 2.5.

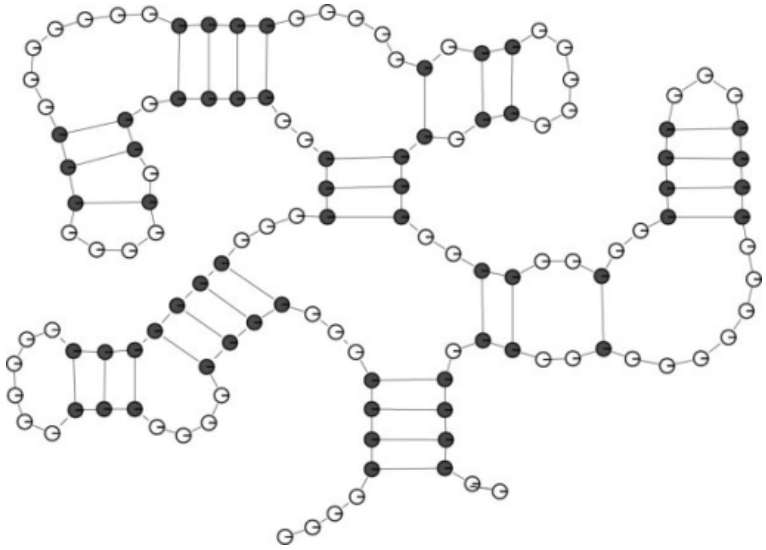


Figure 1.2: An RNA secondary structure, by M.E. Nebel, A. Scheid, and F. Weinberg. Department of Computer Science, University of Kaiserslautern, April 4, 2017, retrieved from https://openi.nlm.nih.gov/ Copyright Policy - open-access by Open-i [11].

The formation of base pairs via hydrogen bonds creates folds and twists in the sequence of nucleotides, giving the *secondary structure* [5], an example of which can be seen in Figure 1.2 [11]. The main result of this thesis constructs RNA sequences for an infinite class of secondary structures. Although we now describe two further representations for RNA, it is the secondary structure that is central to this thesis.

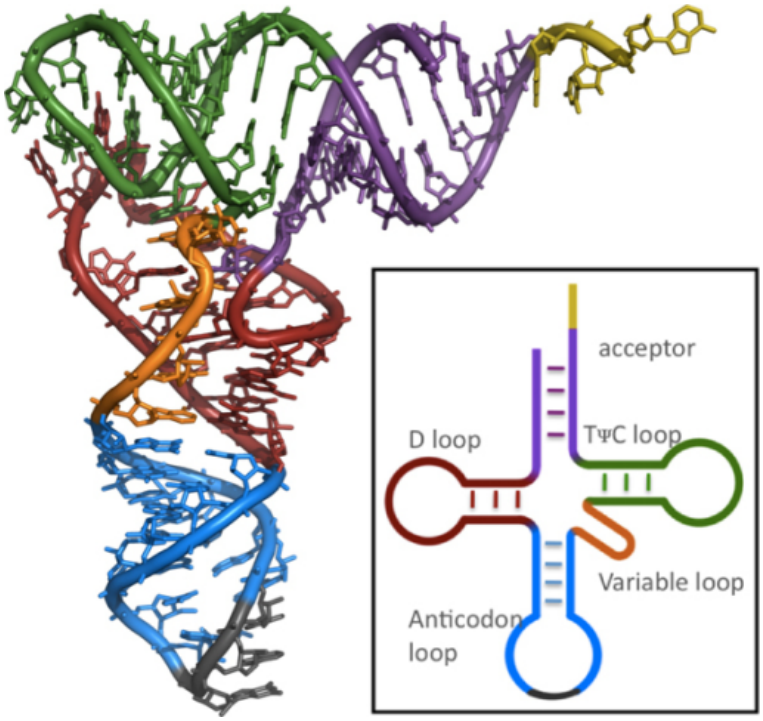


Figure 1.3: The general tRNA tertiary structure (and the secondary structure in the box), by P. Shareghi, Y. Wang, R. Malmberg, and L. Cai. Department of Computer Science, University of Georgia, Athens, GA 30602, USA, April 15, 2017, retrieved from https://openi.nlm.nih.gov/ Copyright Policy - open-access by Open-i [18].

Triples of consecutive nucleotides in the sequence form amino acids, and the sequence can bend further depending on hydrophobic and hydrophilic amino acids (those that are repelled from and attracted to water, respectively). So what was represented as a one-dimensional sequence in the primary structure takes on a complex three-dimensional shape known as the *tertiary structure*, an example of which can be seen in Figure 1.3 [18].

A single chain of nucleotides can make up a whole enzyme or protein. Proteins like hemoglobin are often made up of several different interacting chains, giving the *quaternary structure* of RNA [14]. In this thesis we are concerned only with the primary and secondary structures of RNA, specifically the possible base pairs and resulting representation in two dimensions. These two structures (primary and secondary) are a source of many challenging problems of combinatorial arrangement.

1.2 Two problems involving primary and secondary structures

In general, there are many ways in which a sequence of nucleotides can fold in two dimensions, and the folding affects the function of the RNA sequence [4]. In the field of

synthetic biology, RNA sequences are designed to have particular, and sometimes novel, biological functions [7]. Furthermore, for a given primary structure, the stability of the RNA secondary structure increases with the number of base pairs, and a structure with the maximum number of base pairs is more likely to be adopted as the secondary structure [20]. Nussinov and Jacobson showed that, given an RNA sequence with n nucleotides, a secondary structure with the maximum number of base pairs can be predicted in order n^3 time [12]. Secondary structure prediction gives an idea of how a cell might be organized [23]. It has also been examined as an optimization problem which focuses on minimizing energy. Algorithms for secondary structure prediction are presented in [10, 16, 23].

In the reverse direction, and of particular interest in synthetic biology, we have the *combinatorial RNA design problem*: given a target secondary structure, find an RNA sequence which can achieve its maximum number of base pairs only by folding into the specified secondary structure, or else show that no such sequence exists [5]. The combinatorial RNA design problem turns out to be much harder than secondary structure prediction.

As stated previously, the only three allowable base pair types are {G,C}, {A,U}, {G,U}, and the third of these occurs less frequently. The Nussinov-Jacobson model assigns a negative energy score α, β, γ, respectively, to each of these base pair types (where $\alpha < \gamma$ and $\beta < \gamma$), and seeks RNA sequences which minimize the total energy score [5]. Base pair types other than {G,C}, {A,U}, {G,U} are excluded by assigning them an energy score of ∞. The special case of this model studied by Reinharz, Ponty, and Waldispühl in [15] excludes the base pair type {G,U} by setting $\gamma = \infty$. The Watson-Crick energy model further specializes to the case $\alpha = \beta = -1$ and $\gamma = \infty$; in this model, minimizing the total energy score is equivalent to maximizing the total number of {G,C} and {A,U} base pairs [5]. This thesis primarily uses the Watson-Crick energy model, although in Section 2.5 we extend our main result to the case $\alpha = \beta = \gamma = -1$; minimizing the total energy score is then equivalent to maximizing the total number of {G,C} and {A,U} and {G,U} pairs.

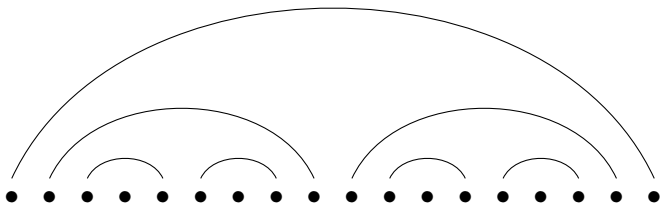
There is a wide range of algorithms for solving the combinatorial RNA design problem [4, 2, 3, 6, 8, 9, 21], including an algorithm presented by Taneda which, for each of 29 target secondary structures, produces 23 different RNA sequences that fold (uniquely and with a maximum number of base pairs) into the target structure [19]. An extensive literature on this problem also includes examinations of its complexity (for example, see [1]), and exhaustive enumeration of potential sequences for target secondary structures [22]. Nonetheless, there is currently no known polynomial-time algorithm for solving the combinatorial RNA design problem, and its complexity remains open [5]. Indeed, a more general problem was shown by Schnall-Levin, Chindelevitch, and Berger to be NP-hard [17].

1.3 Arc representation for secondary structure

We now formalize the main objects of study. The *primary structure* of an RNA strand is an ordered sequence of nucleotides, each represented by the symbols A, U, G, C. For example,

A G A C U U C A C C A C U U C A G U

is the primary structure of an RNA strand. We indicate that two nucleotides are paired by drawing an *arc* from one nucleotide to the other along the top of the primary structure, giving a two-dimensional image known as the *secondary structure*. For example,



is the secondary structure of an RNA strand, where the dots represent positions for nucleotides. Two nucleotides incident with an arc represent a base pair, and those nucleotides not incident with an arc are unpaired. We consider only secondary structures that are *pseudoknot-free*. This means that each nucleotide in the sequence is involved in at most one base pair (so each nucleotide is attached to at most one arc) and base pairs are non-crossing (so arcs do not intersect when represented in two dimensions). The motivation for the non-crossing condition is that arc crossings lead to complex constraints that are often not realizable in three-dimensional space [13].

Given a nucleotide sequence, a secondary structure for that sequence having the maximum number of arcs is a *maximum-size arc set*. A nucleotide sequence that admits a unique maximum-size arc set is a *design*. Given a target secondary structure, the combinatorial RNA design problem is to find a design whose unique maximum-size arc set equals that structure, or else to show that no such design exists [5].

1.4 Tree representation for secondary structure

We study the combinatorial RNA design problem in the context of graph theory by representing the secondary structure with a tree (or more generally a forest). Nucleotides of a base pair, along with the arc joining them, are represented by a vertex with two dots. An unpaired nucleotide is represented by a vertex with one dot. For each two-dot vertex v, the children of v are all vertices formed by nucleotides nested directly beneath the arc corresponding to v in the secondary structure. Each one-dot vertex is always a pendant vertex in the tree representation.

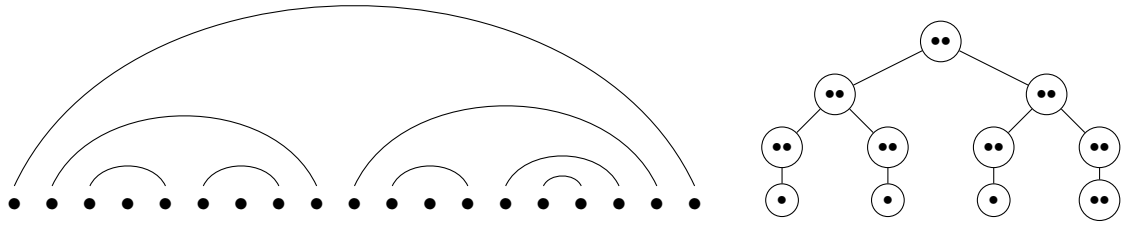


Figure 1.4: Secondary structure (left) and corresponding tree representation (right)

Since we assume the RNA secondary structure is pseudoknot-free, the edges of the tree representation are well-defined. If the first and last nucleotide of the sequence form a base pair, then the graph representation is a rooted tree. (Otherwise it is a forest; in this case, some authors introduce a virtual root in order to produce a tree.) Throughout this thesis, we assume every tree vertex is labelled with exactly one or exactly two dots.

1.5 Saturated and unsaturated secondary structures

A secondary structure in which all nucleotide positions are paired is *saturated*, and otherwise is *unsaturated* [5]. In nature, RNA secondary structures are typically unsaturated.

Figure 1.5: Saturated (left) and unsaturated (right) secondary structures

Each vertex in the tree representation of a secondary structure is labelled with exactly two dots, except possibly the pendant vertices. When all pendant vertices are labelled with two dots this means that every vertex represents a base pair in the associated secondary structure, and hence the structure is saturated. When at least one pendant vertex is labelled with only one dot, then the associated secondary structure has at least one nucleotide not involved in a base pair, and hence is unsaturated. We call the tree saturated or unsaturated in accordance with the associated secondary structure.

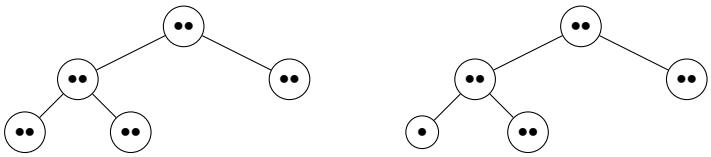


Figure 1.6: Saturated (left) and unsaturated (right) tree representations

Labelling a secondary structure with nucleotides corresponds to assigning a nucleotide to each dot of the associated tree to form a *labelled tree*.

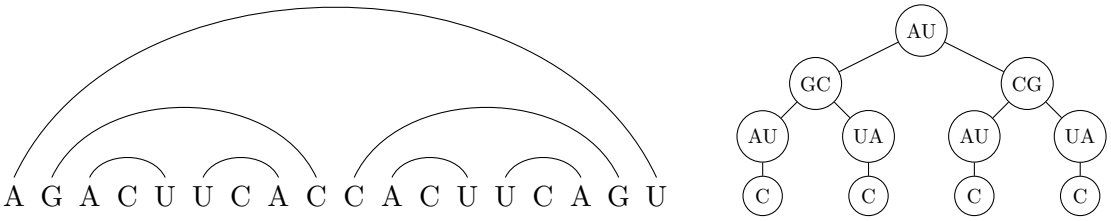


Figure 1.7: Labelled secondary structure (left) and corresponding labelled tree representation (right)

We now demonstrate how to map between the labelled secondary structure and the corresponding labelled tree representation shown in Figure 1.7. Suppose we are given the labelled tree representation. We obtain the nucleotide sequence AGACUUCACCACUUCAGU by traversing the tree as shown in Steps 1 to 18 of Figure 1.8. This traversal method is similar to a pre-order tree traversal, except that all internal vertices are visited twice.

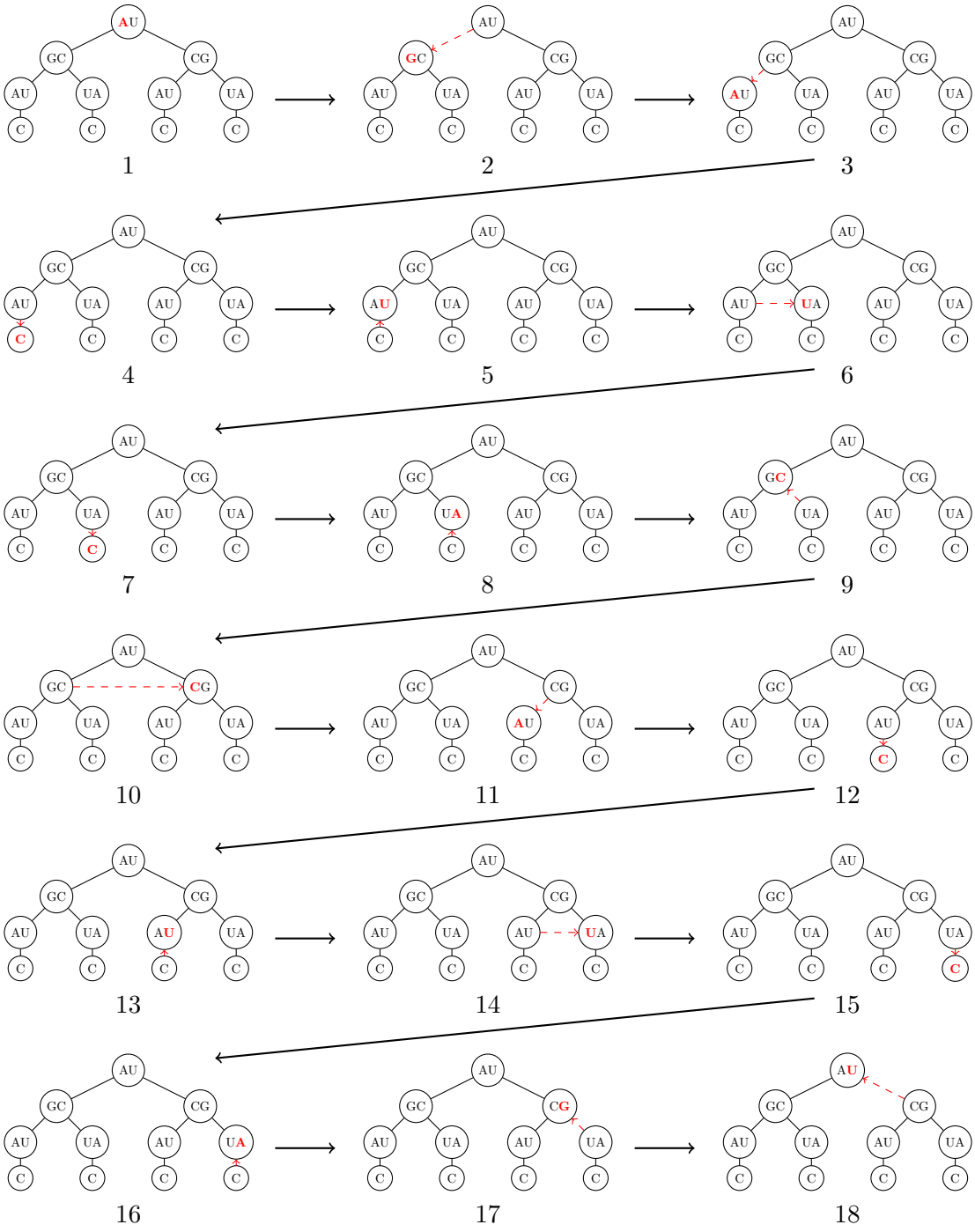
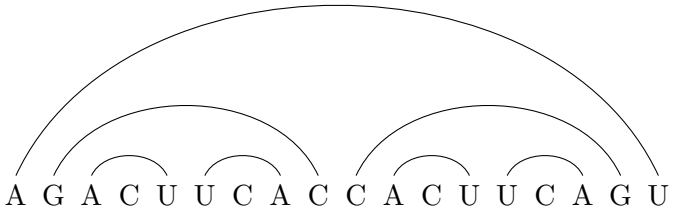


Figure 1.8: Traversing the labelled binary tree of Figure 1.7

Figure 1.8 shows how to obtain the nucleotide sequence AGACUUCACCACUUCAGU from the labelled tree representation of Figure 1.7. The associated arcs are now determined in the following way. When two nucleotides share a vertex in the labelled tree it means that they are a base pair in the labelled secondary structure. We represent this pairing with an arc, giving the following labelled secondary structure, in agreement with Figure 1.7:



Conversely, suppose we are given the labelled secondary structure of Figure 1.7. To transform to the corresponding labelled tree representation, we look at the arcs. The outermost arc will be the root of the tree, with labelling AU. Then its left child is GC and its right child is CG, and so on.

1.6 The contributions of this thesis

Our main result solves the combinatorial RNA design problem for an infinite class of unsaturated secondary structures, by exhibiting a design for these secondary structures (namely, a sequence of nucleotides which can achieve its maximum number of base pairs only by folding into the secondary structure). Showing that the number of base pairs is maximum is easy; however, proving that the sequence folds uniquely into the structure is very difficult, and is the main focus of our proof.

Although this extends a previously known result involving saturated secondary structures to unsaturated secondary structures, the resulting RNA sequences are still much simpler than those that are found in nature. However, it is our hope that the techniques used in the proof can be further generalized to more closely resemble actual biological objects. We raise such extensions as potential future directions in Chapter 5. As the details of the proof of the main result are quite subtle, we illustrate the proof techniques on examples in Chapter 3. This acts as a primer for Chapter 4, in which we prove the main result.

Chapter 2

P-unsaturated Floral Trees are Designable

The goal of this thesis is to extend a previously known result (Theorem 3) involving saturated trees to unsaturated trees. However, we show (in Proposition 4) a very small unsaturated tree which does not admit a design. This motivates us to constrain the number of children each internal vertex can have while still allowing the structure to be unsaturated. We then present our main result (Theorem 6) and two extensions (Corollaries 7 and 8). In Sections 3.1 and 3.2 we give a detailed illustration of the proof method for Theorem 6, as preparation for the formal proof of Chapter 4. Recall that {G,C} and {A,U} are the only possible base pairs considered, except in Section 2.5. We recap some important notions from Chapter 1.

Definition 1 (Design)**.** *An RNA sequence is a design if it admits a unique maximum-size arc set.*

Definition 2 (Designable secondary structure)**.** *An RNA secondary structure is designable if there exists a design whose unique maximum-size arc set equals that structure.*

Combinatorial RNA Design Problem: Given a target secondary structure M, find a design whose (unique) maximum-size arc set equals M, or else show that M does not admit a design.

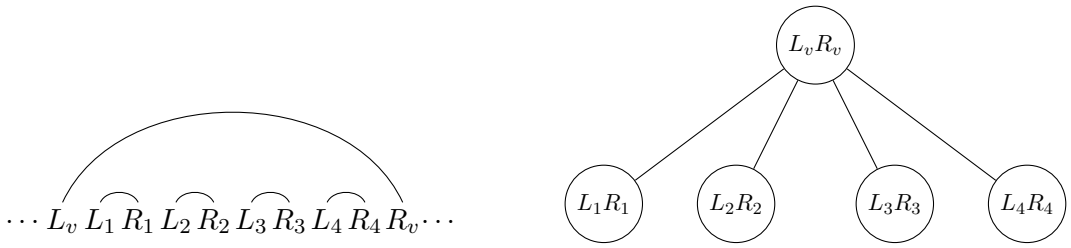
2.1 Known result for saturated trees

Theorem 3. *(Haleš et al. [5]) The secondary structure M corresponding to a rooted saturated tree T is designable if and only if every vertex of T has at most three children.*

We outline the main ideas for the proof of Theorem 3, as given in [5]. We first show that the condition that every vertex of T has at most three children is necessary. Suppose

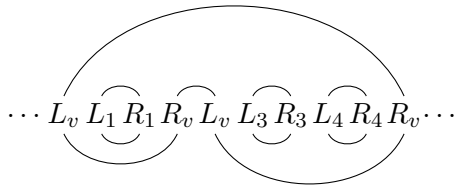
that some vertex v of T has at least four children and, for a contradiction, that there is a design S of nucleotides whose (unique) maximum-size arc set M corresponds to T.

Let the labelling of T corresponding to S label vertex v with L_vR_v and four of the children of v with L_1R_1, L_2R_2, L_3R_3, and L_4R_4. We have the following labelled secondary substructure and its corresponding labelled subtree:

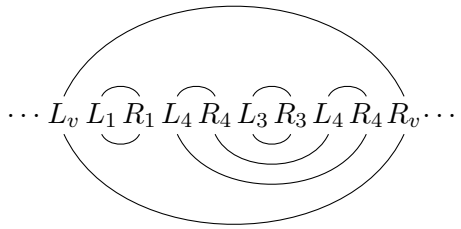


Since there are only four nucleotide types from which to choose, and five labelled vertices, either $R_i = L_v$ (for some $i \in \{1, 2, 3, 4\}$) or $R_i = R_j$ (for some distinct $i, j \in \{1, 2, 3, 4\}$). However, in both cases there are alternative secondary structures compatible with S and having the same number of base pairs as T (hence the maximum number of arcs), giving the required contradiction. In the following pictures we demonstrate the alternative structures (below the sequence) for two illustrative cases.

Case 1: $R_2 = L_v$ (so $L_2 = R_v$ because the only possible base pairs are A with U, and G with C).



Case 2: $R_2 = R_4$ (so $L_2 = L_4$).



We now consider the sufficiency of the condition that every vertex of T has at most three children. The following algorithm is shown in [5] to construct a design whose (unique) maximum-size arc set M corresponds to T; we omit the details of the justification.

Input: A rooted saturated tree T in which every vertex has at most three children.

Step 1. Label the root with some base pair L_0R_0.

Step 2. If there is no labelled vertex whose children are unlabelled, stop.

Step 3. Choose a labelled vertex L_vR_v whose children have not yet been labelled.

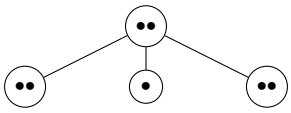
Step 4. Label the children of the vertex with base pairs L_iR_i (where i satisfies $1 \leq i \leq 3$) such that $L_i \neq R_v$ (for all i) and $L_i \neq L_j$ (for all distinct i and j). Go to Step 2.

Output: A labelling of the tree T.

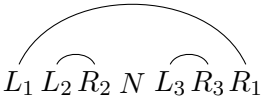
The proof of Theorem 3 given in [5] is constructive (using the algorithm above) and gives many possible nucleotide labellings for the tree. However, the result of Proposition 4 below shows that we cannot remove the constraint 'saturated' from Theorem 3.

2.2 Unsaturated tree that is not designable

Proposition 4. *There is no design whose (unique) maximum-size arc set M corresponds to the unsaturated tree*



Proof. A labelling of M must have the form



where each set $\{L_i, R_i\}$ is one of {G, C} and {A, U} (and, for example, L_1 is not necessarily distinct from L_2 or R_2). Suppose, for a contradiction, that this labelling defines a design whose maximum-size arc set equals M. Then by uniqueness of the arc set, we have

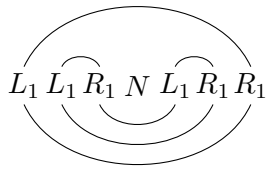
(1) $N \notin \{L_1, R_1, L_2, R_2\}$ and

(2) $L_1 \neq R_2$,

By (1), $\{L_1, R_1\} = \{L_2, R_2\}$ since there are only four choices for N, and then using (2) we obtain $L_1 = L_2$. Repeating the argument on the right hand side of the sequence we find $R_1 = R_3$. So the sequence can be written as

$$L_1\, L_1\, R_1\, N\, L_1\, R_1\, R_1$$

which admits two distinct maximum-size arc sets (shown above and below the sequence)



This gives the required contradiction. □

Proposition 4 shows that not all unsaturated trees are designable structures. However, we shall identify a class of designable unsaturated trees. In order to do so, we introduce the following definition.

2.3 P-unsaturated floral trees

A tree is unsaturated if even just one nucleotide is unpaired; however, in nature, it is more likely that there are many unpaired nucleotides. This motivates the following definition.

Definition 5 (P-unsaturated)**.** *A tree is P-unsaturated if every pendant vertex at the maximum height is assigned exactly one dot and all other vertices exactly two dots.*

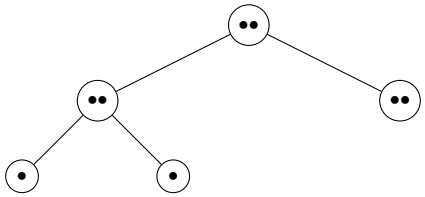


Figure 2.1: P-unsaturated binary tree

A *binary tree* is a tree in which every vertex has at most two children. A *full binary tree* is a tree in which every internal vertex has exactly two children and all pendant vertices are at the same height. We enlarge the class of binary trees and full binary trees in the following way. Call a tree *floral* when removing all vertices at the maximum height leaves a binary tree, and call a tree *full floral* when removing all vertices at the maximum height leaves a full binary tree. For example, the following is a P-unsaturated full floral tree of height 3:

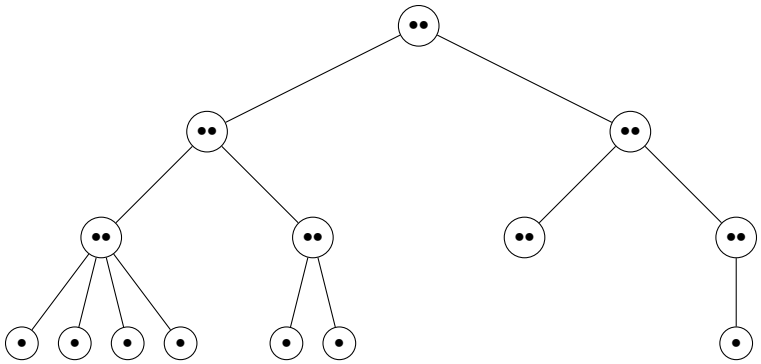


Figure 2.2: P-unsaturated full floral tree

The height of a vertex v in a rooted tree is equal to the length of the path from the root to v. Consider the following assignment of nucleotides to a P-unsaturated full floral tree of height $n+1$. Label all the vertices at height $n+1$ with A if $n+1$ is even, and with C if $n+1$ is odd. Label the remaining vertices at even heights from left to right with alternating AU and UA, and those at odd heights from left to right with alternating GC and CG. Call this labelling the *natural labelling* for a P-unsaturated full floral tree of height $n+1$. For example, the natural labelling of the tree in Figure 2.2 is:

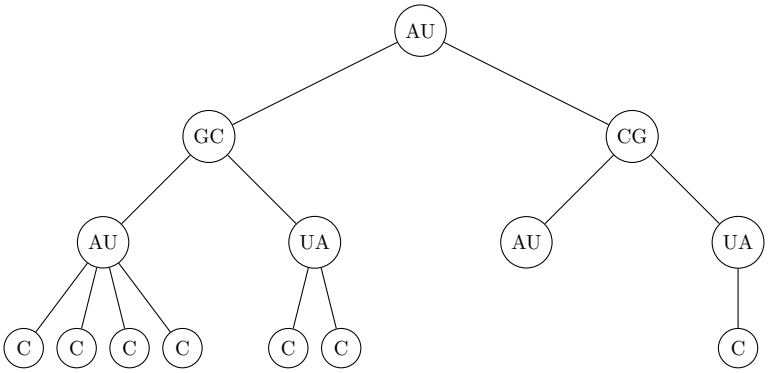


Figure 2.3: Labelled P-unsaturated full floral tree

The main result of this thesis concerns a P-unsaturated full floral tree, rather than the saturated tree of Theorem 3 whose vertices all have at most three children.

2.4 Main result for P-unsaturated full floral trees

Theorem 6. *(Main Result) Given a P-unsaturated full floral tree T, there is a design whose (unique) maximum-size arc set M corresponds to T. Such a design is obtained from the natural labelling of T.*

Although the natural labelling follows a simple pattern, and it is relatively straightforward to show using *ad hoc* methods that Theorem 6 holds when T has small height, finding

a general argument that applies to all tree heights appears to be very delicate. This is because the number of potential nucleotide pairings increases as the tree grows, so demonstrating that the sequence folds uniquely into the target tree structure becomes much more complex.

Let S be the nucleotide sequence corresponding to the natural labelling of a P-unsaturated full floral tree T, as described above. In order to prove Theorem 6, we must establish that S has a unique maximum-size arc set, and that this arc set corresponds to T. In Chapter 4 we will prove Theorem 6 according to the following outline. To show the uniqueness of a maximum-size arc set for S, we show that all nucleotides in S that were originally unpaired (and at height $n+1$) in the labelled tree T must remain unpaired in every maximum-size arc set. Given this constraint, we show how to reduce to the case of a saturated full binary tree. Our main tool is a running difference of strategically chosen subsets of the nucleotides in S, and a key insight is that only the parity of these running differences is important. We then demonstrate that the base pairs in a maximum-size arc set are forced to be those of M.

Note that once we have reduced a P-unsaturated full floral tree to a saturated full binary tree, we could then apply Theorem 3 to show that this forces a maximum-size arc set for S to be M. However, to keep our proof self-contained and demonstrate the versatility of the method of running differences, we instead give our own proof. In this proof we show that for every height j of the reduced tree, nucleotides at height j must all pair with one another, and that this forces a maximum-size arc set for S to be M.

From Theorem 6 we obtain a corollary which removes the restriction that the tree T must be full.

Corollary 7. *Given a P-unsaturated floral tree T, there is a design whose (unique) maximum-size arc set M corresponds to T.*

We give the following structures that provide an illustrative example of the proof of Corollary 7. They can be read in conjunction with the proof of Corollary 7.

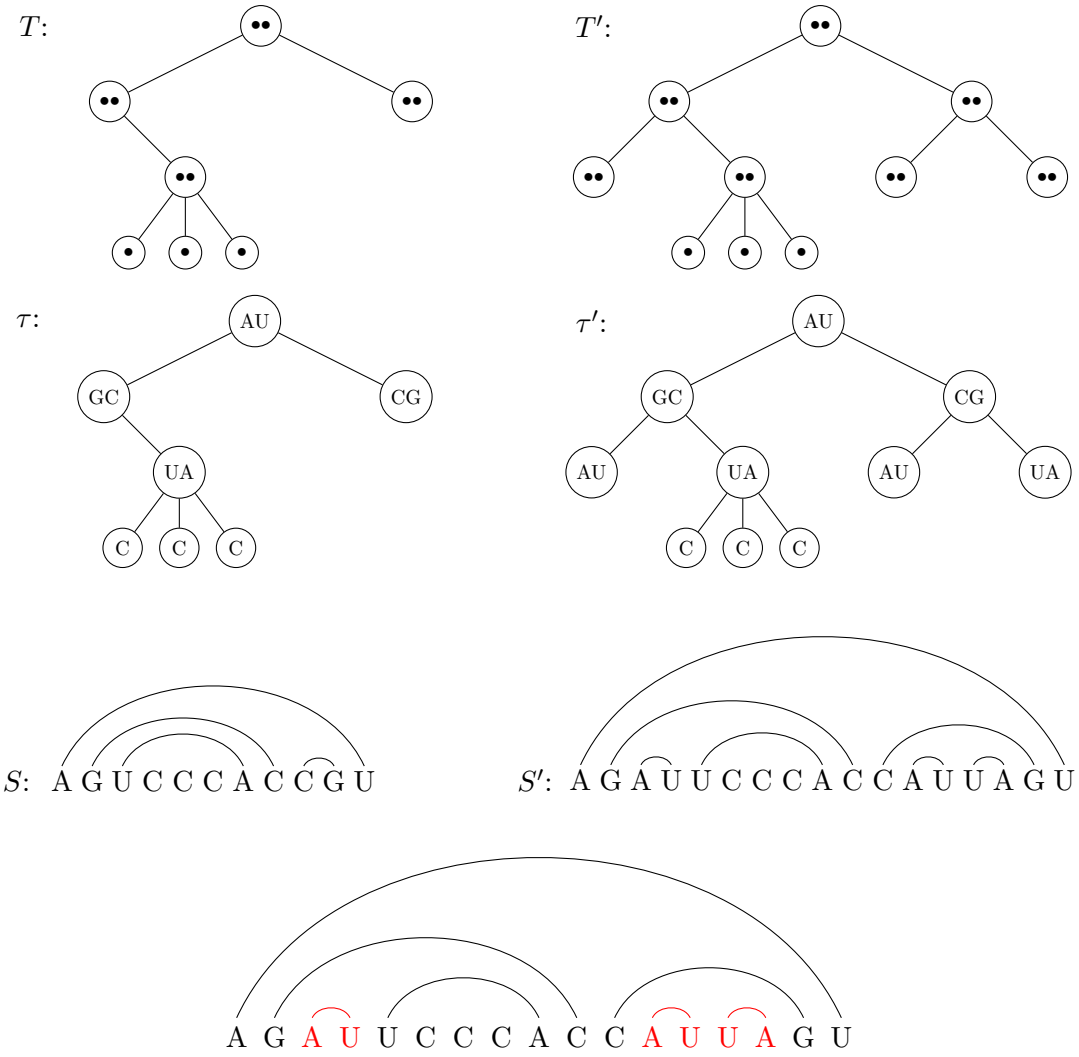


Figure 2.4: Illustrative structures for the proof of Corollary 7

Proof of Corollary 7. Let T be a P-unsaturated floral tree. Let T' be the smallest full floral tree containing T as a subtree; T' is necessarily P-unsaturated. Let τ' be the labelled full floral tree obtained by assigning nucleotides to T' according to the natural labelling. By Theorem 6, the labelled tree τ' yields a design S' of nucleotides whose (unique) maximum-size arc set corresponds to T'. Let τ be the labelled subtree of τ' whose unlabelled version is T, and let S be the associated sequence of nucleotides.

Colour the differences between S and S' and their corresponding arc sets in red, and leave the common parts (corresponding to τ) in black. The red structure is completely nested beneath the arcs of the black structure because it consists of all nucleotides and corresponding arcs that must be pruned from the labelled full floral tree τ' to obtain the labelled floral tree τ.

It follows that the secondary structure corresponding to T is a maximum-size arc set for S: if S admits a larger arc set then that arc set, together with the nested red structure,

gives a larger arc set for S' (contradicting that the secondary structure corresponding to T' is a maximum-size arc set for S').

It also follows that S is a design: if S admits an alternative maximum-size arc set then that arc set, together with the nested red structure, gives an alternative maximum-size arc set for S' (contradicting that S' is a design). □

2.5 Extension to include {G,U} base pairs

As outlined in Section 1.2, Theorem 6 and Corollary 7 assume the Watson-Crick energy model, which assigns to every base pair {G,C} and every base pair {A,U} an energy score of -1, and a score of ∞ to every other base pair [5]. We show in Corollary 8 below that these results can be extended to also allow {G,U} base pairs, by changing the {G,U} energy score from ∞ to -1; the size of an arc set in the combinatorial RNA design problem then becomes the total number of {G,C}, {A,U}, and {G,U} base pairs. The ideas in the proof below were developed with the help of Yann Ponty [13].

Corollary 8. *Given a P-unsaturated floral tree T, and an energy model where each base pair type $\{G,C\}$, $\{A,U\}$, and $\{G,U\}$ is assigned an energy score of -1, there is a design whose (unique) maximum-size arc set M corresponds to T.*

Proof. By Corollary 7, given a P-unsaturated floral tree T, there exists a design S whose (unique) maximum-size arc set M corresponds to T under the Watson-Crick energy model. The number of arcs in M is equal to the number of {G,C} base pairs plus the number of {A,U} base pairs.

There is exactly one nucleotide type in S that can remain unpaired in M (the A nucleotides if the tree height is even, and the C nucleotides if the tree height is odd). So we can count the (maximum) number of arcs for S as the number ζ of Gs plus the number μ of Us in S.

Now allow for {G,U} base pairs. Every {G,U} base pair will use both a G and a U. Hence, any alternative structure to M having at least one {G,U} base pair will have at most

$$(\zeta - 1) + (\mu - 1) + 1 = \zeta + \mu - 1$$

base pairs. Hence, any alternative arc set to M that includes one or more {G,U} base pairs will have fewer arcs than M.

□

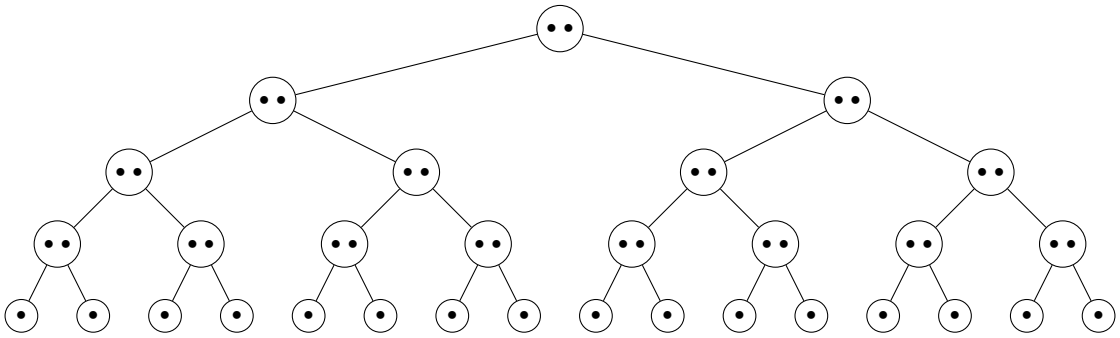
Chapter 3

Illustrations of the proof of Main Result

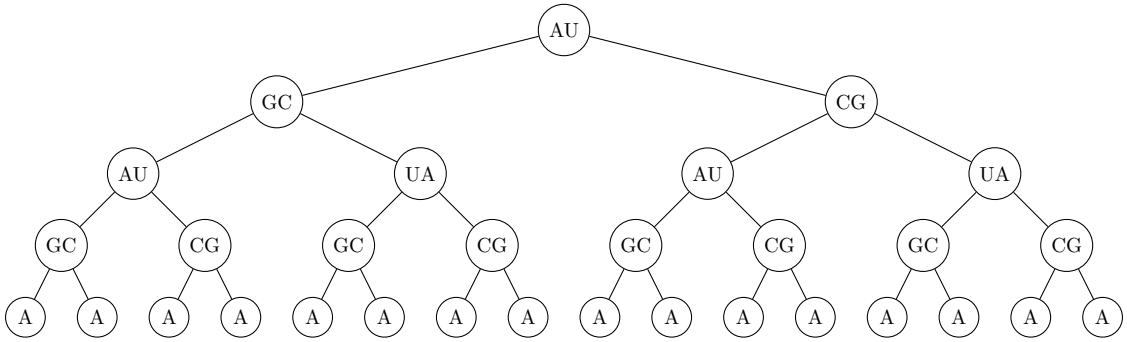
3.1 Illustration for a tree of height 4

We now give a detailed illustration of the proof of Theorem 6 using the example of a P-unsaturated full binary tree T of height 4 as shown in Example 9 below. We use a binary tree in this illustration simply for convenience: the arguments extend to P-unsaturated full floral trees, as will be shown in the proof in Chapter 4.

Example 9.



The natural labelling τ of T is:

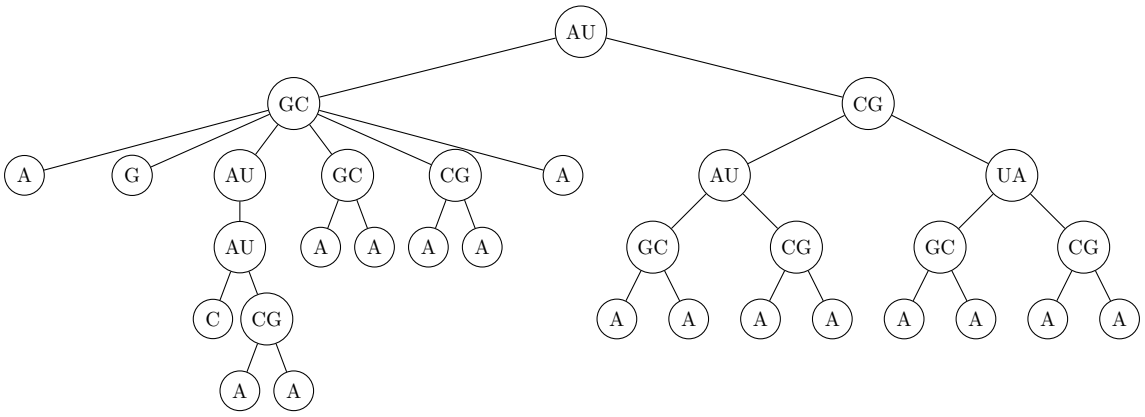


The corresponding sequence S (represented on two lines for convenience of reading) is:

A G A G A A C C A A G U U G A A C C A A G A C

C A G A A C C A A G U U G A A C C A A G A G U

The natural labelling τ of T gives a secondary structure in which the sequence S has fifteen base pairs. The labelled tree shown below gives an alternative secondary structure into which S could fold, but notice that it has only fourteen base pairs.



We shall show that S admits a unique maximum-size arc set, and so is a design, and that this arc set corresponds to T. To explain the proof method, we introduce the following definition:

Definition 10 (Balanced set of nucleotides)**.** *A set of nucleotides is balanced if it comprises either an equal number of Gs and Cs or an equal number of As and Us.*

For example, in Figure 3.1 the subset of G and C nucleotides shown coloured and boxed forms a balanced set $\mathcal{B}$ within the nucleotide sequence S.

0		1		2	2			0	0		1	1		2	2			0	0		1		
A	G	A	G	A	A	C	C	A	A	G	U	U	G	A	A	C	C	A	A	G	A	C	
		-1		0	0			-2	-2		-1	-1		0	0			-2	-2		-1		0
	C	A	G	A	A	C	C	A	A	G	U	U	G	A	A	C	C	A	A	G	A	G	U

Figure 3.1: A balanced set of nucleotides and its corresponding running differences

The number shown above each nucleotide N (not in $\mathcal{B}$) is the difference between the number of Gs and the number of Cs occurring in $\mathcal{B}$ to the left of N, namely the *running difference* with respect to $\mathcal{B}$. We use the corresponding definition for the running difference with respect to a balanced set of A and U nucleotides.

In order to prove Theorem 6 for T, we iteratively define a succession of balanced sets of nucleotides, occurring as disjoint subsets of the sequence S, and use the running difference with respect to these balanced sets to constrain the possible arcs of a maximum-size arc set.

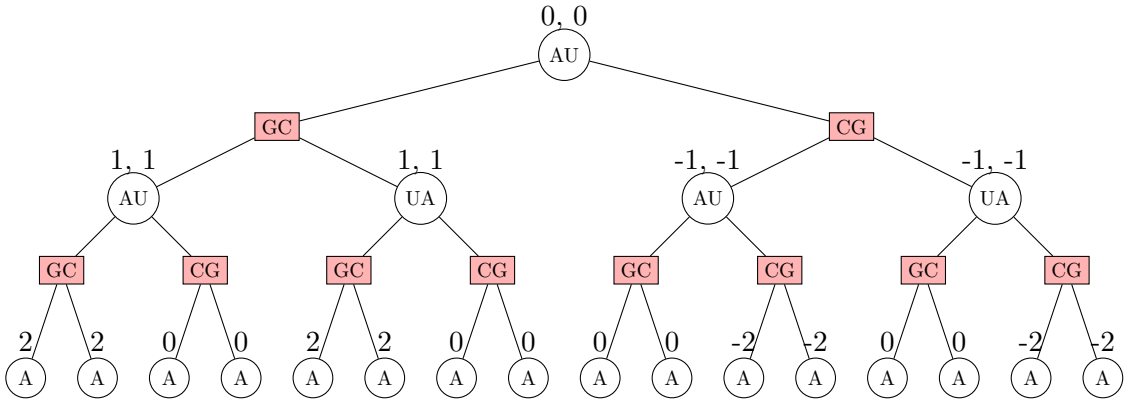
Suppose a maximum-size arc set M is applied to the sequence S. The labelled tree τ from which S is derived shows that S admits an arc set for which only A nucleotides are unpaired; therefore in M all G, C, and U nucleotides of the sequence must be paired.

Iteration 1:

S has an equal number of Gs and Cs, forming a balanced set $\mathcal{B}$ which is boxed and coloured as shown in Figure 3.1. To achieve the maximum number of arcs all elements of $\mathcal{B}$ must pair with one another; this is indicated by the colouring, which will be retained in all subsequent iterations (whereas the boxes are a temporary notation used only in the current iteration). To avoid inducing an arc crossing, every arc in M joining uncoloured nucleotides (As and Us) must therefore enclose equally many of the two types of boxed nucleotides (Gs and Cs). Tag the uncoloured nucleotides with the running difference with respect to $\mathcal{B}$, as in Figure 3.1.

Then the uncoloured nucleotides with a given tag cannot pair with uncoloured nucleotides with a different tag, and cannot pair with coloured nucleotides (because those must all pair with one another). Therefore, the uncoloured nucleotides with a given tag can pair only with one another. This implies the weaker result that uncoloured nucleotides whose tags have the same parity can pair only with one another. (We have not yet shown that the uncoloured nucleotides with a given tag must all pair with one another, nor what the pairings must be.)

As a visual aid, we copy the colouring, boxing, and tagging applied to the nucleotides of S onto the corresponding labelled tree τ: the nucleotides belonging to $\mathcal{B}$ (the Gs and Cs) are coloured and boxed, and the uncoloured nucleotides (the As and Us) are tagged as above. (This does not assume the result we wish to prove, namely that applying a maximum-size arc set to S results in the labelled tree τ.)



As described, all uncoloured nucleotides whose tags are odd (this being the opposite parity to that of the nucleotides at height 4) can pair only with one another. But these nucleotides form a new balanced set $\mathcal{B}'$ (comprising A and U nucleotides), and so must all pair with one another to achieve the maximum number of arcs; this will be indicated by colouring in the next iteration. Remove the tags and boxes but retain the colouring.

Iteration 2:

Box and colour the nucleotides of the new balanced set $\mathcal{B}'$, as shown below.

A G [A] G A A C C A A G [U] [U] G A A C C A A G [A] C

C [A] G A A C C A A G [U] [U] G A A C C A A G [A] G U

To avoid inducing an arc crossing, every arc in M joining two nucleotides not in $\mathcal{B}'$ must enclose an equal number of the two nucleotide types in $\mathcal{B}'$ (boxed As and boxed Us). Tag the uncoloured nucleotides with the running difference with respect to $\mathcal{B}'$.

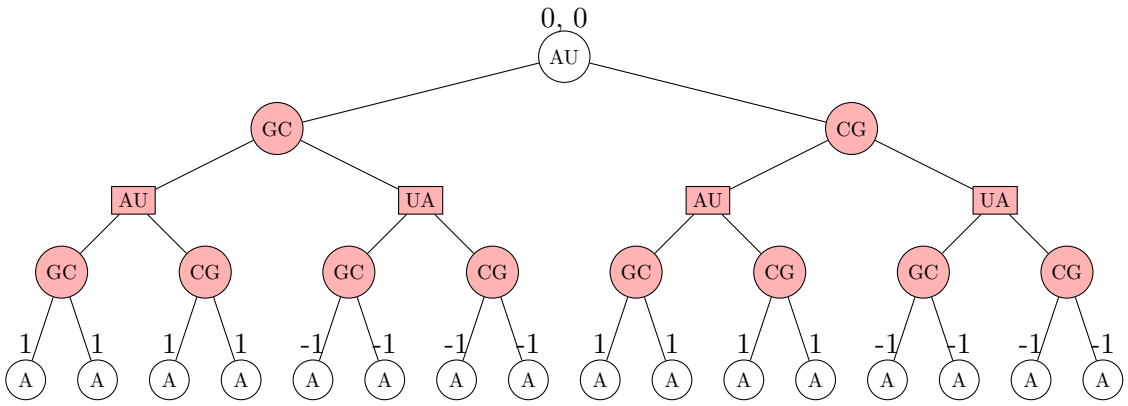
0 1 1 1 1 -1 -1 -1 -1

A G [A] G A A C C A A G [U] [U] G A A C C A A G [A] C

1 1 1 1 -1 -1 -1 -1 0

C [A] G A A C C A A G [U] [U] G A A C C A A G [A] G U

Uncoloured nucleotides whose tags have the same parity can pair only with one another. The corresponding version of τ, again shown only for convenience, is:



All uncoloured nucleotides whose tags are even (this being the opposite parity to that of the nucleotides at height 4) can pair only with one another. There are only two such nucleotides, namely the initial A and final U of the sequence S. They form a trivial new balanced set $\mathcal{B}''$ and so must pair with one another. Remove the tags and boxes but retain the colouring.

Iteration 3:

Box and colour the nucleotides of the new balanced set $\mathcal{B}''$.

$$\boxed{A}\ G\ A\ G\ A\ A\ C\ C\ A\ A\ G\ U\ U\ G\ A\ A\ C\ C\ A\ A\ G\ A\ C$$

$$C\ A\ G\ A\ A\ C\ C\ A\ A\ G\ U\ U\ G\ A\ A\ C\ C\ A\ A\ G\ A\ G\ \boxed{U}$$

At this point, only the nucleotides at height 4 are uncoloured, which is the stopping criterion. Since all coloured nucleotides must pair with one another, and all nucleotides at height 4 (the uncoloured nucleotides) are As, the nucleotides at height 4 must remain unpaired in M.

We now use this fact to determine M completely. Before doing so, we observe several properties that allow us to simplify (and later generalize) the argument presented so far.

Consider the balanced sets $\mathcal{B}$, $\mathcal{B}'$, $\mathcal{B}''$ in iterations 1 to 3. Observe that the nucleotides in S occurring at a given height in τ are either all in, or all not in, the current balanced set. Furthermore the tags (with respect to this balanced set) of all nucleotides at a given height share the same parity, and the parity switches between successive tagged heights. As a result, the labelled tree τ for each iteration 1 to 3 can be condensed as follows:

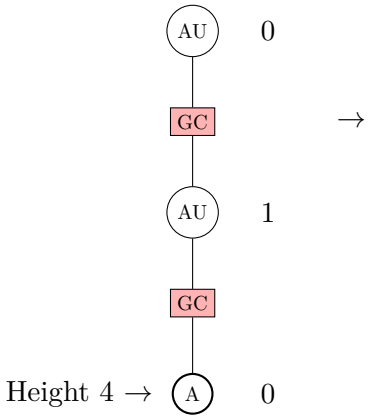


Figure 3.2: Iteration 1

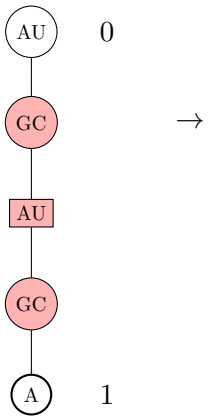


Figure 3.3: Iteration 2

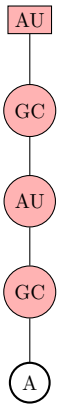


Figure 3.4: Iteration 3

In the condensed tree, all vertices at a given height in the original tree τ are represented by a single vertex labelled with either AU or GC according to the nucleotide types appearing at that height in τ. We now outline how the argument simplifies with the introduction of the condensed tree.

Figure 3.2:

The Gs and Cs in the sequence S form a balanced set $\mathcal{B}$ of nucleotides and so must all pair with one another (to achieve the maximum number of arcs); box and colour the heights at which they occur. To the right of the condensed tree, record at all uncoloured heights

the parity of the running difference with respect to $\mathcal{B}$. Nucleotides with odd parity form a new balanced set $\mathcal{B}'$, and so must all pair with one another.

Figure 3.3:

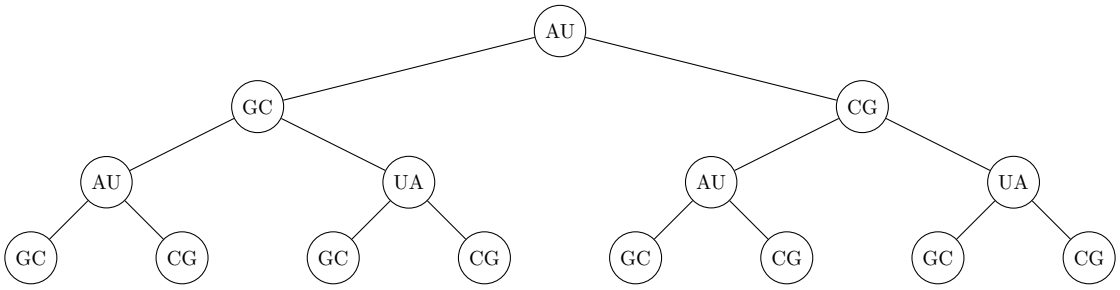
Box and colour the heights at which the nucleotides of the new balanced set $\mathcal{B}'$ occur. To the right of the condensed tree, record at all uncoloured heights the parity of the running difference with respect to $\mathcal{B}'$. Nucleotides whose running difference has even parity form a new balanced set $\mathcal{B}''$, and so must all pair with one another.

Figure 3.4:

Box and colour the height at which the nucleotides of the new balanced set $\mathcal{B}''$ occur. Only nucleotides at height 4 are uncoloured, so stop.

This completes our description of the condensed tree. We now show that M must be the arc set corresponding to T.

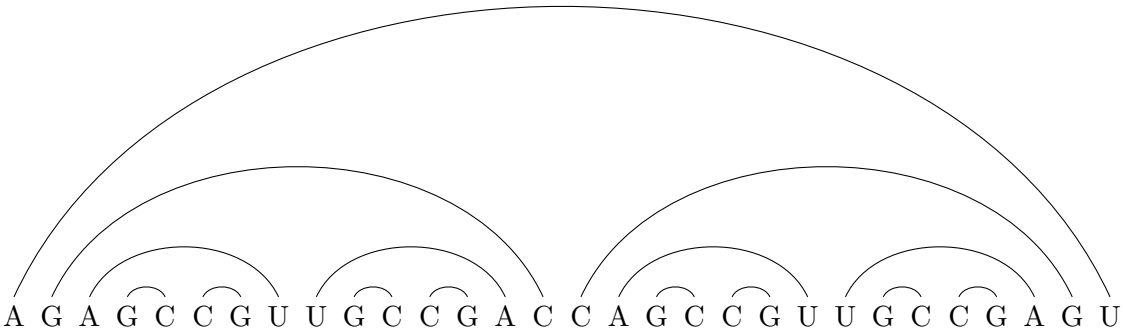
Since all nucleotides in S occurring at height 4 in τ remain unpaired in M, we may remove them from τ to obtain the labelled height 3 saturated full binary tree $\hat{\tau}$, namely:



The corresponding nucleotide sequence $\hat{S}$ is:

A G A G C C G U U G C C G A C C A G C C G U U G C C G A G U

We claim that all nucleotides in $\hat{S}$ occurring at a given height in $\hat{\tau}$ must pair with one another in M. By then considering arcs joining nucleotides in $\hat{S}$ that occur at height 0 to 3 in $\hat{\tau}$ in that order, and applying the condition of no arc crossings, we obtain the following nested sequence of arcs:



This arc set is exactly the arc set of M (because the nucleotides occurring at height 4 in τ remain unpaired in M). Therefore M corresponds to the tree T, as required.

It remains to prove the claim. To show that all nucleotides in $\hat{S}$ occurring at height j ($0 \leq j \leq 3$) in $\hat{\tau}$ must pair with one another in M, we modify the previous process involving boxes, colouring, and tags so that the last remaining uncoloured tree height is j (rather than 4, as in Figure 3.4). We illustrate this modified process for the case $j = 1$, using a condensed tree representation for $\hat{\tau}$. The argument for other values of j is similar.

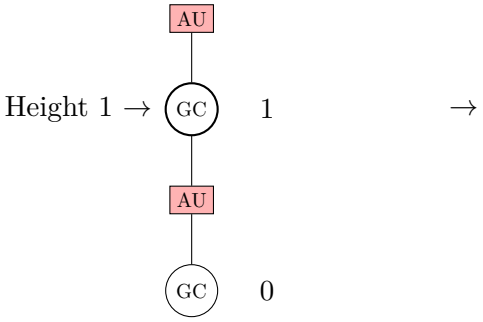


Figure 3.5: Iteration 1

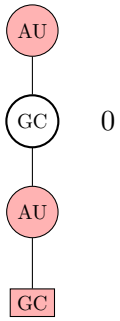


Figure 3.6: Iteration 2

In Figure 3.5 box and colour the even heights (having opposite parity to $j = 1$). The nucleotides occurring at these heights belong to a balanced set $\mathcal{B}$, and so they must all pair with one another. To the right of the condensed tree, record at all uncoloured heights the parity of the running difference with respect to $\mathcal{B}$. Nucleotides whose running difference has even parity (opposite to that of the nucleotides at height $j = 1$) form a new balanced set $\mathcal{B}'$.

In Figure 3.6 box and colour the height at which the nucleotides of the new balanced set $\mathcal{B}'$ occur. All nucleotides at coloured heights must pair with one another. The only height remaining uncoloured is $j = 1$, all of whose nucleotides can therefore pair only with one another. The existence of $\hat{\tau}$ (in which every nucleotide is paired) then shows that the nucleotides at height $j = 1$ must all pair with one another. This establishes the claim.

We shall prove Theorem 6 in Chapter 4 by generalizing the above example.

3.2 Illustration for a larger tree

The example in Section 3.1 uses a labelled P-unsaturated full binary tree τ of height 4. To further illustrate the proof given in Chapter 4, consider the same example but with tree height 11. Figure 3.7 contains the sequence of condensed trees used to show that the nucleotides occurring at height 11 in τ remain unpaired in a maximum-size arc set.

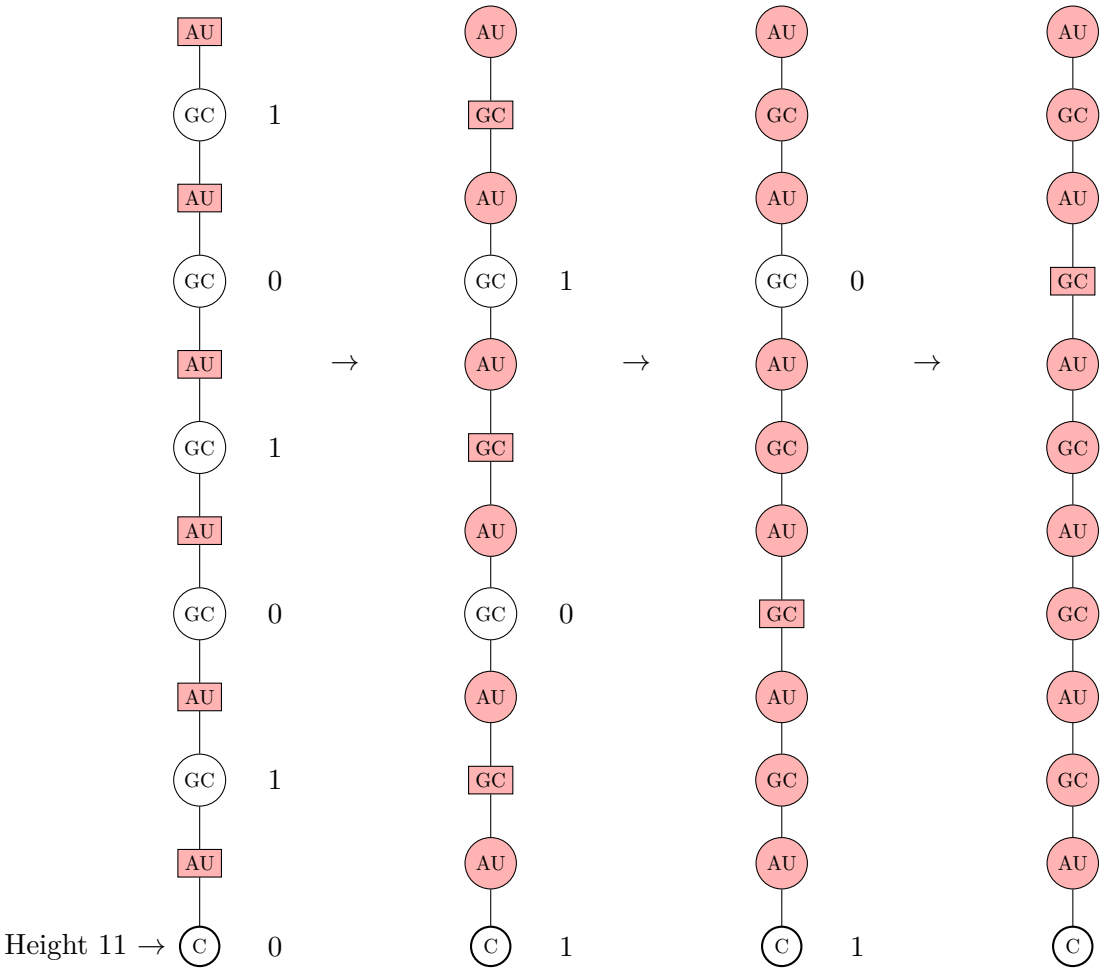


Figure 3.7: Condensed tree process for a labelled P-unsaturated full binary tree τ of height 11

Now remove the nucleotides at height 11 from τ to obtain a labelled saturated full binary tree $\hat{\tau}$ of height 10. A sequence of condensed trees can now be used to show that the nucleotides at height j $(0 \leq j \leq 10)$ in $\hat{\tau}$ must pair with one another in a maximum-size arc set. Figure 3.8 shows this sequence for $j = 3$, and Figure 3.9 shows this sequence for $j = 6$.

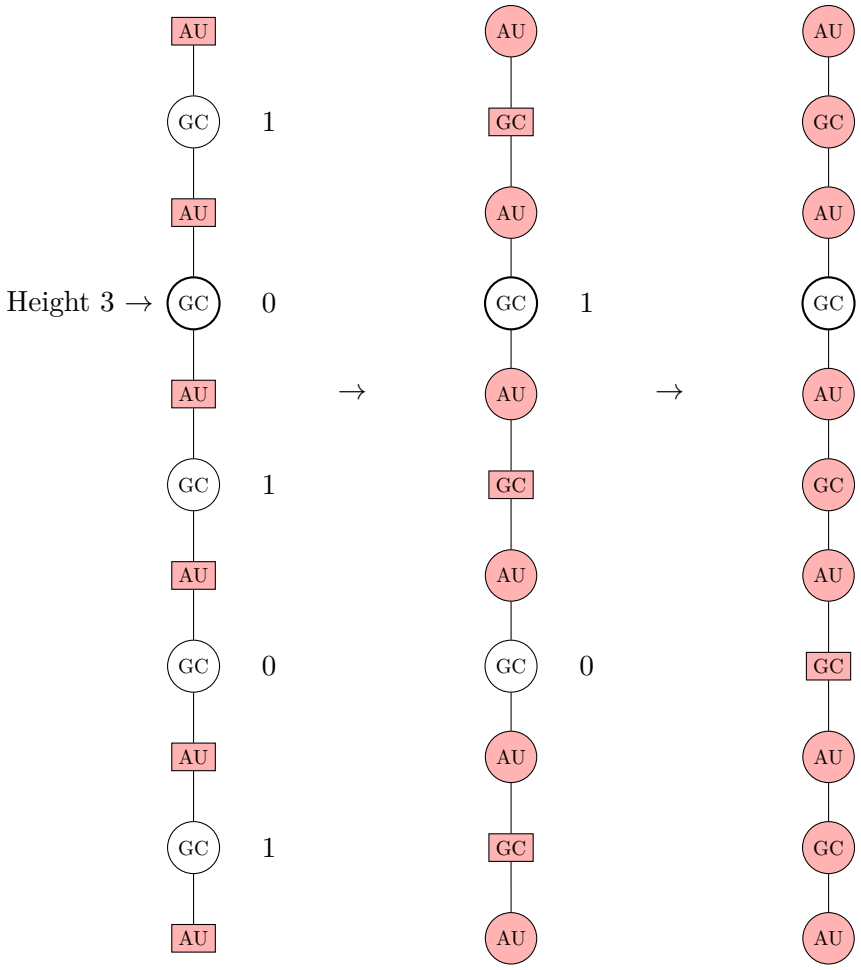


Figure 3.8: Condensed tree process for $\hat{\tau}$ at height 3

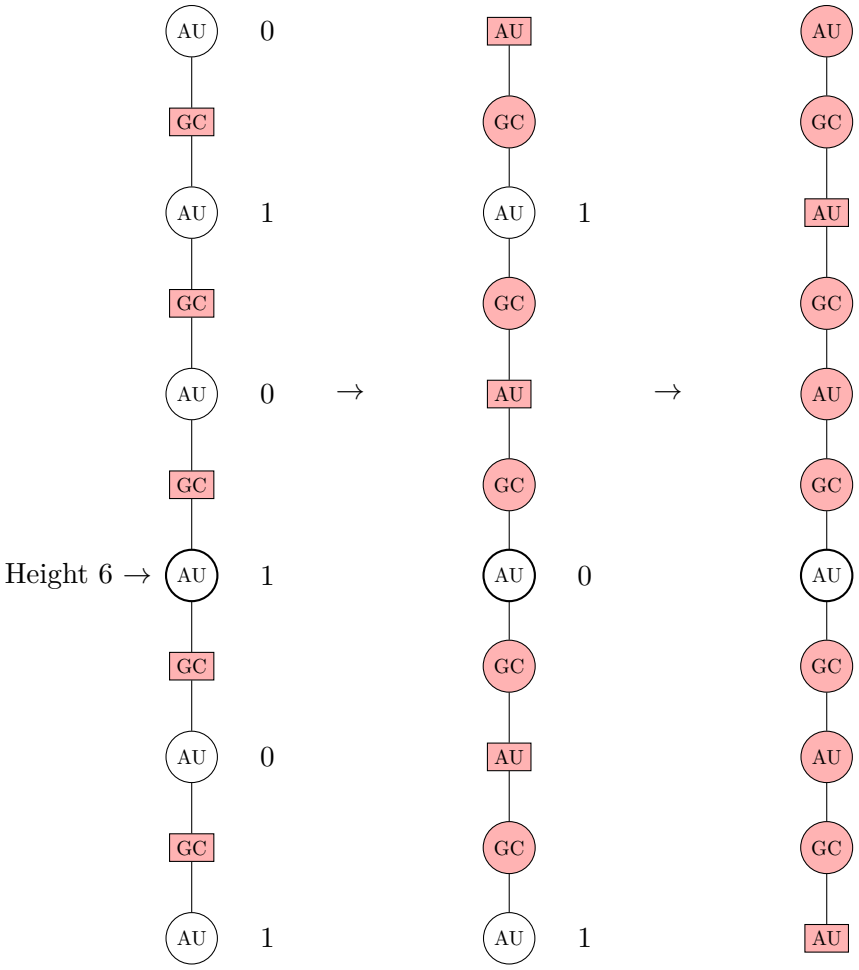


Figure 3.9: Condensed tree process for $\hat{\tau}$ at height 6

Chapter 4

Proof of Main Result

In this chapter, we prove Theorem 6 by following the model exemplified in Chapter 3. For convenience, we restate the theorem.

Theorem 6. *Given a P-unsaturated full floral tree T, there is a design whose (unique) maximum-size arc set M corresponds to T. Such a design is obtained from the natural labelling of T.*

4.1 Two preliminary results

Throughout the proof we require the following two results. The first states that we cannot "jump over" a height of a labelled tree as it is traversed to produce its corresponding nucleotide sequence. The second counts the number of nucleotides at a fixed height that occur in a subsequence of the nucleotide sequence corresponding to the natural labelling of a P-unsaturated full floral tree.

Observation 11. *Let S be the nucleotide sequence corresponding to a labelled tree. Then adjacent nucleotides in S occur either at the same height or at heights that differ by one.*

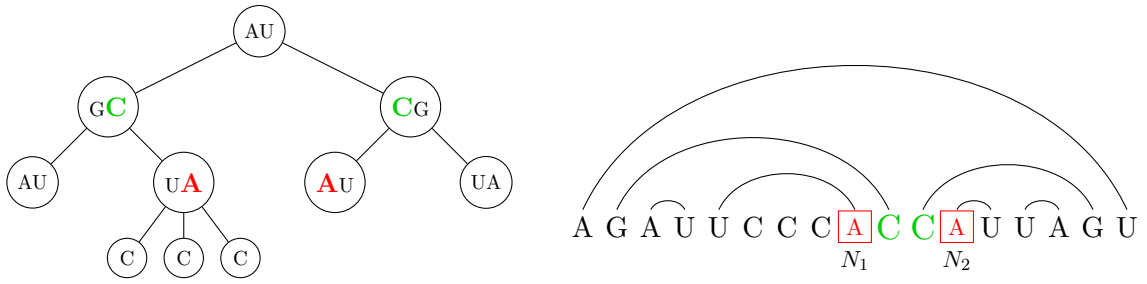
Lemma 12. *Let S be the nucleotide sequence corresponding to a P-unsaturated full floral tree of height $n+1$ labelled according to the natural labelling. Let N_1 be a nucleotide at height k. Suppose at least one nucleotide following N_1 in S is at height ℓ, and let N_2 be the first such nucleotide.*

Further suppose the subsequence of S starting at N_1 and ending at N_2 passes through height $h \notin \{k, \ell\}$ at least once. Then the total number of nucleotides at height h occurring

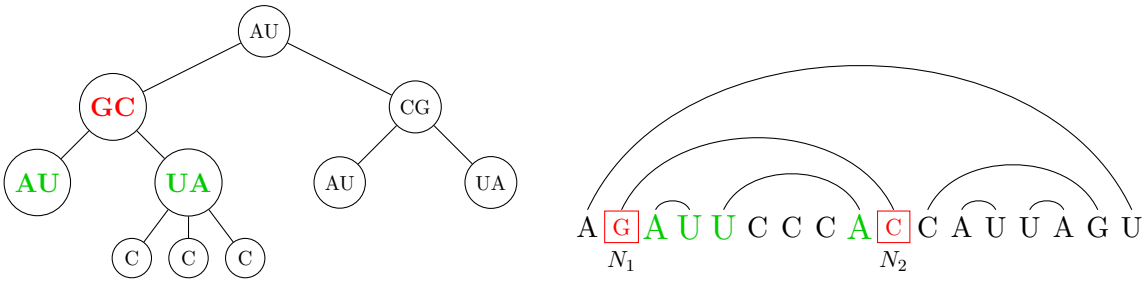
between N_1 *and* N_2 *in* S *is*

$$\begin{cases} 2 & \text{if } \ell = k \text{ and } h < k \\ 2^{h-k+1} & \text{if } \ell = k \text{ and } k < h < n+1 \\ 1 & \text{if } k < h < \ell \\ \text{odd} & \text{if } \ell < h < k. \end{cases}$$

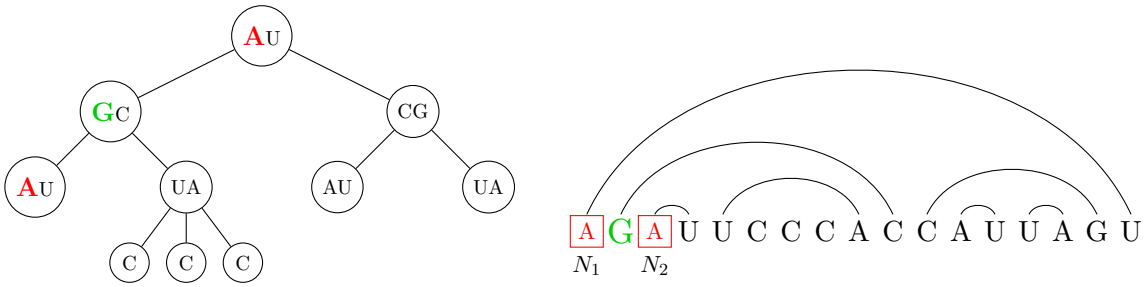
We give the proof of Lemma 12 below. We first demonstrate examples of the four cases of the lemma. Each example uses the same tree (of height $n+1 = 3$) and corresponding arc set. For each case, N_1 and N_2 are enlarged and indicated in red in both the tree representation (left) and arc representation (right). They are also boxed in the arc representation. The nucleotides at a specified height h occurring between them are enlarged and marked in green.



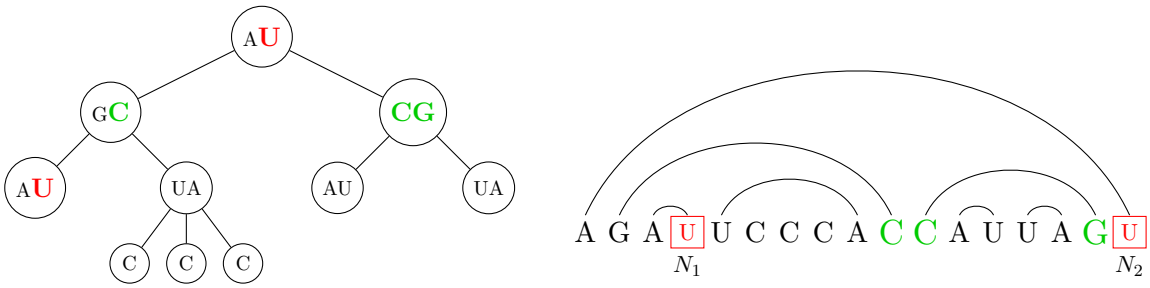
Case 1a: $\ell = k$ and $h < k$. For this case, $\ell = k = 2$ and $h = 1$.



Case 1b: $\ell = k$ and $k < h < n + 1$. For this case, $\ell = k = 1$ and $h = 2$.



Case 2a: $k < h < \ell$. For this case, $\ell = 2$, $h = 1$, and $k = 0$.



Case 2b: $\ell < h < k$. For this case, $\ell = 0$, $h = 1$, and $k = 2$.

Figure 4.1: Illustration of Lemma 12

Proof of Lemma 12. Consider the path through the tree corresponding to the subsequence of S that starts at N_1 and ends at N_2.

Case 1: $\ell = k$. If $h < k$, then the path passes through height h exactly once in the upward direction and exactly once in the downward direction. If $k < h < n+1$, then N_1 and N_2 form a base pair (by definition of N_2), and the path traverses the entire subtree rooted at the vertex labelled by the base pair N_1N_2.

Case 2: $\ell \neq k$. If $k < h < \ell$, then by definition of N_2 the path passes through height h exactly once. If $\ell < h < k$, then consider the subtrees rooted at the vertices at height h. Nucleotide N_1 occurs in one of these subtrees, say σ, and in S it lies between the two nucleotides labelling the root of σ. To reach N_2 from N_1, the path passes through the second of these two nucleotides and then through both the nucleotides labelling the root of all the subtrees (if any) lying to the right of σ in the tree before reaching height ℓ. □

The crucial property which we shall use from Lemma 12 is that the nucleotide count is even if $\ell = k$ and is odd if $\ell \neq k$. Note that we have excluded from Lemma 12 the case where $\ell = k$ and $h = n+1$ both hold because the count of the nucleotides at height $n+1$ of the tree is then not determined by the conditions of the lemma.

4.2 Proof that P-unsaturated full floral trees are designable

We now prove Theorem 6. Given a P-unsaturated full floral tree T of height $n+1 \geq 1$, let S be the sequence corresponding to the natural labelling τ of T. Suppose a maximum-size arc set M is applied to the sequence S.

Claim 1: All nucleotides in S occurring at height $n+1$ in τ remain unpaired in M.

By Claim 1, we may remove the nucleotides at height $n+1$ in τ to obtain a labelled saturated full binary tree $\hat{\tau}$ of height $n \geq 0$. Let $\hat{S}$ be the sequence corresponding to $\hat{\tau}$.

Claim 2: For each j satisfying $0 \leq j \leq n$, all nucleotides in $\hat{S}$ occurring at height j in $\hat{\tau}$ must pair with one another in M.

Apply Claim 2 to successive values of j starting from $j = 0$. The case $j = 0$ forces the leftmost and rightmost nucleotide of $\hat{S}$ to be paired, giving the outermost arc shown below. Each successive case $j \geq 1$, together with the condition that there are no arc crossings, forces a further 2^j arcs of M to take the nested form shown below.

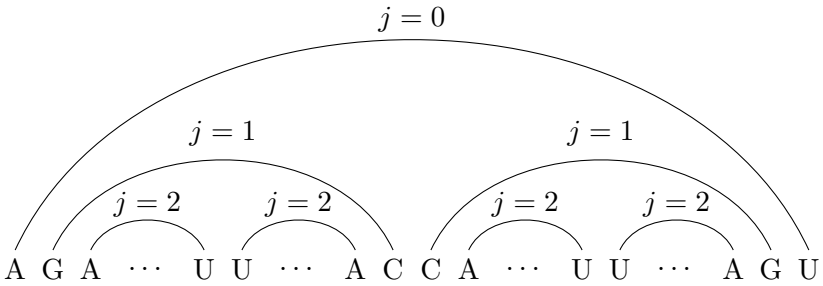


Figure 4.2: Forced secondary structure

Note that this repeated application of Claim 2 to $\hat{S}$ establishes that $\hat{S}$ admits the unique maximum-size arc set shown above, corresponding to a saturated full binary tree, and so $\hat{S}$ is a design. Furthermore, by Claim 1 this arc set is exactly the arc set M. Therefore S is a design whose unique maximum-size arc set M corresponds to T. It remains to prove Claims 1 and 2.

Proof of Claim 1. We use the following algorithm to iteratively tag the nucleotides of the labelled tree τ. This algorithm is modelled on the examples of Chapter 3 involving boxes, colouring, and tags. However, instead of indicating successive balanced sets by $\mathcal{B}, \mathcal{B}', \mathcal{B}'', \ldots$ and using boxes to indicate the current balanced set, we update the current balanced set $\mathcal{B}$ by replacing it with a set $\mathcal{B}'$ that will be shown to be balanced.

Algorithm 1.

Input: Labelled tree τ and corresponding sequence S.

Step 1. Initialize the set $\mathcal{B}$ to be all nucleotides in S occurring in τ at heights whose parity is opposite to that of $n+1$.

Step 2. Colour all nucleotides in $\mathcal{B}$.

Step 3. If only the nucleotides at height $n+1$ are uncoloured, stop.

Step 4. Tag all uncoloured nucleotides by the running difference with respect to $\mathcal{B}$.

Step 5. Let $\mathcal{B}'$ be the set of all tagged nucleotides whose running difference has opposite parity to that of the leftmost nucleotide at height $n+1$. Replace $\mathcal{B}$ by $\mathcal{B}'$.

Step 6. Remove tags (but not the colouring) and go to Step 2.

Output: Sequence S with some nucleotides coloured.

We shall establish that the following six properties hold for Algorithm 1.

Property 1. *For each set $\mathcal{B}$, nucleotides at a given height are either all in $\mathcal{B}$, or all not in $\mathcal{B}$.*

Property 2. *For each set $\mathcal{B}$, the nucleotides at height $n+1$ are not coloured at Step 2 (and so, if Step 4 is reached, these nucleotides receive a tag and therefore Step 5 is well-defined).*

Property 3. *For each set $\mathcal{B}$, the tags (with respect to $\mathcal{B}$) of all nucleotides at a given height share the same parity.*

Property 4. *Parity alternates between successive tagged heights.*

Property 5. *Each set $\mathcal{B}$ is balanced.*

Property 6. *Elements of each balanced set $\mathcal{B}$ must all pair with one another in M.*

Assume Properties 1 to 6 hold, and consider how Algorithm 1 terminates at Step 3. Step 2 colours all nucleotides in set $\mathcal{B}$, which by Property 1 comprises all nucleotides at one or more heights. Therefore each application of Step 2 strictly decreases the number of uncoloured heights. This number is bounded below by 1 because, by Property 2, the nucleotides at height $n+1$ are never coloured. Therefore eventually this number is reduced to 1 at Step 2, at which point only the nucleotides at height $n+1$ are uncoloured and Algorithm 1 immediately terminates at Step 3. At this point, all nucleotides at other heights have been coloured (at some iteration of Step 2), and so belong to some balanced set by Property 5; by Property 6 these nucleotides must all pair with one another in M. The nucleotides at height $n+1$ all have the same type (A or C) and so cannot pair with one another; therefore they remain unpaired in M, establishing Claim 1.

To complete the proof of Claim 1, we must prove Properties 1 to 6.

We first show that Properties 1 to 4 hold for the initial set $\mathcal{B}$ (defined in Step 1).

Property 1: The initial set $\mathcal{B}$ is all nucleotides at tree heights whose parity is opposite to that of $n+1$.

Property 2: All nucleotides at height $n+1$ lie outside $\mathcal{B}$ and so are uncoloured.

Properties 3 and 4: From Step 4, the tagged nucleotides are exactly the uncoloured nucleotides and so occur at heights whose parity is the same as that of $n+1$. Let N_1 be an uncoloured nucleotide, and suppose another uncoloured nucleotide follows N_1 in S. Let N_2 be the first such uncoloured nucleotide. We may assume that at least one nucleotide from $\mathcal{B}$ occurs between N_1 and N_2 in S (otherwise N_1 and N_2 receive identical tags with respect to $\mathcal{B}$).

Let N_1 occur at height k and N_2 at height ℓ. By Property 1 for $\mathcal{B}$, all nucleotides at height ℓ remain uncoloured at Step 2, and so N_2 is the first nucleotide at height ℓ following N_1 in S. Observation 11 shows that $\ell \in \{k-2, k, k+2\}$, and all nucleotides in $\mathcal{B}$ occurring in the subsequence of S starting at N_1 and ending at N_2 occur at a single height $h \in \{k-1, k+1\}$.

The possibilities for (ℓ, h) are $(k, k-1), (k, k+1), (k+2, k+1)$, and $(k-2, k-1)$. By definition of $\mathcal{B}$ we have $h \neq n+1$ (which in fact implies that the case $(\ell, h) = (k, k+1)$ cannot occur). Then by Lemma 12, the number of nucleotides in $\mathcal{B}$ occurring in the subsequence of S starting at N_1 and ending at N_2 is even if $\ell = k$ and is odd if $\ell \neq k$. Therefore the tags with respect to $\mathcal{B}$ of all uncoloured nucleotides at a given height share the same parity, and the parity alternates between successive tagged heights.

We now assume Properties 1 to 4 hold for the current set $\mathcal{B}$, and show that they hold for the set $\mathcal{B}'$ defined in Step 5.

Property 1: By Properties 2 and 3 for $\mathcal{B}$, all nucleotides at height $n+1$ are tagged with respect to $\mathcal{B}$ and have the same parity. So $\mathcal{B}'$ comprises all tagged nucleotides whose running difference with respect to $\mathcal{B}$ has opposite parity to that of all the nucleotides at height $n+1$. Then by Property 3 for $\mathcal{B}$, we see that Property 1 holds for $\mathcal{B}'$.

Property 2: nucleotides at height $n+1$ do not belong to $\mathcal{B}'$ (from Step 5) and so are not coloured (in Step 2).

Properties 3 and 4: We represent the application of Steps 2 and 4 to $\mathcal{B}'$ by the following sequence of "condensed trees".

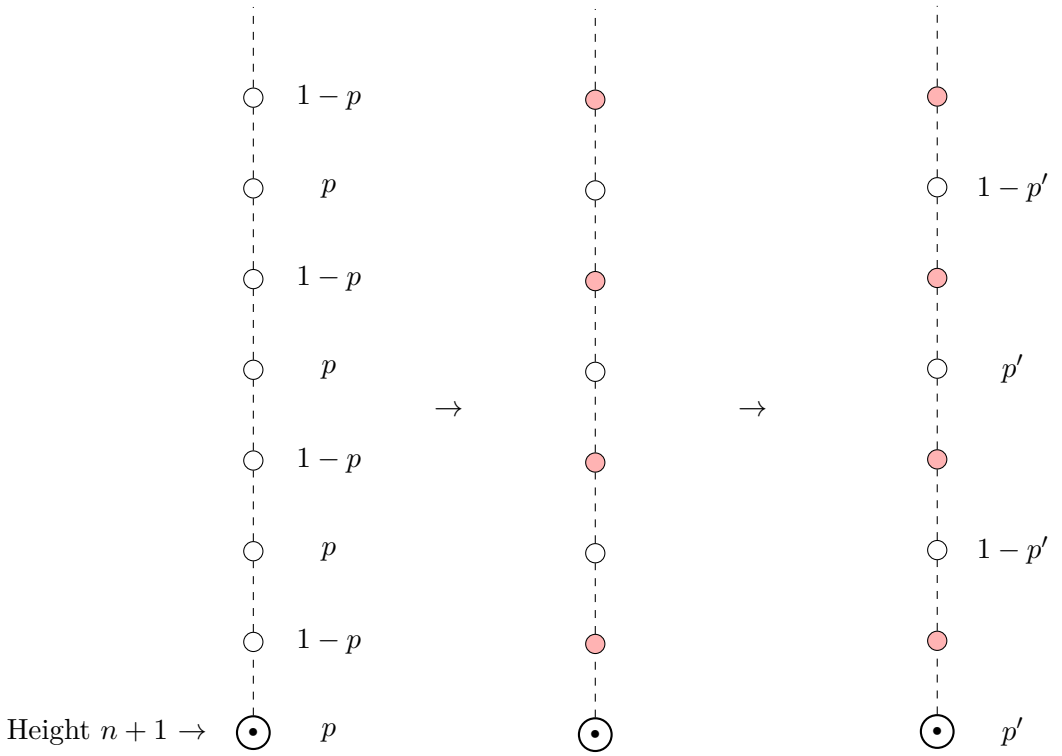


Figure 4.3 Figure 4.4 Figure 4.5

Figure 4.3 represents the tree prior to applying Step 5 to produce $\mathcal{B}'$; only the heights containing nucleotides that are tagged with respect to $\mathcal{B}$ (and which are necessarily un-

coloured) are shown. (Previously coloured nucleotides may occur at other heights not shown.) By Property 1 for $\mathcal{B}$, all nucleotides at a given such height may be represented by a single vertex. By Property 3 for $\mathcal{B}$, each such vertex may be assigned a single parity (p or $1-p$), representing the parity of the tag shared by all nucleotides at that height. By Properties 2 and 4 for $\mathcal{B}$, the nucleotides at height $n+1$ appear as a tagged vertex in the condensed tree, say with parity p, and the parity of vertices at successive tagged heights alternates. The set $\mathcal{B}'$ therefore comprises all nucleotides at heights tagged with parity $1-p$. These heights are coloured in Figure 4.4 to indicate that the nucleotides at these heights are coloured when Step 2 is applied to $\mathcal{B}'$.

Figure 4.5 shows the parity of the tags assigned to nucleotides when Step 4 is applied to $\mathcal{B}'$, as we now describe; this establishes Properties 3 and 4 for $\mathcal{B}'$. The uncoloured nucleotides are tagged in Step 4 by the running difference with respect to $\mathcal{B}'$. Let N_1 be an uncoloured nucleotide, and suppose another uncoloured nucleotide follows N_1 in S. Let N_2 be the first such uncoloured nucleotide. We may assume that at least one nucleotide from $\mathcal{B}'$ occurs between N_1 and N_2 in S (otherwise N_1 and N_2 receive identical tags with respect to $\mathcal{B}'$).

Let N_1 occur at height k and N_2 at height ℓ. By Property 1 for $\mathcal{B}'$, all nucleotides at height ℓ remain uncoloured at Step 2, and so N_2 is the first nucleotide at height ℓ following N_1 in S. Observation 11 shows that k and ℓ are either the same height or are successive uncoloured heights. In both cases, all nucleotides in $\mathcal{B}'$ occurring in the subsequence of S starting at N_1 and ending at N_2 occur at a single height h (see Figure 4.4).

In order to apply Lemma 12, we verify the appropriate conditions on k, h, and ℓ. By Property 1 for $\mathcal{B}'$, all nucleotides at height h are in $\mathcal{B}'$ (coloured at Step 2 for $\mathcal{B}'$) and all nucleotides at heights k and ℓ are not in $\mathcal{B}'$ (uncoloured). Therefore $h \notin \{k, \ell\}$, and by Property 2 for $\mathcal{B}'$ we have $h \neq n+1$. By Observation 11, if $k \neq \ell$ then either $k < h < \ell$ or $\ell < h < k$.

Then by Lemma 12, the number of nucleotides in $\mathcal{B}'$ occurring in the subsequence of S starting at N_1 and ending at N_2 is even if $\ell = k$ and is odd if $\ell \neq k$. Therefore the tags with respect to $\mathcal{B}'$ of all uncoloured nucleotides at a given height share the same parity, and the parity alternates between successive tagged heights.

This completes the proof that Properties 1 to 4 hold for Algorithm 1.

We next show that Properties 5 and 6 hold for the initial set $\mathcal{B}$ (defined in Step 1).

Property 5: The initial set $\mathcal{B}$ comprises all G and C nucleotides if $n+1$ is even, or all A and U nucleotides if $n+1$ is odd, and so is balanced (by definition of the natural labelling).

Property 6: The labelled tree τ from which the sequence S is derived shows that S admits an arc set in which only nucleotides of a single type (A if $n+1$ is even, or C if $n+1$ is odd) are unpaired; therefore in the maximum-size arc set M all nucleotides from

the initial set $\mathcal{B}$ must pair with one another.

We now assume Properties 5 and 6 hold for the current set $\mathcal{B}$, and show that they hold for $\mathcal{B}'$.

Property 5: The initial application of Step 2 colours either all G and C nucleotides, or all A and U nucleotides. Since colouring is never removed, the set $\mathcal{B}'$ comprises only (a subset of the) A and U nucleotides, or G and C nucleotides, respectively. Furthermore, the set $\mathcal{B}'$ comprises all nucleotides occurring at some subset of heights not including height $n+1$ (see Figure 4.4). Therefore, from the definition of the natural labelling, the nucleotides at this subset of heights form a new balanced set.

Property 6: The elements of the balanced set $\mathcal{B}$ must all pair with one another, by Property 6. So each arc joining two nucleotides not in $\mathcal{B}$ must enclose an equal number of the two nucleotide types in $\mathcal{B}$ (otherwise the arc will induce a crossing). Therefore nucleotides tagged in Step 4 with the same running difference with respect to $\mathcal{B}$ can pair only with one another. In particular, nucleotides tagged in Step 4 whose running differences with respect to $\mathcal{B}$ have the same parity can pair only with one another. It follows that the elements of the set $\mathcal{B}'$ defined in Step 5 (whose tags with respect to $\mathcal{B}$ all share the same parity) can pair only with one another.

Now the labelled tree τ from which the sequence S is derived shows that S admits an arc set in which only nucleotides of a single type (A or C) are unpaired; therefore in the maximum-size arc set M all nucleotides of the other three types must be paired. By Property 5 we know that $\mathcal{B}'$ forms a balanced set, and we have shown that its elements can pair only with one another. Therefore the elements of $\mathcal{B}'$ must all pair with one another.

This completes the proof that Properties 5 and 6 hold for Algorithm 1. □

Proof of Claim 2. Fix j satisfying $0 \leq j \leq n$. We shall apply Algorithm 1 to $\hat{\tau}$ with $n+1$ replaced throughout by j. We shall show that Properties 1 to 6 hold for this modified algorithm, again with $n+1$ replaced by j. By the same argument as previously, these properties imply that all nucleotides at heights other than j must pair with one another in M. The nucleotides at height j can therefore pair only with one another in M, and the existence of $\hat{\tau}$ (in which every nucleotide is paired) then shows that the nucleotides at height j must all pair with one another in M, proving Claim 2.

In order to establish that Properties 1 to 6 hold for the modified algorithm, we highlight only the places in which the argument differs from that given previously.

Changes throughout:

Replace $n+1$ by j, τ by $\hat{\tau}$, and S by $\hat{S}$.

Changes to the proof that Properties 3 and 4 hold for $\mathcal{B}'$:

Replace Figures 4.3 to 4.5 by Figures 4.6 to 4.8, respectively.

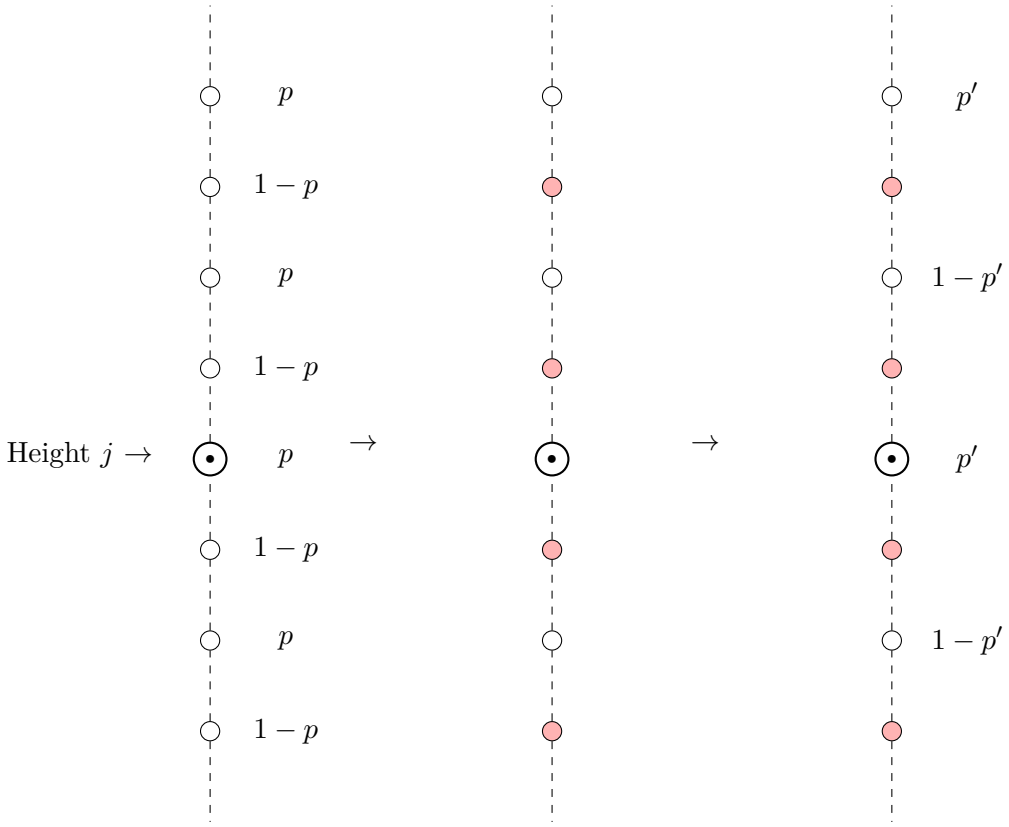


Figure 4.6 Figure 4.7 Figure 4.8

Changes to the proof that Property 6 holds for the initial set $\mathcal{B}$:

The labelled tree $\hat{\tau}$ from which the sequence $\hat{S}$ is derived shows that $\hat{S}$ admits an arc set in which all nucleotides are paired; therefore in the maximum-size arc set M all nucleotides from the initial set $\mathcal{B}$ (comprising all G and C nucleotides or all A and U nucleotides) must pair with one another.

Changes to the proof that Property 6 holds for $\mathcal{B}'$:

As previously, the elements of $\mathcal{B}'$ can pair only with one another. The labelled tree $\hat{\tau}$ from which the sequence $\hat{S}$ is derived shows that $\hat{S}$ admits an arc set in which all nucleotides are paired; therefore in the maximum-size arc set M all nucleotides must be paired. Therefore the elements of $\mathcal{B}'$ must all pair with one another.

□

Claims 1 and 2 have now been established and so the proof of Theorem 6 is complete.

Chapter 5

Future Directions

Here we discuss some questions that arise from this thesis.

Q1. Given a P-unsaturated full floral tree T, how many sequences satisfy the conditions of Theorem 6? Can the proof of Theorem 6 be adapted to count (or give a lower bound on) the number of designs which have a (unique) maximum-size arc set corresponding to T?

It would be easy to produce a number of sequences that satisfy Theorem 6 via a simple mapping of the four nucleotides. However, the proof of Theorem 6 relies on the existence of at least one initial balanced set with respect to which we calculate running differences. It is not obvious how to adapt the method of our proof to count designs which do not contain some balanced set.

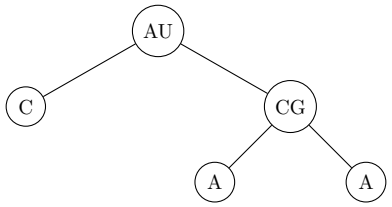
Q2. In the natural labelling of a P-unsaturated full floral tree, only one type of nucleotide (A or C) can remain unpaired. Are there designs, for this or for other secondary structures, in which more than one type of nucleotide can remain unpaired? How many such sequences are there?

One can produce many designs that include more than one unpaired nucleotide type by examining small examples of secondary structures. However, this becomes much more difficult for large secondary structures. We do not think our method can be easily extended to include structures which allow more than one unpaired nucleotide type.

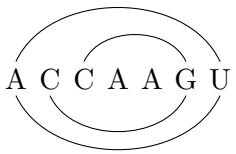
Q3. Theorem 3 states that the secondary structure corresponding to a saturated tree, all of whose vertices have at most three children, is designable. Corollary 7 states that a secondary structure corresponding to a P-unsaturated floral tree is designable. But we know from Proposition 4 that the secondary structure corresponding to an

unsaturated tree is not designable in general. Is it possible to find a design for all unsaturated binary trees?

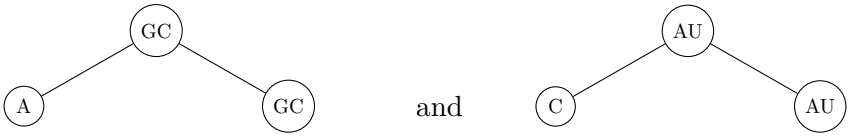
The constraint 'P-unsaturated' in Theorem 6 restricts the positions of unpaired nucleotides to (pendant) vertices at maximum height in the tree, whereas Question 3 allows unpaired nucleotides at all tree heights. In that case, by examining small trees it is easy to show that a simple modification of the natural labelling will not suffice. For example, the following labelled tree contains an unpaired nucleotide at height 1 (and its labelling is a simple modification of the natural labelling):



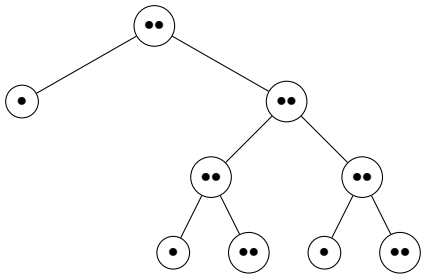
This labelled tree has corresponding sequence and two maximum-size arc sets (shown above and below the sequence):



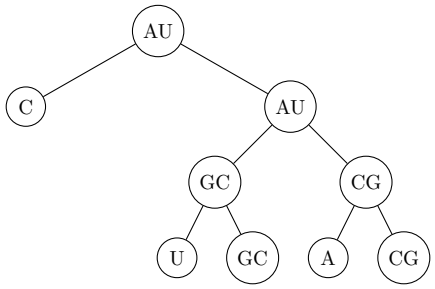
In fact, we found that the sequences corresponding to the labelled trees below were much more effective for constructing designs for subtrees of this form:



So Question 3 is not answered by a simple modification of the labelling used to prove Theorem 6. In fact, there are instances of unsaturated binary trees for which a design exists, but this design must include three types of unpaired nucleotides. For example, the following unsaturated binary tree



is designable with the labelling



and it can be exhaustively shown that no labelling of the tree having fewer than three unpaired nucleotide types will result in a design.

Q4. Proposition 4 shows the existence of a secondary structure that is not designable. Some other classes of structures which are not designable are characterized in [5]. How general can these results be made? Such configurations can be thought of as obstructions for designable secondary structures: if a secondary structure M contains one or more of these configurations, then M will not be designable. Is it possible to find an exhaustive list of obstructions?

Q5. As discussed in Section 1.2, the Watson-Crick energy model assigns to every base pair {G,C} and every base pair {A,U} an energy score of -1, and a score of ∞ to every other base pair [5]. With this model, the goal of the combinatorial RNA design problem is to find a design for a target secondary structure that minimizes the total energy score. This is equivalent to maximizing the number of {G,C} and {A,U} base pairs, and is the model that has been primarily studied in this thesis.

In the more general energy model studied by Reinharz, Ponty, and Waldispühl, each {G,C} base pair is assigned a negative energy score α, and each {A,U} base pair is assigned a negative energy score β [15]. The combinatorial RNA design problem then becomes: given a target secondary structure M, find a design that minimizes the total energy score and whose (unique) maximum-size arc set corresponds to M. An algorithm for solving this more general combinatorial RNA design problem is presented in [15]. Can the methods of this thesis be extended to this problem?

The energy model of Question 5 is more relevant in biology than the Watson-Crick energy model: the assignment of distinct weights α and β reflects differing probabilities of base pair types {G,C} and {A,U}. We think it might be relatively straightforward to answer Question 5.

Q6. How does the answer to Question 5 change under the Nussinov-Jacobson model (also discussed in Section 1.2), in which negative weighted energy scores α, β, γ are assigned to the three base pair types {A,U}, {G,C}, {G,U}, respectively (where $\alpha < \gamma$ and $\beta < \gamma$), and an energy score of ∞ is assigned to all other base pair types [5]?

If it is possible to answer Question 5, then it is likely that Question 6 would follow easily using a similar method as used to prove Corollary 8.

Q7. The combinatorial RNA design problem seeks a nucleotide sequence with a unique maximum-size arc set. How would the outcome change if we allowed a nucleotide sequence to have at most two maximum-size arc sets?

Although the structure of RNA determines its biological function, the condition that the maximum-size arc set must be unique might be too strict for synthetic biology. Indeed, if a target secondary structure corresponds to one of many maximum-size arc sets for a sequence, one could deduce the probability of that sequence folding into the desired secondary structure. We do not anticipate an easy extension of the proof of Theorem 6 to answer Question 7, since the majority of that proof focuses on uniqueness.

Bibliography

[1] R. Aguirre-Hernández, H.H. Hoos, and A. Condon. Computational RNA secondary structure design: empirical complexity and improved methods. *BMC Bioinformatics*, 8(1):34, 2007.

[2] A. Avihoo, A. Churkin, and D. Barash. RNAexinv: An extended inverse RNA folding from shape and physical attributes to sequences. *BMC Bioinformatics*, 12(1):319, 2011.

[3] A. Busch and R. Backofen. Info-RNA–a fast approach to inverse RNA folding. *Bioinformatics*, 22(15):1823–1831, 2006.

[4] A. Esmaili-Taheri, M. Ganjtabesh, and M. Mohammad-Noori. Evolutionary solution for the RNA design problem. *Bioinformatics*, 30(9):1250–1258, 2014.

[5] J. Haleš, A. Héliou, J. Maňuch, Y. Ponty, and L. Stacho. Combinatorial RNA Design: Designability and structure-approximating algorithm in Watson-Crick and Nussinov-Jacobson energy models. *Algorithmica*, pages 1–22. DOI:10.1007/s00453–016–0196–x, 2016.

[6] C. Höner zu Siederdissen, S. Hammer, I. Abfalter, I.L. Hofacker, C. Flamm, and P.F. Stadler. Computational design of RNAs with complex energy landscapes. *Biopolymers*, 99(12):1124–1136, 2013.

[7] F.J. Isaacs, D.J. Dwyer, and J.J. Collins. RNA synthetic biology. *Nature Biotechnology*, 24(5):545–554. DOI:10.1038/nbt1208, 2006.

[8] A. Levin, M. Lis, Y. Ponty, C.W. O'Donnell, S. Devadas, B. Berger, and J. Waldispühl. A global sampling approach to designing and reengineering RNA secondary structures. *Nucleic Acids Research*, 40(20):10041–10052, 2012.

[9] R.B. Lyngsø, J.W.J. Anderson, E. Sizikova, Am. Badugu, T. Hyland, and J. Hein. FRNAkenstein: multiple target inverse RNA folding. *BMC Bioinformatics*, 13(1):260, 2012.

[10] D.H. Mathews, J. Sabina, M. Zuker, and D.H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, 288(5):911–940, 1999.

[11] M.E. Nebel, A. Scheid, and F. Weinberg. An RNA secondary structure. `https://openi.nlm.nih.gov/detailedresult.php?img=PMC2735962_bbi-2008-239f3&req=4`, 2008. [Online; accessed April 3, 2017].

[12] R. Nussinov and A.B. Jacobson. Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proceedings of the National Academy of Sciences*, 77(11):6309–6313, 1980.

[13] Y. Ponty. Personal communication, August 2016.

[14] J.B. Reece, L.A. Urry, M.L. Cain, S.A. Wasserman, P.V. Minorsky, and R.B. Jackson. *Campbell Biology*. Pearson Boston, 9th edition, 2011.

[15] V. Reinharz, Y. Ponty, and J. Waldispühl. A weighted sampling algorithm for the design of RNA sequences with targeted secondary structure and nucleotide distribution. *Bioinformatics*, 29(13):i308–i315, 2013.

[16] G. Rodrigo, T.E. Landrain, E. Majer, J. Daròs, and A. Jaramillo. Full design automation of multi-state RNA devices to program gene expression using energy-based optimization. *PLOS Computational Biology*, 9(8):e1003,172, 2013.

[17] M. Schnall-Levin, L. Chindelevitch, and B. Berger. Inverting the Viterbi algorithm: an abstract framework for structure design. In *Proceedings of the 25th International Conference on Machine Learning*, pages 904–911. ACM, 2008.

[18] P. Shareghi, Y. Wang, R. Malmberg, and L. Cai. The general tRNA tertiary structure (and the secondary structure in the box). `https://openi.nlm.nih.gov/detailedresult.php?img=PMC3394421_1471-2164-13-S3-S7-1&query=tertiary+structure&req=4&npos=18`, 2012. [Online; accessed April 15, 2017].

[19] A. Taneda. MODENA: a multi-objective RNA inverse folding. *Advances and Applications in Bioinformatics and Chemistry*, 4:1–12, 2011.

[20] D.H. Turner and D.H. Mathews. NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Research*, 38:D280–D282, 2010.

[21] J.N. Zadeh, B.R. Wolfe, and N.A. Pierce. Nucleic acid sequence design via efficient ensemble defect optimization. *Journal of Computational Chemistry*, 32(3):439–452, 2011.

[22] Y. Zhou, Y. Ponty, S. Vialette, J. Waldispühl, Y. Zhang, and A. Denise. Flexible RNA design under structure and sequence constraints using formal languages. In *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics*, pages 229–238. ACM, 2013.

[23] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1):133–148, 1981.