CALIFORNIA STATE UNIVERSITY, NORTHRIDGE

PROTEIN FOLDING: PLANAR CONFIGURATION SPACES OF DISC ARRANGEMENTS AND HINGED POLYGONS: *PROTEIN FOLDING IN FLATLAND*

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

by

Clinton Bowen

August 2014

The thesis of Clinton Bowen is approved:

Dr. Silvia Fernandez — Date

Dr. John Dye — Date

Dr. Csaba Tóth, Chair — Date

California State University, Northridge

Table of Contents

ABSTRACT

PROTEIN FOLDING: PLANAR CONFIGURATION SPACES OF DISC ARRANGEMENTS AND HINGED POLYGONS: *PROTEIN FOLDING IN FLATLAND*

By

Clinton Bowen

Master of Science in Computer Science

abstract goes here

Chapter 1

Background

In this section, we cover the background subjects needed to formally pose the problem and solutions in this thesis. We start with two types of combinatorial structures, linkages and polygonal linkages. We then discuss the configuration spaces of linkages and polygonal linkages. We then look into an alternate representation of linkages, disk arrangements and state the disk arrangement theorem but do not show its proof. We then look at satisfiability problems, a logic engine that encodes a type of satisfiability problems. Finally, we cover the basic definitions of algorithm complexity for **P** and **NP**.

1.1 Graphs

A *graph* is an ordered pair $G = (V, E)$ comprising of a set V of vertices or together with a set E of edges or lines. For every edge $e \in E$, there is a distinct pair of vertices in V that represents their adjacency, $(u, v) \in E$. The edge length mapping of a graph is $l : E \mapsto \mathbb{R}^+$. A graph is an abstract combinatorial structure until an *embedding* on the graph is posed, $\Pi : V \mapsto \mathbb{R}^2$. Π has the *proper embedding* property, if for every edge $(u, v) \in E$ such that $l((u, v)) = |\Pi(u) - \Pi(v)|$ is true. The *realization* of the graph is said to be the range Π, $\Pi(V)$.

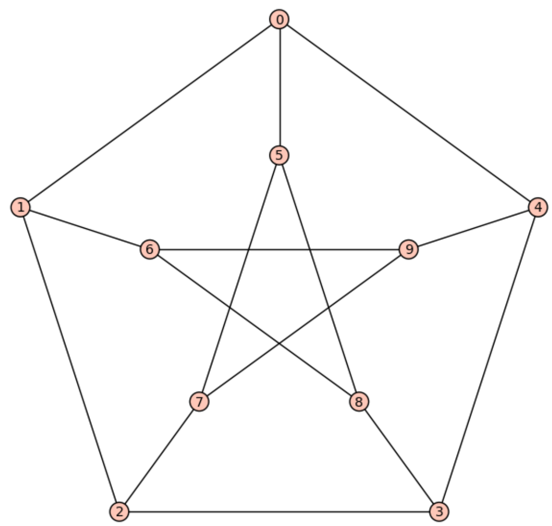


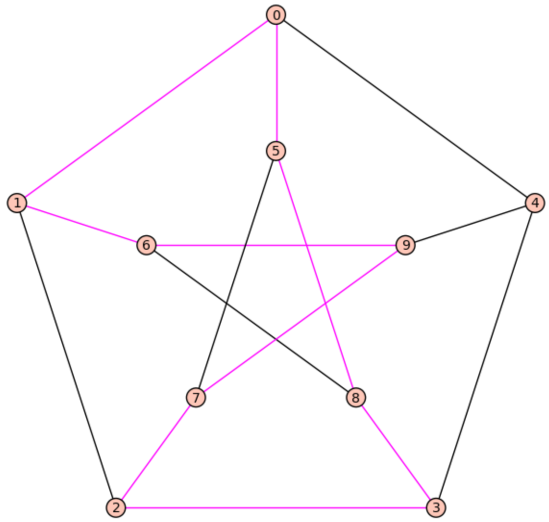
Figure 1.1: A proper embedding of a Peterson graph.

1.1.1 Trees

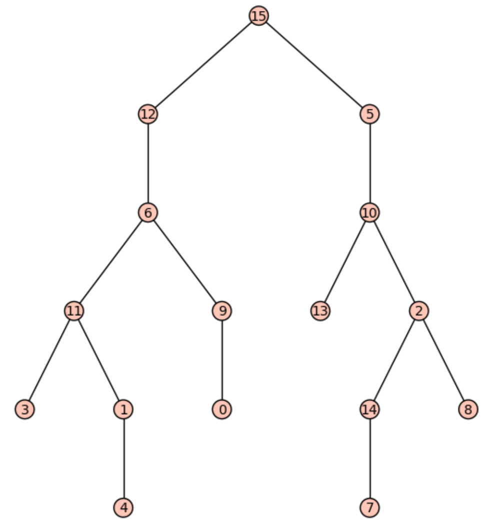
A *simple cycle* is a sequence of adjacent vertices that do not repeat with the exception of the starting (ending) vertex. A *tree* is a graph that has no simple cycles. The vertices of an embedded tree have a nomenclature. One vertex is designated to be the *root* where the edges are oriented away from the root. Adjacent vertices have a parent-child relationship. The *parent* is closer to the root, while the *child* is further way from the root.

1.1.2 Ordered Trees

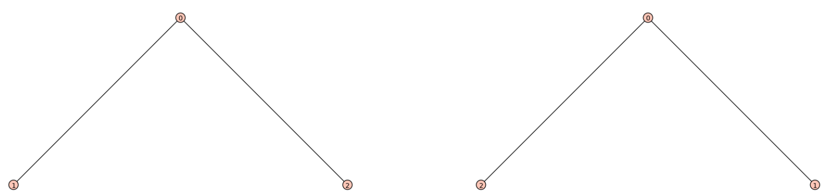
An *ordered tree* is a rooted tree in which the counter-clockwise ordering of children of a common parent matters. The orderedness of a tree is a property of the tree's embedding. A fortiori, embeddings of ordered



(a) A proper embedding of the Peterson graph with a simple cycle of (2,7,9,6,1,0,5,8,3,2).



(b) A tree with 16 vertices



(a) An embedding of a tree.

(b) Another embedding of a tree.

Figure 1.3: A rooted tree with whose children have a different counter-clockwise ordering. Note that these ordered trees are not equivalent.

trees are equivalent if for each parent, the counter-clockwise ordering of children are the same.

1.1.3 Graph Isomorphism

Given two graphs $G = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, a bijection $f : V_1 \mapsto V_2$ such that for any two vertices $u, v \in V_1$ that are adjacent, i.e. $(u, v) \in E_1$, if and only if $(f(u), f(v)) \in E_2$.

Graph	Vertices	Edges
G_1	$\{a,b,c,d,e\}$	$\{(a,b),(b,c),(c,d),(d,e),(e,a)\}$
G_2	$\{1,2,3,4,5\}$	$\{(1,2),(2,3),(3,4),(4,5),(5,1)\}$

Table 1.1: Two graphs that are isomorphic with the alphabetical isomorphism $f(a) = 1$, $f(b) = 2$, $f(c) = 3$, $f(d) = 4$, $f(e) = 5$.

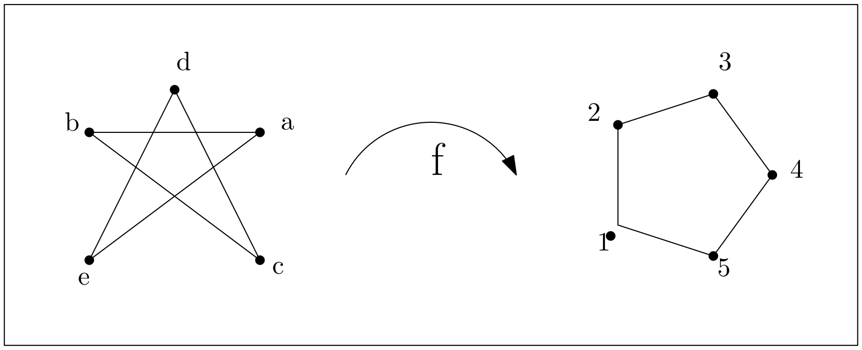


Figure 1.4: This figure depicts the graph isomorphism shown in Table (**??**) between V_1 and V_2 in the plane.

1.1.4 Linkages

There are two parts to a linkage, the graph and the edge length mapping, e.g. (G,l). While graphs can have vertices that may not correspond to any edges, we rule out this possibility for linkages. Like a graph, a linkage is still an abstract combinatorial structure until an *embedding* on the graph of the linkage is posed and the definitions of a proper embedding and realization of a graph are the same for embeddings.

1.1.4.1 Edge Crossings

(a) An edge crossing where two edges overlap.

(b) An edge crossing where two edges intersect at a point.

(c) An edge crossing where two edges intersect at a vertex.

(d) An edge crossing where two edges intersect at two vertices.

Figure 1.5: These figures exhibit the 4 types of edge crossings.

1.1.4.2 Loops

1.1.4.3 Multi-edges

Without loss of generality, for this paper, we focus on linkages that have simple planar graph properties, i.e.:

(i) does not have edges that cross,

(ii) does not have loops, i.e. $(v,v) \in E$, or

(iii) does not have multiple edges between any pair of vertices.

We may visit special cases in which we look at planar graphs that satisfy the last two conditions but not the first, e.g.:

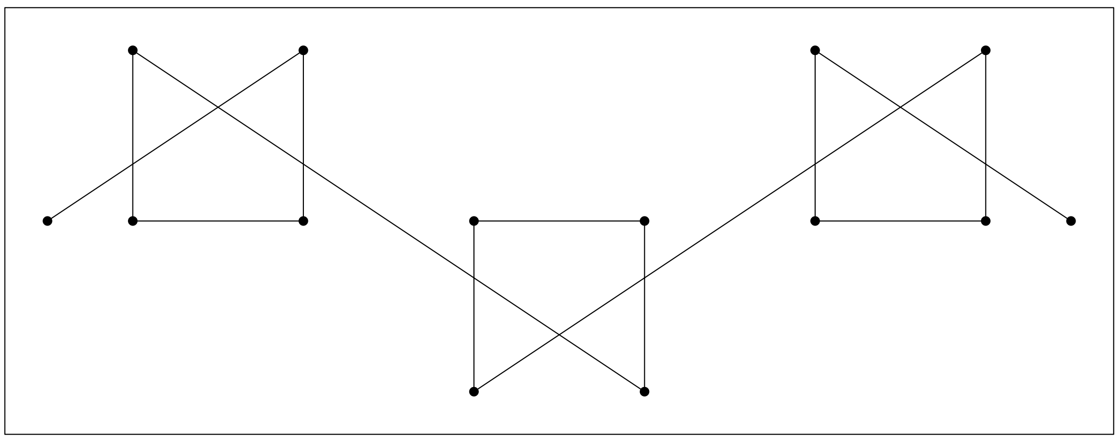


Figure 1.6: A linkage where edges cross however it does not contain loops or multiple edges between vertices.