Analysis of Connected Graphs on 9 Vertices with H8 and A9 Minor

Ryan Watson

May 14, 2010

Advisor: Thomas Mattman

Abstract

Let H8 and A9 denote graphs obtained from K7 and K3311 by a single triangle-Y move. The H8 graph is one of fourteen obtained from K7 in the paper of Kohara and Suzuki [KS] and is the unique one on 8 vertices. Though there are two graphs on 9 vertices that can be obtained from K3311 by a single triangle-Y, A9 and B9, this study will focus on A9.

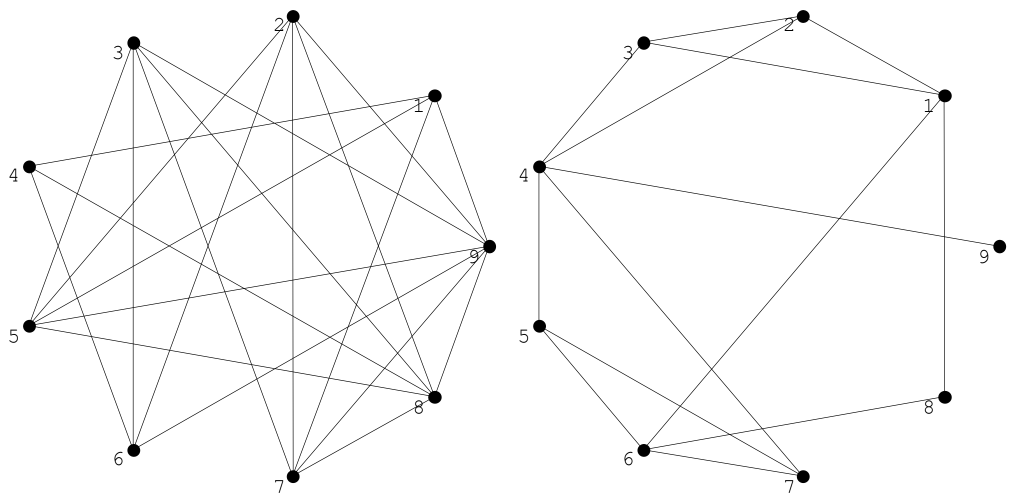
As K7 and K3311 are intrinsically knotted, the same is true of H8 and A9 as well as any graph that contains either of these as a minor. This project is part of an effort to classify connected graphs on 9 vertices. Most contain H8 as a minor, so it's natural to split the problem into those containing H8 and those that do not contain H8. This study pursues a further subdivision of the not H8 class into those that are A9 versus those that are not A9.

Chapter 1

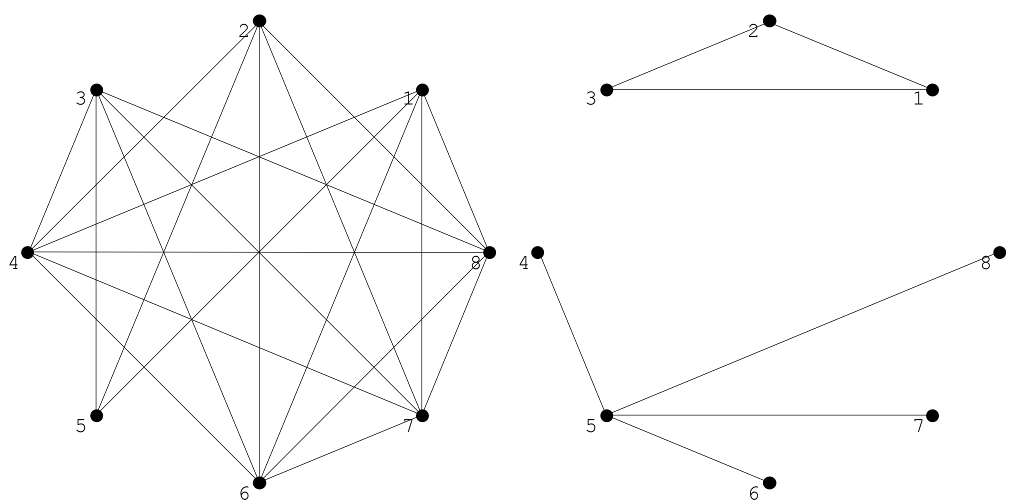
Definition of the graphs A9 and H8:

These graphs are shown below. We'll say a connected graph on nine vertices is A9 if it has the A9 graph as a minor. Similarly, we'll say a graph with H8 as a minor is an H8 graph. If G does not have an A9 minor, we'll say G is not A9, similarly for H8.

Graph of A9 (244448):



Graph of H8:



This thesis will analyze the intrinsically knotted connected graphs on 9 vertices to determine whether they have A9 or H8 minors. This thesis has been broken down into chapters in order to provide a breakdown of topics and make it easier for the reader to follow the steps which were taken to accomplish this task. Chapter one introduces needed definitions. Also included is the background work that was done prior to this project. Chapter two breaks down the code that was written for this project and explains what it does and why it does it. Chapter three consists of the results and proofs of the thesis. All of the propositions and conjectures, as well as selected proofs are included in this chapter. Chapter four will wrap up the thesis and will inform the reader of the work that is currently being done on this thesis as well as plans for future work on this thesis. After the conclusion, there is an appendix which contains all of the graphs which were discussed within the paper.

The graph numbers which are seen throughout this thesis correspond to McKay's listing of the connected graphs on 9 vertices. There are 261080 connected graphs on 9 vertices. McKay gives a one to one relation between each number from 1 and 261080 to a connected graph on 9 vertices. The numbers correspond to which graph or graphs we are considering.

Previous to this study, Chris Morris wrote his thesis on the classification of connected graphs on 9 vertices. For each graph, he determined if it was intrinsically knotted, not intrinsically knotted, or indeterminate (meaning he was unable to verify whether or not the graph was intrinsically knotted).

The program Chris wrote to determine the intrinsically knotted property would test each graph against a series of classification tests. The tests which were used included: Null Classification, Absolute Size Classification, Contains Minor Classification, Minor of Classification, Order Classification, Planarity Classification, and Relative Size Classification. Of these only the Contains Minor Classification and Relative Size Classification would show a graph to be intrinsically knotted, while the others showed that a graph was not intrinsically knotted. Both the Contains Minor Classification and the Minor of Classification tests are executed as many as seven times per connected graph on 9 vertices, once for each of these graphs to be tested: K7, K3311, H8, H9, B9, A9, and F9.

Each connected graph on 9 vertices had successive classification tests applied to it. Once a classification test determined whether the graph was intrinsically knotted or not intrinsically knotted the program would terminate for that graph, i.e., not run anymore classification tests, and print out the necessary information. The information included the graph number, whether it was intrinsically knotted or not intrinsically knotted, the number of edges, which classification test it passed, as well as the amount of time it took to process that graph. If the graph did not pass any classification test it was labeled as 'Indeterminate' and was included with the output as Indeterminate.

From the output file created by the program, Chris wrote several programs in Ruby to analyze the results. The main program which did the bulk of the work was called ik_summarizer. This program would parse through all of the data in the output file, and print out the raw numbers associated with each classification test (e.g. how many graphs passed that specific test). These numbers are the basis of his results.

His results included those graphs on seven as well as eight vertices, but the results which apply directly to this thesis are on nine vertices. On nine vertices Chris determined that there were 1992 intrinsically knotted graphs as well as 32 graphs which were Indeterminate. The rest of the 261080 graphs were determined to be not intrinsically knotted.

Chapter 2
Project Code

Described above you will find the outline of the work done by Chris Morris which was used as a base to this project. Several modifications were made to his code, and in fact many new programs were developed to assist in determining many of the results for this project. This section will be broken down into two sections, the first being the changes that were made to Chris Morris's project and the second being the code that was written for this project.

Changes:

The most important change that was made to Chris' project is that instead of skipping the classification tests once a graph has been determined to be intrinsically knotted, the program now runs through every classification test to see which ones the graphs pass.

The ik_summarizer program was changed to print the results of the analysis by edges instead of printing the results by passed classification tests. The new output from this program produces an edge by edge breakdown of how many graphs with that number of edges passed each classification test.

New Programs:

ik_summarizer_edges:

To be able to analyze the relations between the minors on graphs with the same number of edges a new program had to be written called ik_summarizer_edges. This program, written in Ruby, takes as input the number of edges to analyze as well as the classification tests which you want the desired graphs to pass, also as an option is to opt out of certain classification tests. Allowing the user to be able to opt out of classification tests is very important to the success of this program. This allows users to get the results from graphs which are A9 but not H8. Whereas without the opt-out option then the user could only retrieve which graphs were A9.

Mathematica:

All of the programs within this section were written in Mathematica.

Graph Printer:

In order to study the graphs in question, a program, Graph Printer, was written which displayed images of both the graph as well as the complement of that graph in Mathematica.

Input: List of graphs on n edges, A_list.
Output: Images of each graph and its complement.

Process input file of graphs on n edges
While there is a graph on n edges, call it A, to be processed
    Remove the descriptive information

Construct graph from list of edges
Use ShowGraph function to print graph
Construct complement of graph from list of edges
Use ShowGraph function to print the complement of the graph
Repeat until all graphs are printed.

Subgraph Comparison Test:

Perhaps the most important aspect of the project involves the subgraph comparison test. The subgraph comparison test, which is really finding the subgraphs within the complements of the selected graphs, meaning it is really determining the supergraphs of the given graphs. Two different lists are input into the test, both lists contain a list of graphs and their edges, but the first list only contains graphs with n edges while the second list contains graphs with n+1 edges.

Input: List of graphs on n edges, A_list; list of graphs on n+1 edges, B_list.
Output: List of graphs from B_list that have an A_list graph as a subgraph.

Process input file of graphs of n edges, A_list
While there is a graph on n edges, call it A, to be processed
Remove the descriptive information
Create the complement of the graph on n edges, call it A'
Remove list of edges from input

Delete each edge in graph A' one at a time, so as to only have one fewer edge on graph A', and one greater edge on A, at any given time. Call this graph A''.

Process input file of graphs on n+1 edges, B_list
While there is a graph on n+1 edges, call it B, to be processed
Remove the descriptive information
Create the complement of the graph on n+1 edges, call it B'

If B' is isomorphic to A''
Insert B' Graph ID into list of subgraphs
If B' is not isomorphic to A''
Do not insert B' Graph ID into list of subgraphs
Repeat until all n+1 edge graphs are compared to all n edge graphs.

Find Subgraph (Supergraph):

In order to determine whether a graph was A9 or H8, this program was written. The program does its work on the complements of the graphs in order to shorten the execution time. Three different lists are input into this program, all of the lists contain a list of graphs and their edges. The first list only contains graphs with n edges, while the second list contains graphs with n+2 edges. The output of this program is a graphic showing which edges were missing from the original graph complement as well as a comparison image of the original graph complement. Also included in the output is the id number of the graph as well as the id number corresponding to the subgraph.

Input: List of graphs on n edges, A_list; list of graphs on n+2 edges, B_list.
Output: Image of original graph complement and its id, image of the subgraph complement with highlighted edges and its id.

```
Process input file of graphs on n edges, A_list
While there is a graph on n edges, call it A, to be processed
    Remove the descriptive information
    Create the complement of the graph on n edges, call it A'
    Remove list of edges from input

    Delete two different edges in graph A', so as to only have a graph A' with two
    fewer edges, and graph A has two greater edges, at any given time.  Call this
    graph A''.  Do this until all possibilities to delete edges has been exhausted.

    Process input file of graphs on n+2 edges, B_list
    While there is a graph on n+2 edges, call it B, to be processed
        Remove the descriptive information
        If B has not already been matched with a subgraph
            Create the complement of the graph on n+2 edges, call it B'

        If B' is isomorphic to A''
            Insert B' Graph ID into list of subgraphs
            Determine edges which are missing from A'' in relation to B'
            For each missing edge
                Set edgeColor to Red
            Print ID of B' and A''
            Print Image of B' and A''
        If B' is not isomorphic to A''
            Do not insert B' Graph ID into list of subgraphs
    Repeat until all n+2 edge graphs are compared to all n edge graphs.
```

Chapter 3
Results

# Edges	# Total Graphs	# H8 Graphs	# A9 Graphs	# A9 & H8 Graphs	# A9 Not H8 Graphs
22	21	13	1	0	1
23	120	71	6	1	5
24	257	202	28	15	13
25	388	353	80	66	14
26	413	397	154	145	9
27	345	341	197	194	3
28	222	221	166	165	1
29	122	122	106	106	0
30	58	58	54	54	0

Graph Summary 1: Only Intrinsically Knotted Graphs Counted

The following propositions and lemmas are statements that have been verified by computer. Those that have not been mathematically proven are proposed as questions or problems for others to come up with the mathematical proofs for them.

Proposition 1:

The 13 H8 Graphs with 9 vertices on 22 edges are:

143978, 144004, 144007, 145557, 145656, 243809, 243810, 243823, 243969, 244553, 244556, 244631, 244831.

See appendix for figures of these graphs. The proof is omitted.

Lemma 1.1: Let G be a connected graph on 9 vertices. If G is H8, then it has one of the 13 graphs in Proposition 1 as a subgraph.

Proof: As H8 has 8 vertices and G has 9, we can find the H8 minor by a sequence of edge deletions giving the graph G' followed by a single edge contraction on G'. Also, by removing enough edges beforehand, we can ensure that the edge contraction results in the loss of a single edge. Thus, G' is a graph of 22 edges since contracting the edge of G' results in the loss of a single edge and also leaves us with the 21 edge graph H8. In other words, G' is one of the 13 graphs in Proposition 1.

So, by doing only edge deletions, we find G', one of the 13, as a subgraph of G.
☑

Proposition 2:

The only A9 and H8 Graph with 9 vertices on 23 edges is:

243976.

Proposition 3:

The 15 A9 and H8 Graphs with 9 vertices on 24 edges are:

243977, 244006, 244021, 244023, 244024, 244081, 244645, 244706, 244847, 244863, 245286, 255327, 255349, 255398, 255930.

Remark:

On 24 edges there is a disparity between the number of graphs found using the java program and the number of graphs found when taking the A9 and H8 graph found on the 23 edges (Graph 243976) and finding which graphs are super graphs of that 23 edge graph. This is why E > 24 in Lemma 7.3.

There are 9 graphs which pass the subgraph comparison test (these are the graphs that have 243976 as a subgraph), these are:

243977, 244021, 244024, 244847, 244863, 245286, 255327, 255349, 255398.

There are 6 graphs which do not pass the subgraph comparison test, these are:

244006, 244023, 244081, 244645, 244706, 255930.

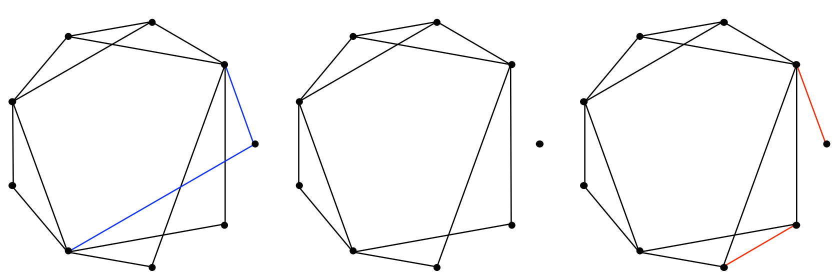
See appendix for figures of these graphs.

Proof:

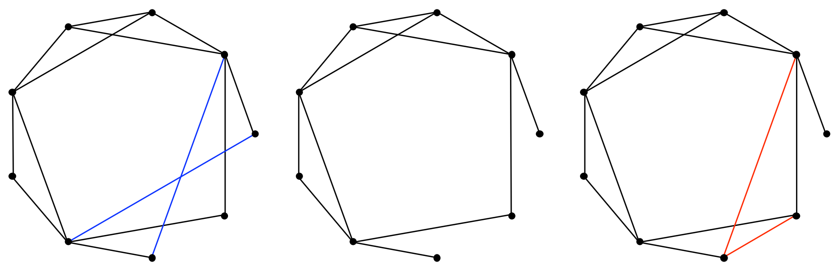
The plan is to first argue that each of the 15 graphs are A9 and H8. As in the remark, there are two cases to look at.

Below are representations of the complements of the 9 A9 and H8 graphs on 24 edges which pass the subgraph comparison test, meaning they contain Graph 243976. The left most graph is the H8 representation of the graph on 9 vertices, the blue edges signify the edges which are not present in the complement of that A9 and H8 graph. Listed next to the Graph number is the H8 graph on 22 edges which is associated with that graph. The middle graph is the complement of the A9 and H8 graph on 9 vertices. The right most graph is the A9 representation of the graph, the red edges signify the edges which are not present in the complement of that A9 and H8 graph.

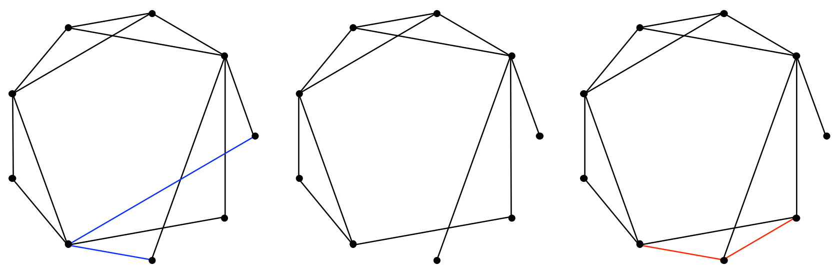
Graph 243977 with H8 Graph 243969



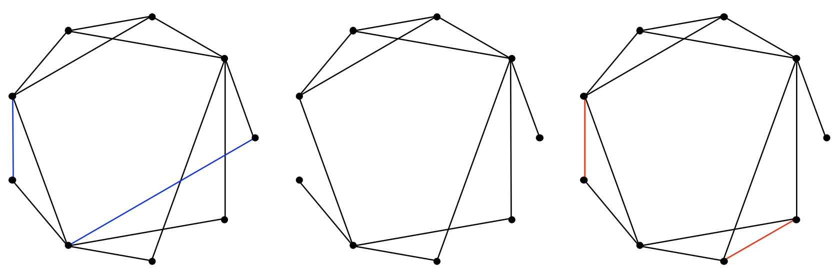
Graph 244021 with H8 Graph 243969



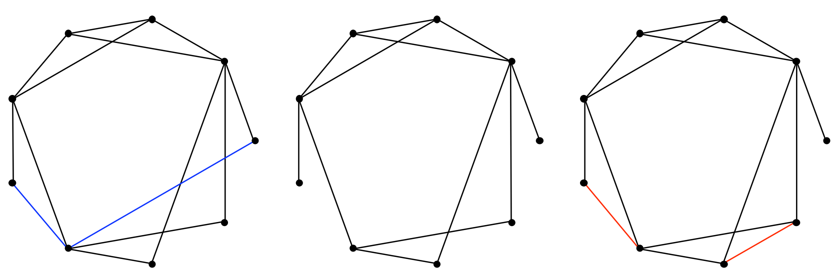
Graph 244024 with H8 Graph 243969



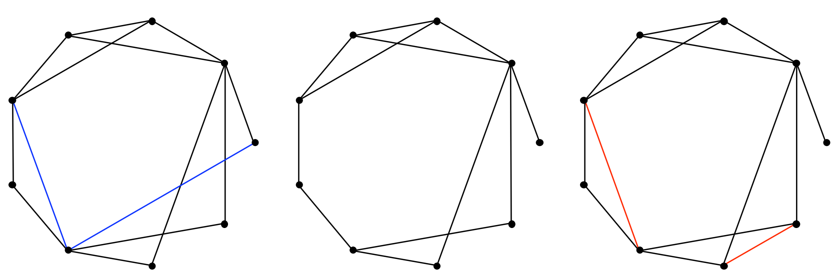
Graph 244847 with H8 Graph 243969



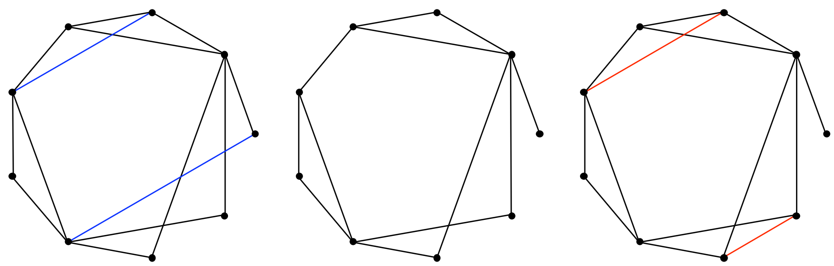
Graph 244863 with H8 Graph 243969



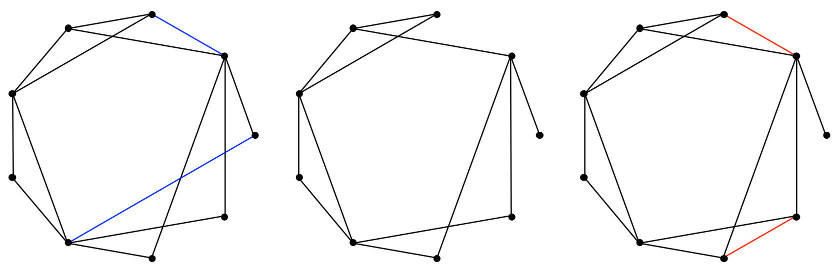
Graph 245286 with H8 Graph 243969



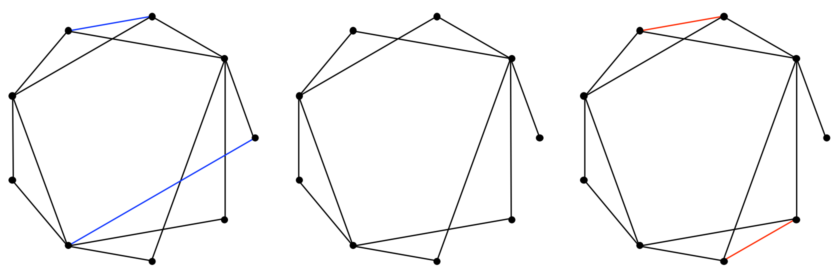
Graph 255327 with H8 Graph 243969



Graph 255349 with H8 Graph 243969

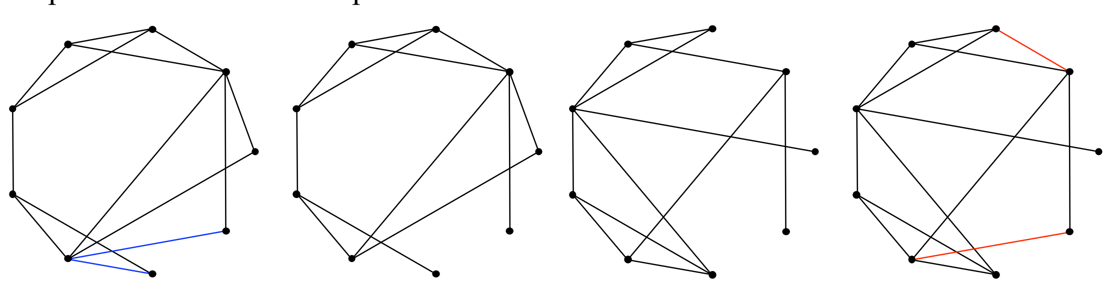


Graph 255398 with H8 Graph 243969

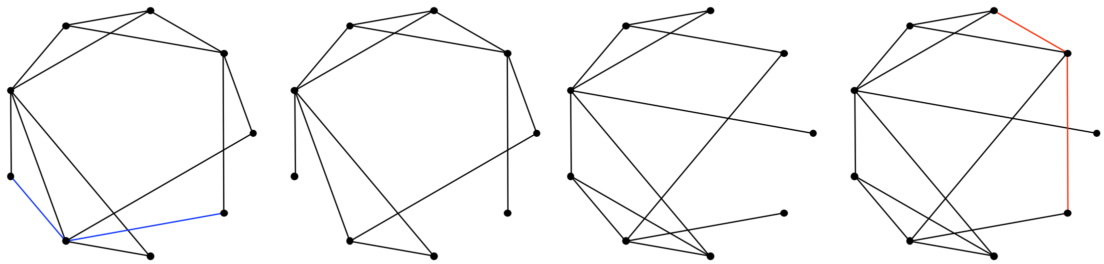


Below are representations of the complements of the 6 A9 and H8 graphs on 24 edges which did not pass the subgraph comparison test, meaning they do not contain Graph 243976. The left most graph is the H8 representation of the graph on 9 vertices, the blue edges signify the edges which are not present in the complement of that A9 and H8 graph. Listed next to the Graph number is the H8 graph on 22 edges which is associated with that graph. The two middle graphs are two different representations of the complement of that A9 and H8 graph on 9 vertices, with the left middle corresponding to the H8 graph and the right middle corresponding to the A9 graph. The right most graph is the A9 representation of the graph, the red edges signify the edges which are not present in the complement of that A9 and H8 graph.

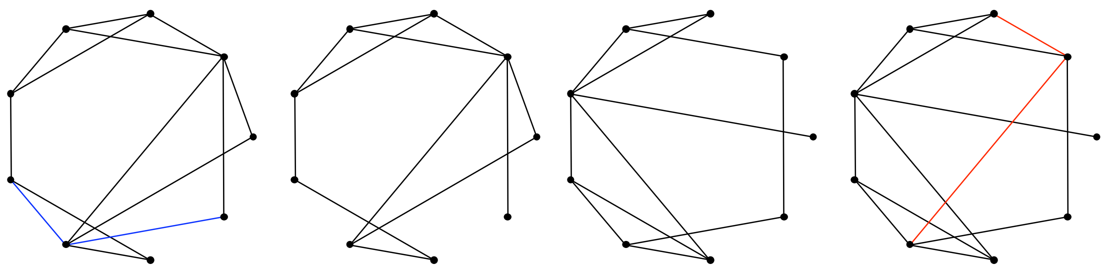
Graph 244006 with H8 Graph 244631



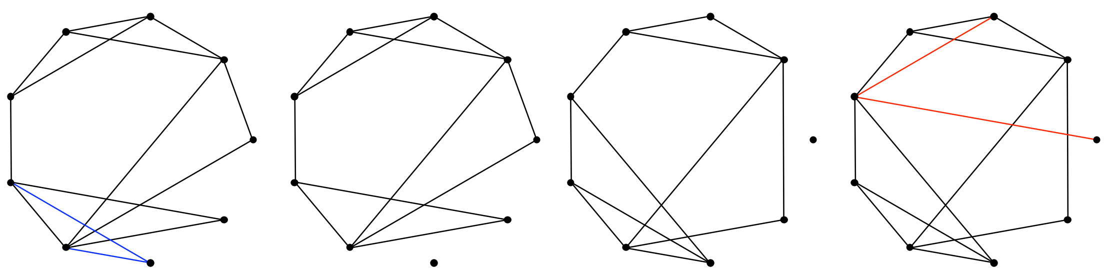
Graph 244023 with H8 Graph 243823



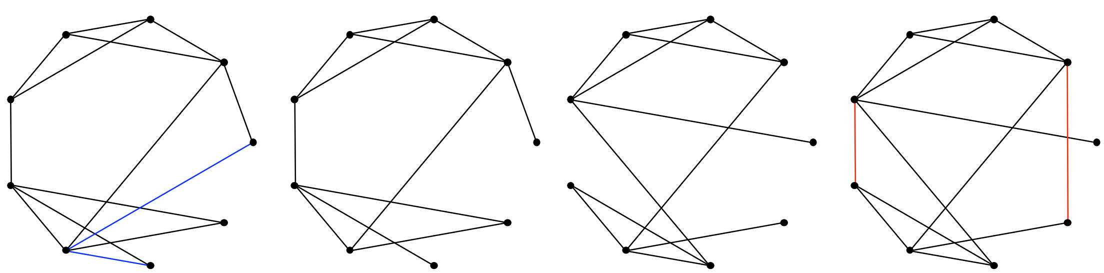
Graph 244081 with H8 Graph 244631



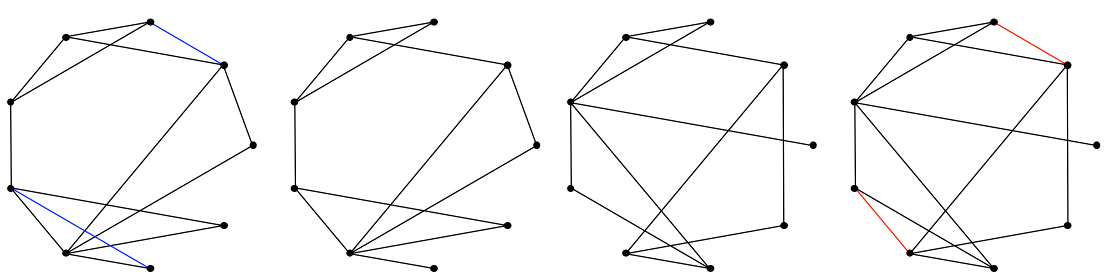
Graph 244645 with H8 Graph 244556



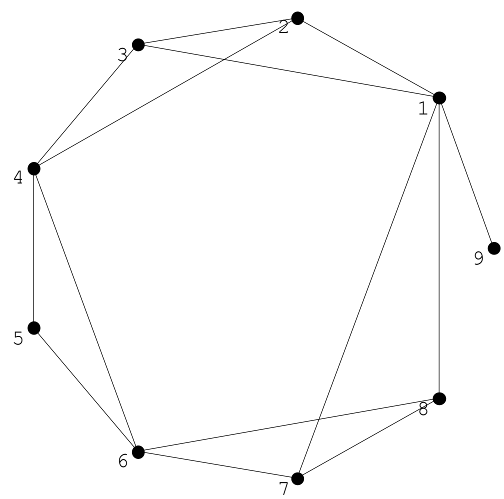
Graph 244706 with H8 Graph 244556



Graph 255930 with H8 Graph 244556



Simplified Graph of the complement of A9:



The next step is to argue that if G is a graph on 24 edges that is both A9 and H8, then G must be one of the 15 graphs given above.

Suppose G has 24 edges and is A9. Then G' is obtained by removing two edges from A9's complement. So, consider the ways to remove two edges.

There are six types of edge in A9's complement (see figure above), represented by [5,6], [4,6], [6,7], [7,8], [1,7] and [1,9].

First note that if G' is obtained from A9 by removing [7,8] (or its analogue [2,3]) then G is one of the 15 graphs above. And further, it will be one of the 9 that pass the subgraph comparison test. Indeed, on removing [7,8], then there are 9 ways to remove a further edge: [4,5], [5,6], [4,6], [6,7], [2,4], [2,3], [1,2], [1,7], or [1,9]. These are exactly the 9 graphs that pass the subgraph comparison test:

Type [7,8] with [1,9]: This is Graph 243977.
Type [7,8] with [1,2]: This is Graph 255349.
This is symmetric to: [7,8] with [1,3].
Type [7,8] with [2,4]: This is Graph 255327.
This is symmetric to: [7,8] with [3,4].
Type [7,8] with [1,7]: This is Graph 244021.
This is symmetric to: [7,8] with [1,8].
Type [7,8] with [6,7]: This is Graph 244024.
This is symmetric to: [7,8] with [6,8].
Type [7,8] with [4,6]: This is Graph 245286.
Type [7,8] with [4,5]: This is Graph 244847.
Type [7,8] with [5,6]: This is Graph 244863.
Type [7,8] with [2,3]: This is Graph 255398

Henceforth, assume G' is obtained by removing two edges other than [2,3] or [7,8] from A9'. The goal is to show that either G is not H8 or else it is one of the 15 graphs (in fact, one of the 6 that are not given by subgraph comparison) .

In removing two edges, it is possible to either remove two edges of one type or two edges of two different types. The argument for two edges of one type follows. The case of two edges of different types will be left for future researchers. However, the proof will conclude with a listing of how the 6 graphs occur.

Two of one type:

This is not possible for types [4,6] and [1,9] as there is only one edge of these types.

The possibilities for this are: [5,6] with [4,5], [6,7] with [6,8], [6,7] with [3,4], [1,7] with [1,8], and [1,7] with [1,2].

Type [5,6] with [4,5]:

This graph, call it G, has no H8 minor as will now be demonstrated. G is a connected graph on 24 edges whose complement G' has degree sequence (5,3,3,3,3,3,3,1,0). If G were H8, it would have one of the 13 H8 graphs on 22 edges as a subgraph. However, on adding two edges to G', the degree sequence will contain at least seven vertices of degree three or more. None of the 13 H8 graphs have such a degree sequence, so G is not H8.

Type [6,7] with [6,8]:

This is symmetric to: [2,4] with [3,4].

This graph, call it G, has no H8 minor as will now be demonstrated. G is a connected graph on 24 edges whose complement G' has degree sequence (5,4,3,3,2,2,2,2,1). This graph has two cut vertices, 1 and 4. Each of the 13 H8 graphs has a complement with at most two cut vertices. The strategy is to investigate graphs formed by adding two edges to G' and resulting in two, one, or zero cut vertices

Suppose adding two edges to G' results in an H8 graph whose complement has two cut vertices. Among the complements of these graphs, in each case when the two vertices are removed, there is a K3 as a component of the resulting graph. Now the cut vertices must still be the vertices 1 and 4. However, notice that in G'-{1,4} there's no way to add edges to make a K3 component while maintaining 1 and 4 as cut vertices.

Next, add two edges to G' resulting in an H8 graph G2 whose complement has a single cut vertex. There are two such graphs, both having a K3 component in the graph that results from removing the cut vertex.

If the remaining cut vertex is vertex 4, note that G'-{4} consists of two components, one being a K2. There's no way to add two edges such that G2'–{4} has a K3 component. If, instead, the cut vertex is 1, then, G'-{1} again has a K2 component that could become a K3 by the addition of two edges. However, this would leave 4 as a cut vertex, contradicting the assumption that G’’ was formed with a single cut vertex.

Finally, assume adding two edges to G’ gives an H8 graph with no cut vertices. This means, (using symmetry) one of the edges is [5,7], [5,9], or [7,9]. Consider these three cases in turn.

Add [5,7] to form the graph G1. Then G1 has diameter 4 while the 13 H8 complements all have diameter at most 3. To force G1 to have diameter 3, the second edge added is [6,9]. However, adding the two edges [5,7] and [6,9] to G’ does not result in the complement of an H8 graph. So, it’s not possible to form an H8 graph in this way.

Similarly, adding [5,9] or [7,9] leads to a graph G1 of diameter 4. There is exactly one way to make the graph from [5,9] have diameter 3 and two possibilities for the graph from [7,9] (ADD the edges here). However, none of the three 22 edge graphs obtained in this way are H8. So, again it’s not possible to form an H8 graph this way. This completes the argument for graphs formed by removing type [6,7] and [6,8].

Type [1,7] with [1,8]: This graph has no H8 minor.

This is symmetric to: [1,2] with [1,3].

This graph, call it G, has no H8 minor as will now be demonstrated. Note that G’ has degree sequence (4,4,3,3,3,2,2,2,1) and three cut vertices: 1, 4, and 6. By adding two edges to G’, as mentioned in the last case, the 13 H8 graphs have complements with at most two cut vertices. However, the two graphs that have a single cut vertex also have a vertex of degree 7 and there’s no way to form such a graph by adding two edges to G’. It remains to consider H8 graphs having 0 or 2 cut vertices.

Further, among the H8 complement graphs, there’s a single graph that has a degree 1 vertex, the others all having minimal degree 2. Moreover, the single exception also has a vertex of degree 7. We conclude that one of the edges added to G’ to form the complement of an H8 graph is incident to vertex 9. This means that vertex 1 is no longer a cut vertex.

Suppose adding two edges to G’ results in an H8 graph whose complement has two cut vertices. As above, 1 is no longer a cut vertex, so the two cut vertices are 4 and 6. Among the complements of these graphs, in each case when the two vertices are removed, there is a K3 as a component of the resulting graph. The only way to get a K3 from G’-{4,6} is by adding [5,7] and [5,8]. However, we've already argued that one of the edges added must be incident to vertex 9. So, obtaining a graph with two cut vertices is not possible.

Since the graph G does not have two cut vertices, the only possibility left is for G to have no cut vertices. This narrows down the number of H8 graphs from 13 to 7 left to consider. In order to obtain a graph by adding two edges to G’ with degree sequence (6,5,…), G’ must have an edge added which connects the vertices with degree 4. In G’ though, the two vertices with degree 4 are already connected, therefore there is no way to obtain a (6,5,…) degree sequence from G’ when only adding two edges. Now there are only 5 H8 graphs left to consider.

Now consider adding two edges to G' such that the resulting degree sequence is (5,5,4,…). In order to obtain this degree sequence and no cut vertices, edge [4,9] or [6,9] must be added (adding both does not result in the desired degree sequence). Consider these two additions separately:

Firstly adding edge [4,9]: The only way to obtain a degree sequence of (5,5,4,…) is to add edge [2,6] or [1,6], but contrary to the assumption there still exists a cut vertex at vertex 6.

Next consider adding edge [6,9]: To obtain the degree sequence of (5,5,4,…) only edge [1,4] can be added since [2,4] and [3,4] are already edges in G', but when edges [6,9] and [1,4] have been added there still exists a cut vertex at vertex 6. This is contrary to the assumption that there are no cut vertices. So therefore H8 graphs with the degree sequence of (5,5,4,…) cannot be obtained from G' by adding two edges. Now there are only 3 H8 graphs left to consider.

Type [1,7] with [1,2]: This graph has no H8 minor.
This is symmetric to: [1,7] with [1,3], [1,8] with [1,2], and [1,8] with [1,3].

Type [6,7] with [3,4]: This graph is one of the 15 shown above, Graph 255930.
This is symmetric to: [6,7] with [2,4], [6,8] with [3,4], and [6,8] with [2,4].

And here's a list of how the other five occur:
Graph 244006 is of type [6,7] with [4,5].
Graph 244023 is of type [6,7] with [5,6].
Graph 244081 is of type [6,7] with [4,6].
Graph 244645 is of type [1,7] with [1,9].
Graph 244706 is of type [1,7] with [5,6].
☑

Remark: All of the graphs in Proposition 3 are obtained by removing an edge incident to vertex 7 (or, equivalently, vertex 2, 3, or 8). Perhaps there is a more direct argument that takes advantage of this observation.

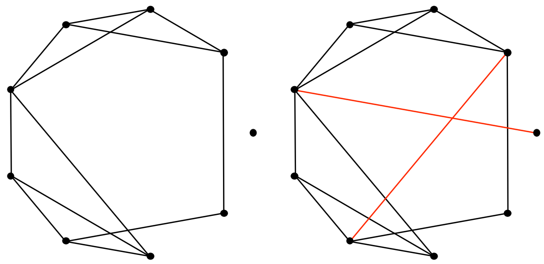
Proposition 4:
The one A9 and not H8 graph with 9 vertices on 22 edges is:
244448.

Proposition 5:
The 5 A9 and not H8 graphs with 9 vertices on 23 edges are:
243691, 243694, 244449, 244469, 244642.

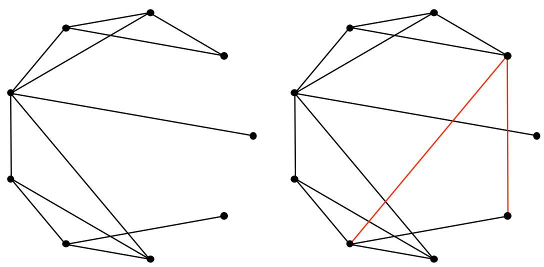
Proposition 6:
The 13 A9 and not H8 Graphs with 9 vertices on 24 edges are:
243692, 243748, 243966, 244020, 244043, 244083, 244279, 244471, 244475, 244714, 245619, 255932, 255966.

Below are representations of the complements of the 13 A9 and not H8 graphs on 24 edges. The left graph is the complement of the A9 and not H8 graph on 9 vertices. The right graph is the A9 representation of the graph; the red edges signify the edges which are not present in the complement of that A9 and not H8 graph.

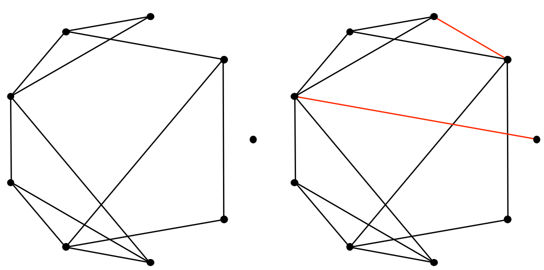
Graph 243692



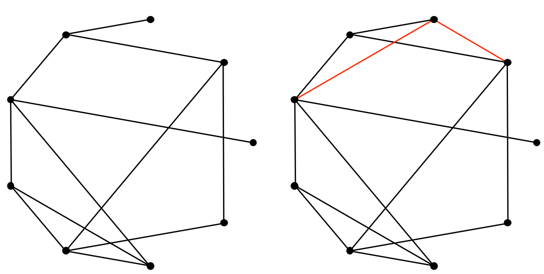
Graph 243748



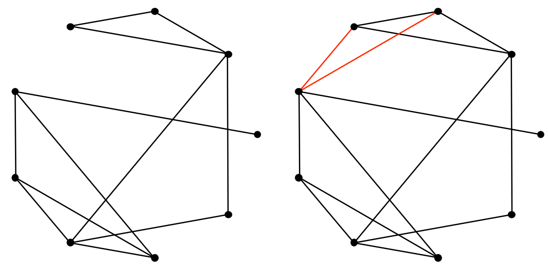
Graph 243966



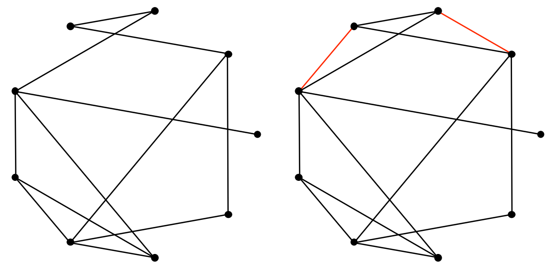
Graph 244020



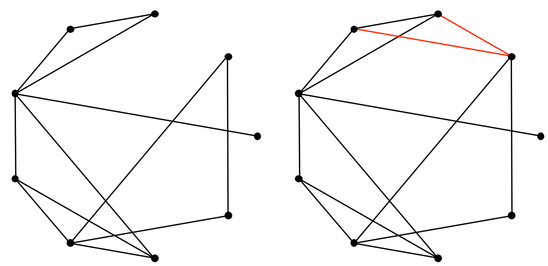
Graph 244043



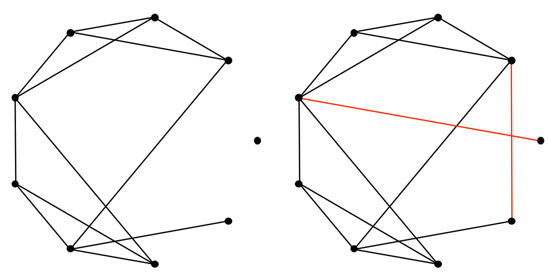
Graph 244083



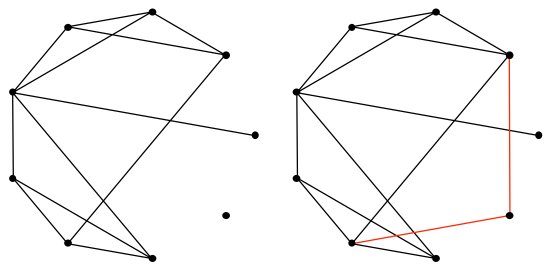
Graph 244279



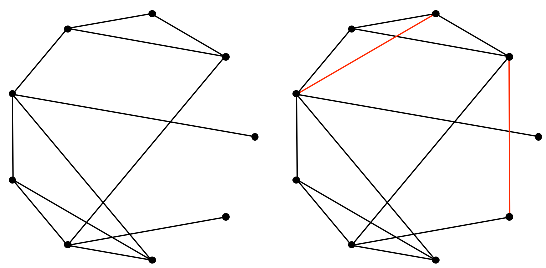
Graph 244471



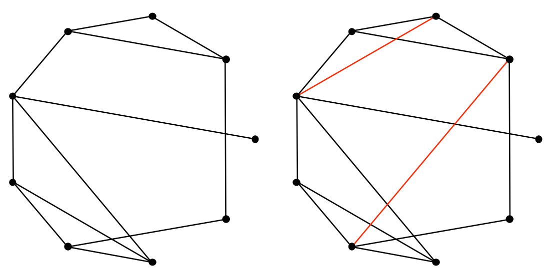
Graph 244475



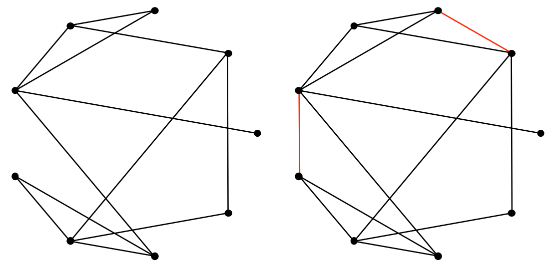
Graph 244714



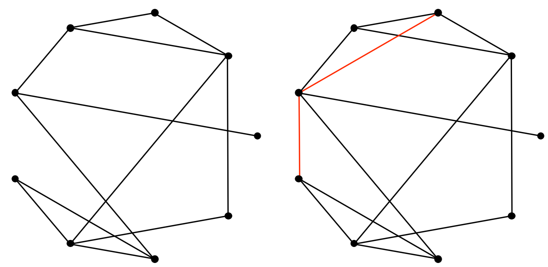
Graph 245619



Graph 255932



Graph 255966



Proposition 7:

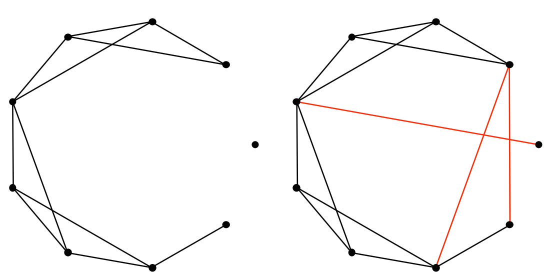
The 14 A9 and not H8 Graphs with 9 vertices on 25 edges are:

243750, 243761, 244088, 244281, 244287, 244477, 245175, 245193, 248474, 248513, 248652, 255226, 255227, 256310.

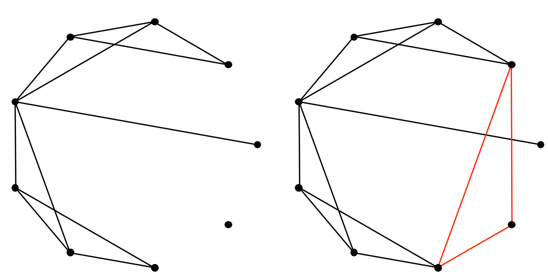
Proof:

Below are representations of the complements of the 14 A9 and not H8 graphs on 25 edges. The left graph is the complement of the A9 and not H8 graph on 9 vertices. The right graph is the A9 representation of the graph, the red edges signify the edges which are not present in the complement of that A9 and not H8 graph.

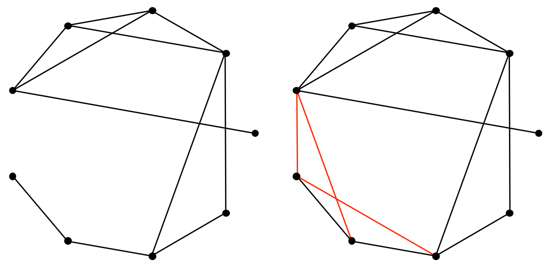
Graph 243750



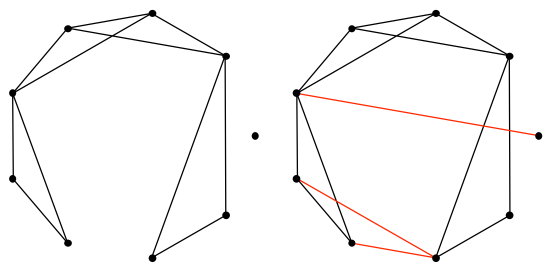
Graph 243761



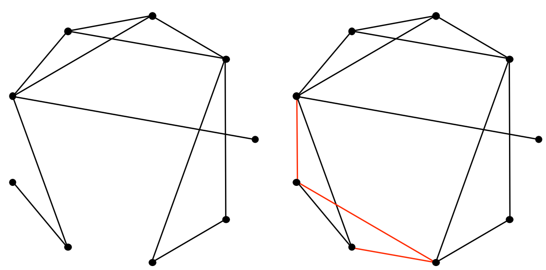
Graph 244088



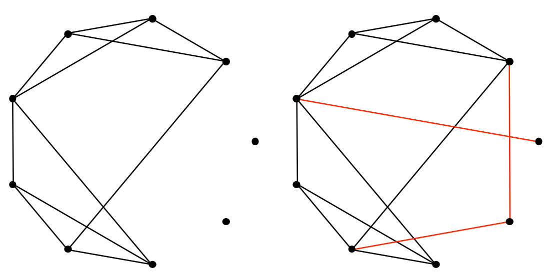
Graph 244281



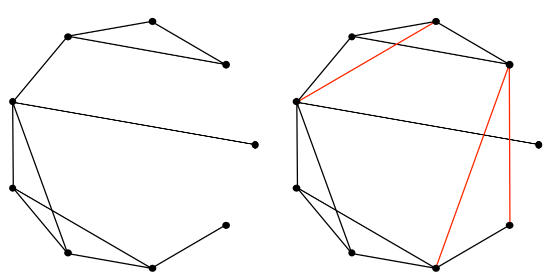
Graph 244287



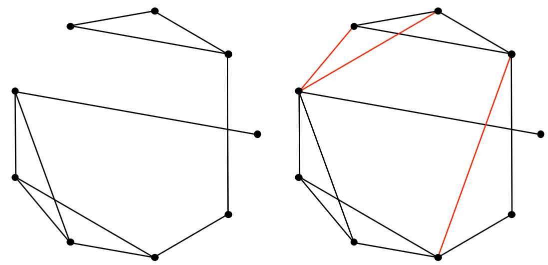
Graph 244477



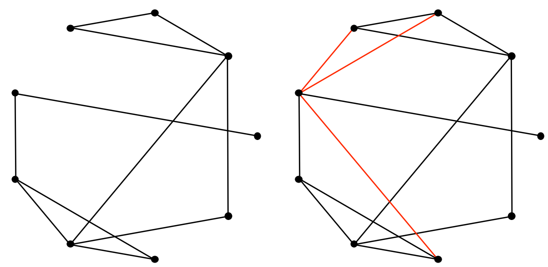
Graph 245175



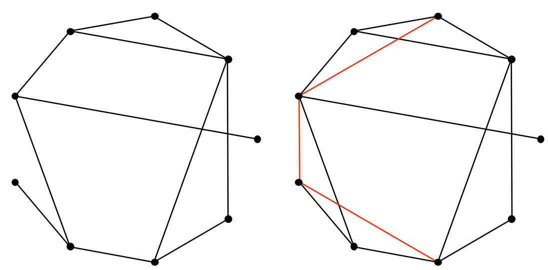
Graph 245193



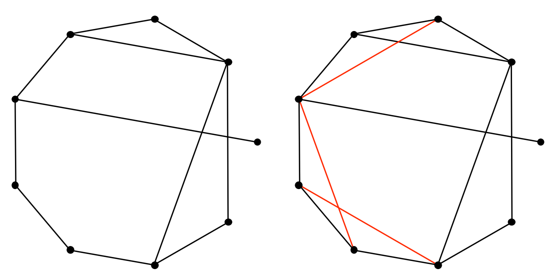
Graph 248474



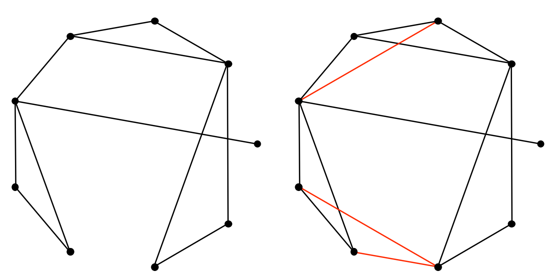
Graph 248513



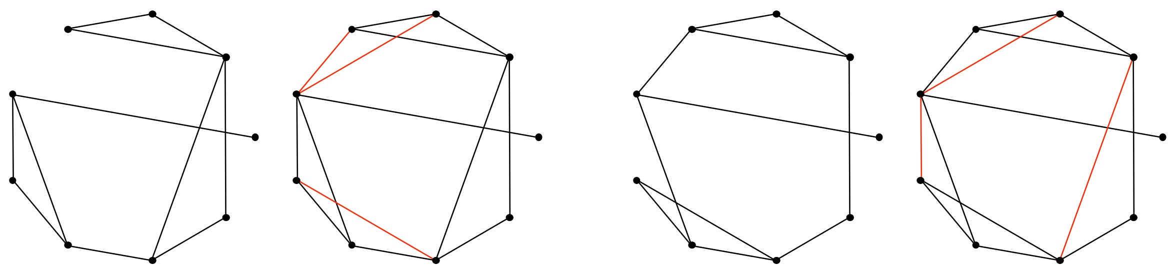
Graph 248652



Graph 255226



Graph 255227



Graph 256310

Step 2:

Suppose G has 25 edges and is A9, then G either:

a. Has H8 as a minor.
b. We want to show G is one of the 14.

Lemma 7.1: Let G be a connected graph on 9 vertices. If G is A9, then G has A9 as a subgraph.

Proof: Since G and A9 both have 9 vertices, there are no edge contractions in forming the minor A9. So A9 is simply a subgraph.
☑

Lemma 7.2: Let G be a connected graph on 9 vertices. If G has n^{3} 23 edges and is A9 then G has a subgraph G’ on n-1 edges which is also A9.

Proof: Because A9 also has nine vertices, A9 is in fact a subgraph of G. So G has a subgraph with 22 edges which is the A9 graph itself. Since G has n^{3} 23 edges, there is an edge of G not in the subgraph. Delete it to obtain G’ which is also A9.
☑

Lemma 7.3: If G is A9 and H8 and has n^{3} 25 edges then G has an A9 and H8 subgraph G’ on n-1 edges.

Proposition 8:

The 9 A9 and not H8 Graphs with 9 vertices on 26 edges are:

243763, 244295, 248443, 248491, 248538, 248542, 248659, 255595, 256453.

Proposition 9:

The 3 A9 and not H8 Graphs with 9 vertices on 27 edges are:

248669, 248767, 255613.

Proposition 10:

The only A9 and not H8 Graph with 9 vertices on 28 edges is:

248770.

Proposition 11:

All of the A9 graphs with 9 vertices on 29 edges or 30 edges are also H8.

Theorem:

Let G have an A9 subgraph and 22 £ E £ 30, where E is the number of edges. Either G also has an H8 minor or else G is one of the graphs above (in Propositions 4 to 10) with A9 minor and no H8 minor.

Chapter 4
Conclusion and Open Questions

In order to divide the intrinsically knotted graphs into smaller and smaller groups, A9 and H8 were chosen as the minor intrinsically knotted graphs because H8 accounts for almost all of the intrinsically knotted graph and because of the small numbers of graphs of the form A9 and not H8. The first graphs analyzed were the 13 H8 graphs on 22 edges; this was proposition one. This was done in order to get a firm hold on where to start analyzing the graphs and to make sure that the base data was correct. The next graphs that were analyzed were those that were A9 and H8, these were propositions two, three, and eleven. This was done because of the earlier remark regarding the difference in number of graphs between the java program and the subgraph comparison test. The final group of graphs that were analyzed were those that are A9 and not H8, these were propositions four through ten. As noted this group was chosen because of the small number of graphs in this group.

The software developed leads to the propositions, four lemmas and theorem stated in Chapter 3. As noted at the beginning of the paper, many of the propositions, lemmas, and theorems have not been mathematically proven. These are left to others as questions to be proven mathematically. These propositions, lemmas, and theorems are believed to be true because the software developed for this project shows that they are true.

Summarize results.

Questions

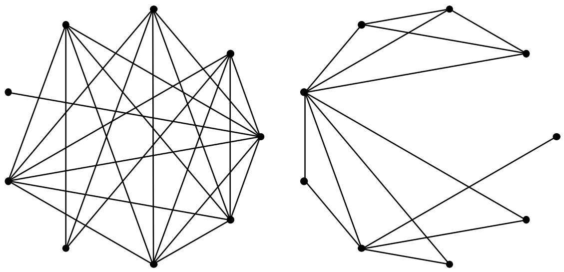
Describe the other families (H8, B9&H8, etc…)
Describe the 32 graphs that are indeterminate (talk about progress on this here).

Appendices

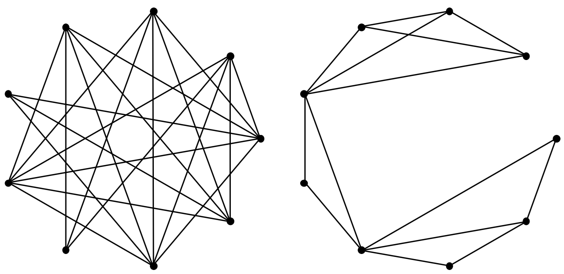
This section will be broken down by the number of edges in each graph as well as by the minors which the graphs contain.

22 Edges, H8:

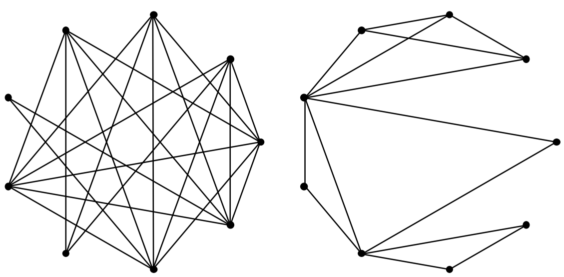
Graph 143978, order 9.



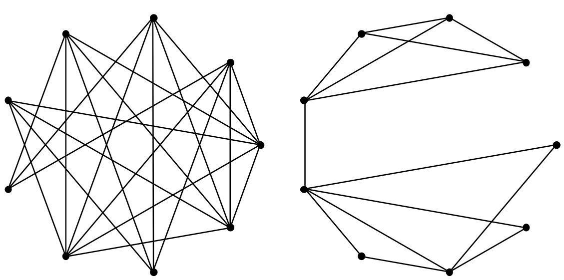
Graph 144004, order 9.



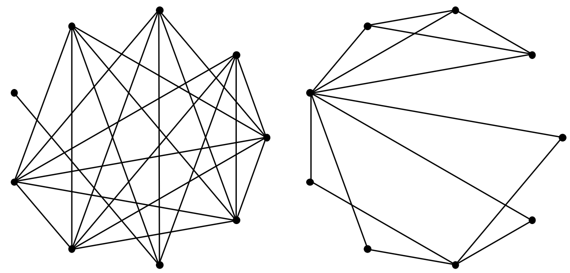
Graph 144007, order 9.



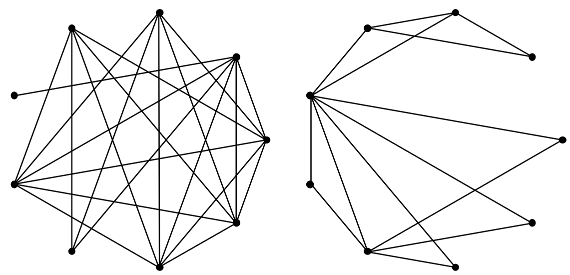
Graph 145557, order 9.



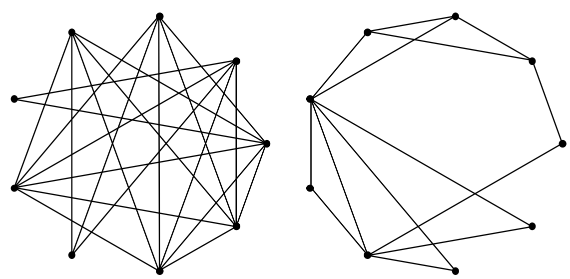
Graph 145656, order 9.



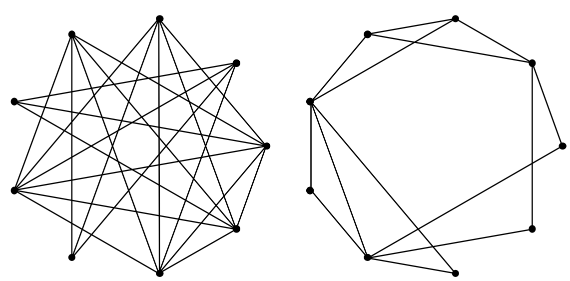
Graph 243809, order 9.



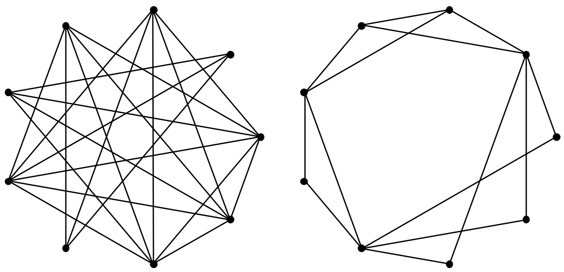
Graph 243810, order 9.



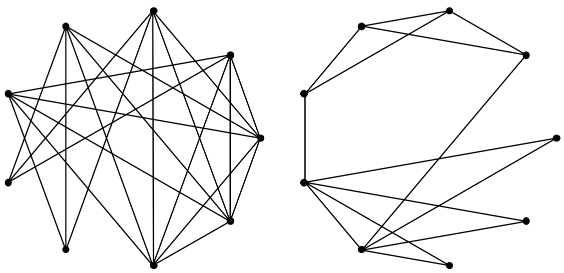
Graph 243823, order 9.



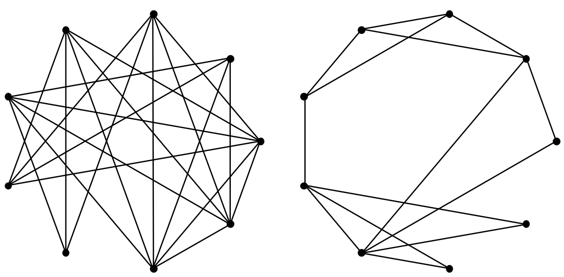
Graph 243969, order 9.



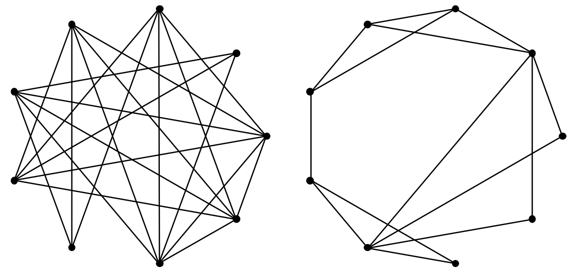
Graph 244553, order 9.



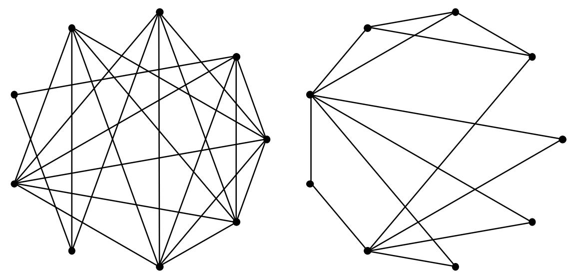
Graph 244556, order 9.



Graph 244631, order 9.



Graph 244831, order 9.

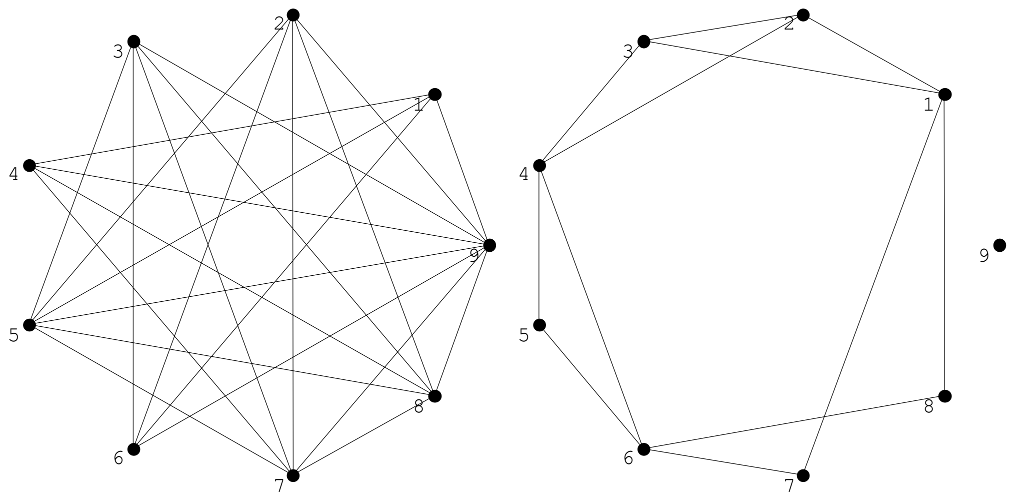


24 Edges, A9 and H8:

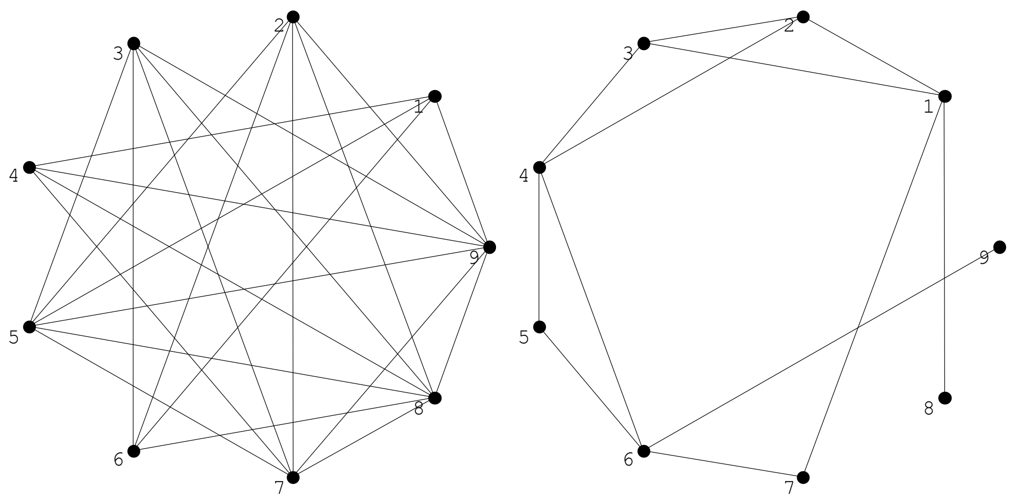
This section will be broken down even further depending on whether the graphs pass the subgraph comparison test or do not.

The 9 graphs which pass the subgraph comparison test:

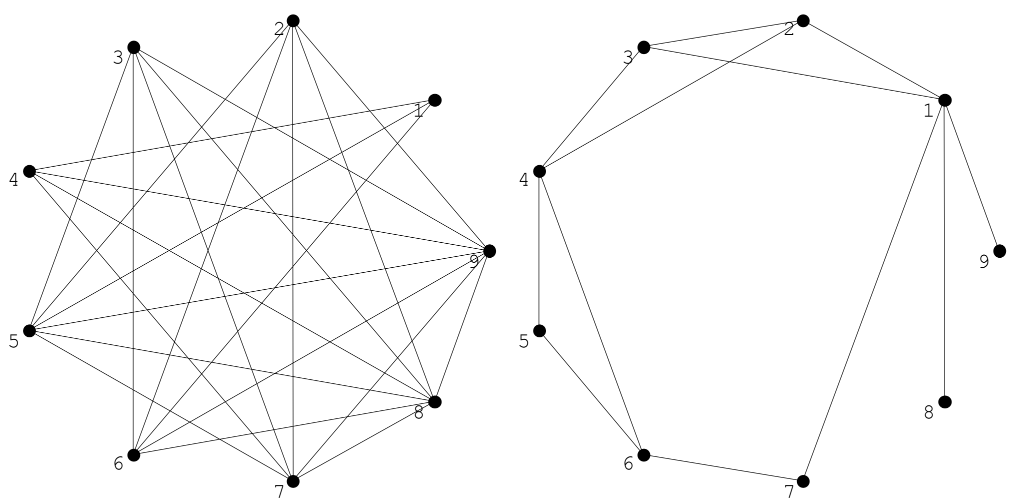
`Graph 243977`



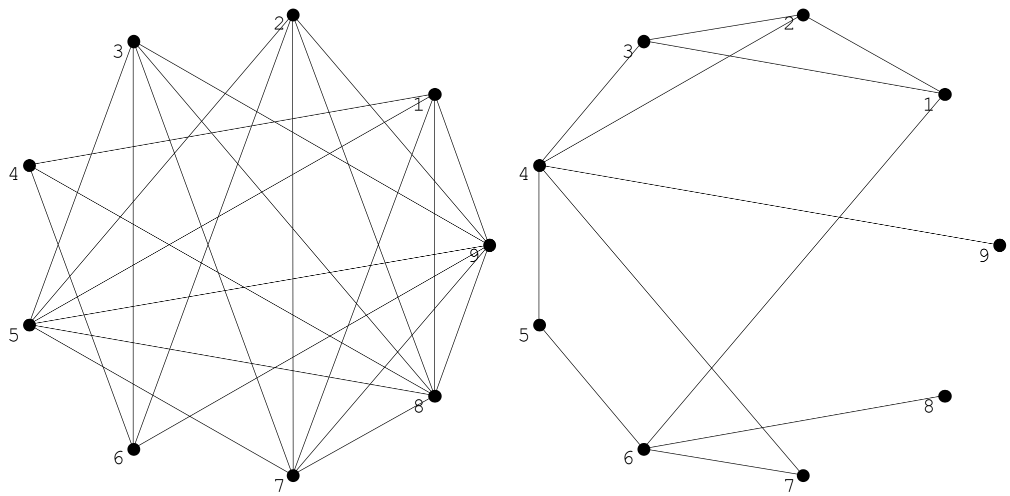
Graph 244021



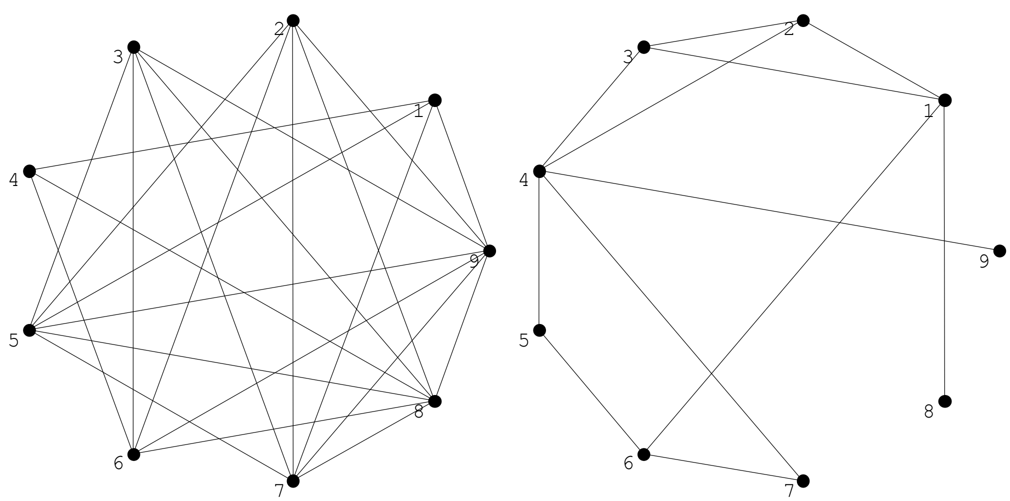
Graph 244024



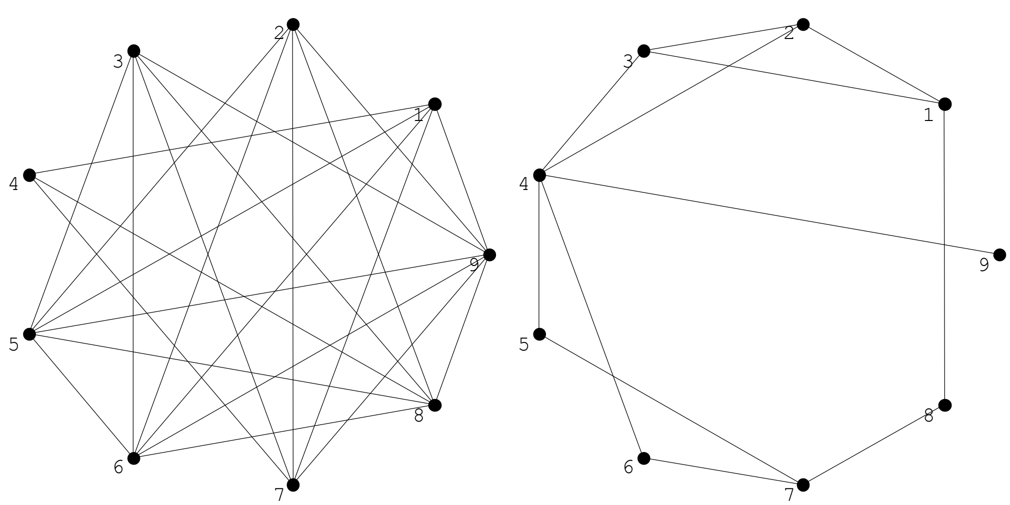
Graph 244847



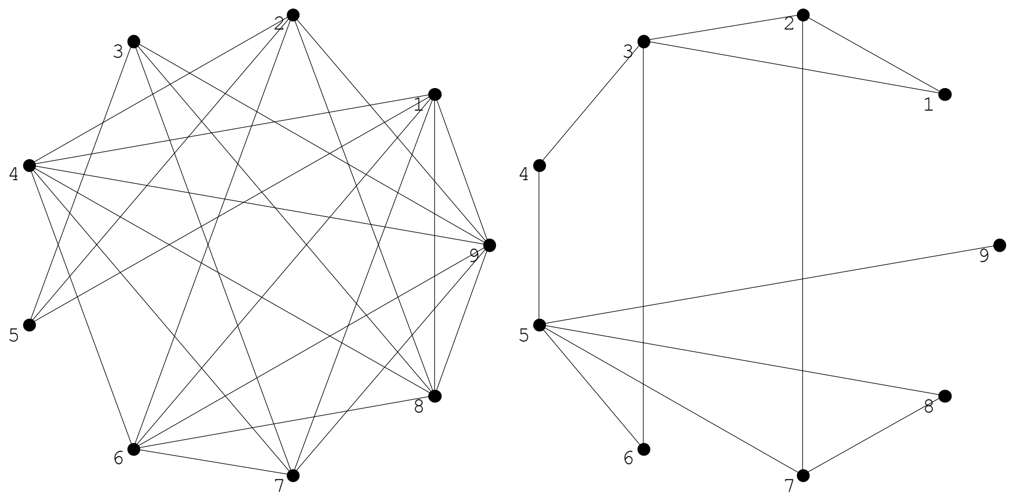
Graph 244863



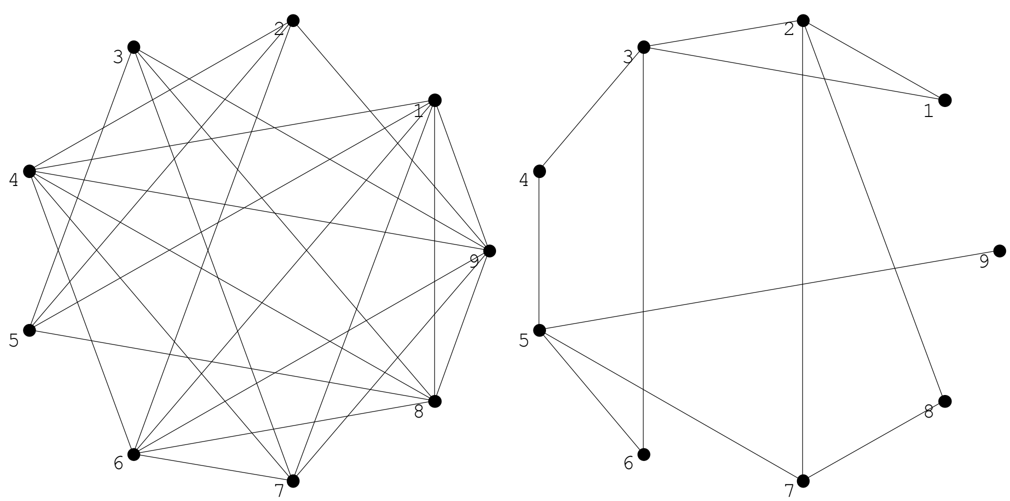
Graph 245286



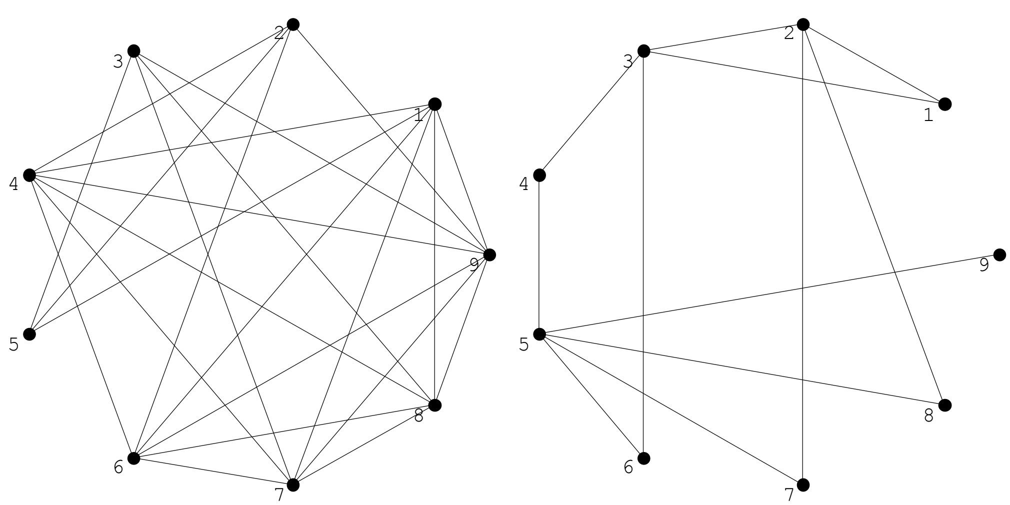
Graph 255327



Graph 255349

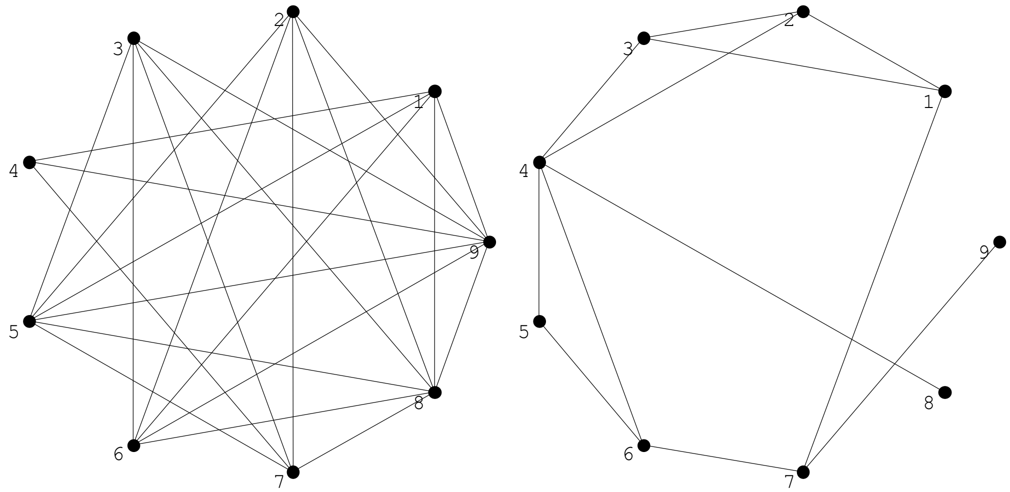


Graph 255398

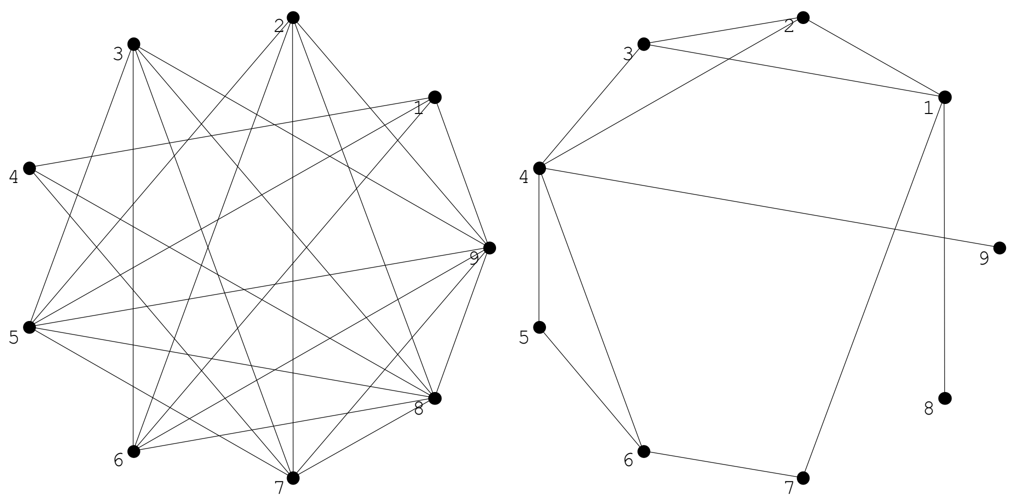


The 6 graphs which do not pass the subgraph comparison test:

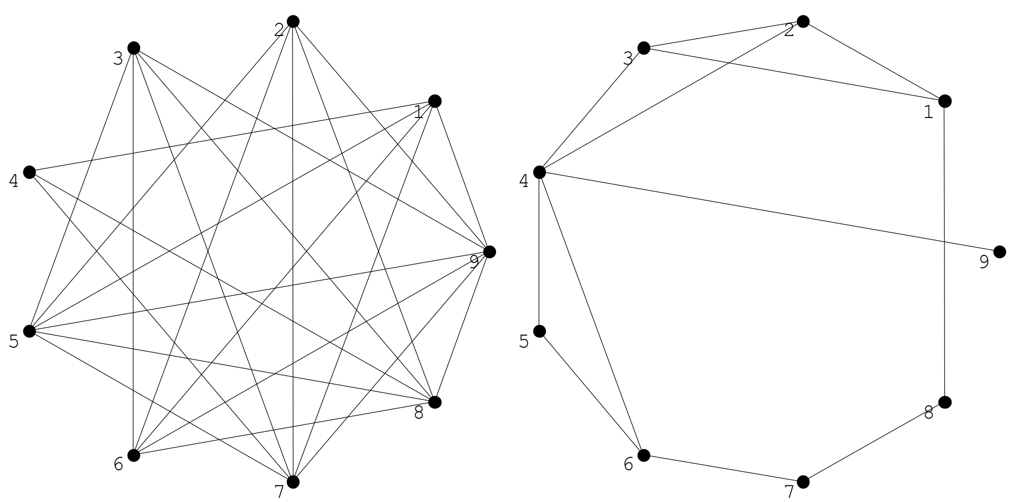
Graph 244006



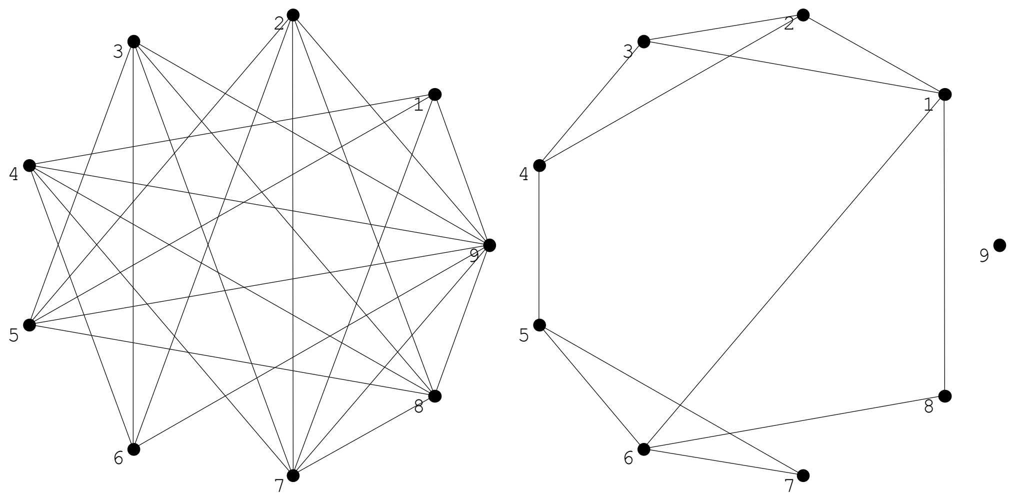
Graph 244023



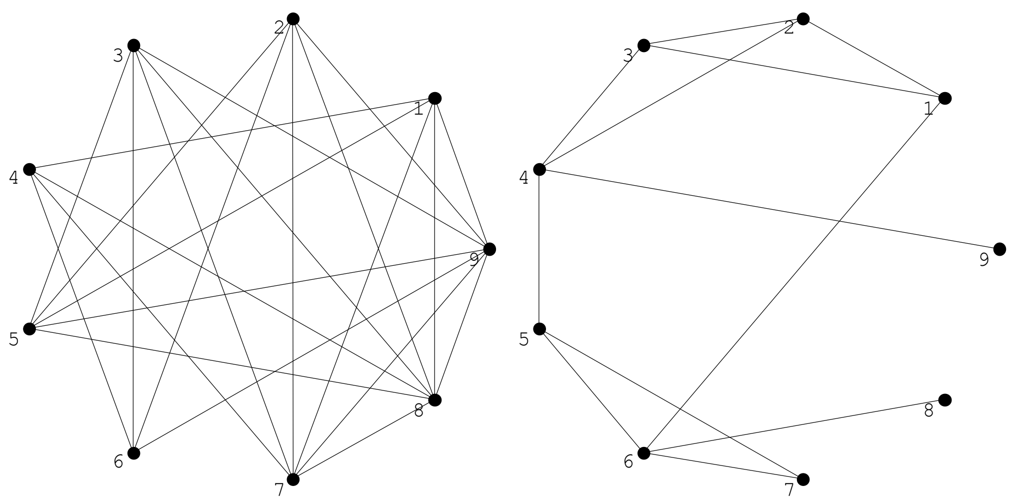
Graph 244081



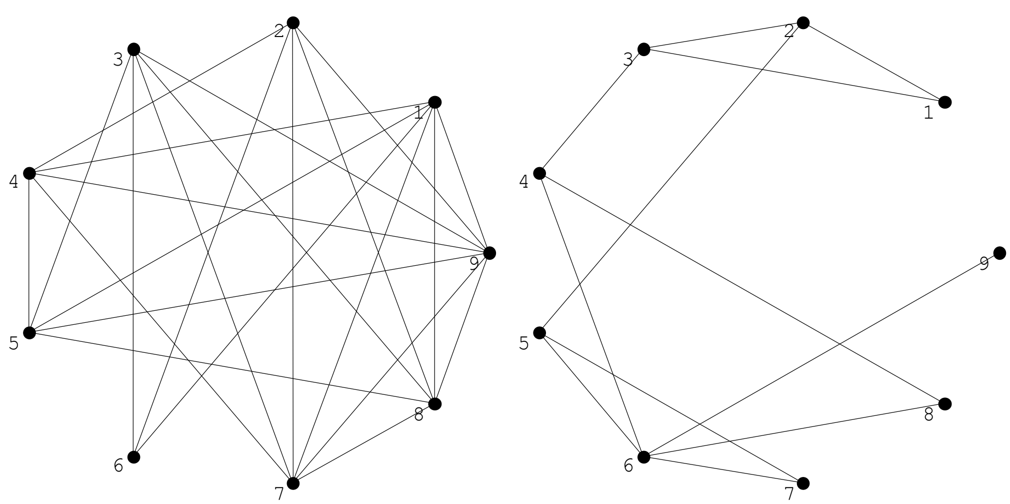
Graph 244645



Graph 244706



Graph 255930



References

KS – Kohara and Suzuki
MOR – Mattman, Ottman, Rodrigues