Graph Theorizing Peg Solitaire

D. Paul Hoilman
East Tennessee State University

December 7, 2010

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

The author would like to thank the ETSU Mathematics department and the Honors-In-Discipline program. Also, the author would definitely like to thank Dr. Beeler for his push, ideas, and dedication to this paper. Finally, the author thanks Dr. Haynes and Dr. Giroux for their time and suggestions.

Chapter 1

INTRODUCTION

We begin by introducing a variety of terms necessary to further investigate the problem. First and foremost, a **graph** is a mathematical relationship consisting of a nonempty finite set V of vertices and a finite set E of edges connecting these vertices. The **order** of graph G is the cardinality n of $V(G)$. The cardinality m of $E(G)$ is the **size** of G. A pair of vertices $u, v \epsilon V$ are said to be **adjacent** if edge $uv \epsilon E(G)$. Concordantly, edge uv is **incident with** u and v. The **degree** $(deg(v))$ of vertex v is the number of vertices adjacent to v.

A **path**, denoted P_n, is a graph in which there are two **end vertices** (vertices of degree 1), u and v, such that $deg(u) = deg(v) = 1$, and all other vertices have degree 2, connecting vertices u and v. An **even path** is a path of even order; an **odd path** is a path of odd order. A **cycle**, denoted C_n, is a closed path. In other words, all vertices have degree 2 since end vertices u and v now have degree 2 with the addition of edge uv. Similarly, an **even cycle** is a cycle of even order, while an **odd cycle** is a cycle of odd order. A **complete graph**, denoted K_n, is the graph in which all pairs of vertices are adjacent. For complete graphs, $|E(K_n)| = \frac{n(n-1)}{2}$. For a graph G, when we can partition V into two disjoint subsets A and B such that every edge contains exactly one vertex from each subset, G is said to be a **bipartite graph**. A **complete bipartite graph**, denoted $K_{a,b}$, graph is a bipartite graph in which every vertex of set A ($|A| = a$) is adjacent to every vertex of B ($|B| = b$). A **star**, denoted S_n, is a special complete bipartite graph of the form $K_{1,n}$.

The **cartesian product** of two graphs G and H, denoted $G \times H$, is the graph with vertex set $V(G) \times V(H)$ such that two vertices (u, v) and (u', v')

are adjacent in $G \times H$ if and only if $u = u'$ and v is adjacent to v' in H, or $v = v'$ and u is adjacent to u' in G. The **hypercube**, denoted Q_n, is a recursively defined graph involving the cartesian product with $Q_1 = K_2$ and $Q_n = K_2 \times Q_{n-1}$. In order to obtain the **join** of two disjoint graphs G and H, denoted $G + H$, append edges to each of the n vertices of G to each and every one of the m vertices of H, a total of mn edges. A **2-mesh**, denoted $M(m, n)$, is the grid formed by the cartesian product of two paths, $M(m, n) = P_m \times P_n$.

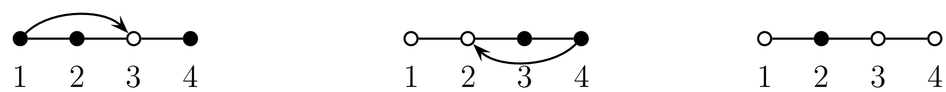
A **connected** graph is a graph in which any pair of vertices is linked by a path, resulting in a graph in which there is only one **component**, or piece. A **tree** is a connected acyclic graph. A **bridge** is an edge in a graph whose removal increases the number of connected components. A graph G has a **subgraph** H if the vertex set $V(H)$ and the edge set $E(H)$ are subsets of $V(G)$ and $E(G)$ respectively. A **spanning subgraph** H of G is a subgraph of G in which $V(H) = V(G)$ and $E(H) \subseteq E(G)$. A **spanning tree** is simply a spanning subgraph that is a tree. Two graphs G and H are said to be **isomorphic** if there exists a bijection from $V(G)$ to $V(H)$ such that any two vertices u and v are adjacent in G if and only if the corresponding vertices are adjacent in H. A graph with a spanning cycle is a **hamiltonian** graph. The **distance** between two vertices u and v, denoted $d(u, v)$, is defined as the smallest number of edges connecting vertices u and v.

We define a **starting state**, S, of graph G, such that $S \subset V(G)$ is a set of vertices which are empty. Similarly, a **terminal state**, T, of graph G, is a set of vertices with pegs at the end of the game, $T \subset V(G)$. To make this more clear, note that in the typical game, S contains one vertex, the initial hole, and T contains one vertex, the final peg.

In playing peg solitaire, a peg can jump an adjacent peg if there is an empty hole adjacent to the jumped peg. The jumped peg is then removed from the board. In the traditional game, jumps were only allowed to occur in a linear fashion. We, however, allow jumps to occur in any direction since the layout of any graph in graph theory is arbitrary. In order to completely solve a board, or graph, these jumps must continue until only one peg remains. See Figure 1.1 which shows the solving of P_4. Strategy must often be applied in graphs in which moves are not forced, and in determining the optimum initial hole location, or the optimum starting state.

A graph is said to be **k-solvable** if it can be solved down to k pegs. A 1-solvable graph is simply known as **solvable**. A graph is said to be **freely solvable** if it is solvable regardless of where the initial hole is placed. A

Figure 1.1: Solving P_4



move is defined as the series of consecutive jumps of a single peg. In other words, as long as you hold on to the same peg, every jump you make with this peg comprises a single move.

Turàn's Theorem states the following:

Theorem 1.0.1 *Let G be any subgraph of K_n such that G is K_{r+1}-free. Then the number of edges is at most $(\frac{r-1}{r})\frac{n^2}{2} = (1-\frac{1}{r})\frac{n^2}{2}$.*

We will need Mantel's Theorem, which is a special case of Turàn's Theorem, for one of our major threshold results. Mantel's Theorem states the following:

Theorem 1.0.2 *The maximum number of edges in an n-vertex, triangle-free graph is $\lfloor \frac{n^2}{4} \rfloor$.*

Also, we mention a necessary condition for a graph to be hamiltonian, outlined in Dirac's Theorem as follows:

Theorem 1.0.3 *If $deg_G(v) > \frac{n(G)}{2}$ for all vertices $v \epsilon G$, then G is hamiltonian.*

A key concept utilized throughout this paper is that of the *Reverse Game*, discussed in detail later. This concept is introduced by the following theorem.

Theorem 1.0.4 *Suppose that S is a starting state of graph G with associated terminal state T. Define the sets S' and T' by reversing the roles of "pegs" and "holes" in S and T, respectively. It follows that T' is a starting state of G with associated terminal state S'.*

Proof. Consider a game that is played in reverse. We begin the *Reverse Game* with a set of pegs in T. In the Reverse Game, if $xy, yz \epsilon E(G)$ with a peg in x and holes in y and z then x can jump over y into z, placing a new peg in y. Clearly, if S is a starting state in the original game with associated terminal state T, then the reverse game has terminal state S associated with T. The Reverse Game is equivalent to the original game. Reversing the roles of "pegs" and "holes" obtains the desired result. ■

Chapter 2

SIMPLE SOLVING CONCEPTS

In this paper, we will only view connected graphs, or graphs with only one component, because clearly any disconnected graph would not be solvable. We begin by considering the solving of very simple types of graphs, such as complete graphs, stars, paths, and cycles, and then examining very logically the solving of combinations of graphs under operations, such as the join and cartesian product. Consider the complete graph, K_n.

Proposition 2.0.5 *Complete graphs, K_n, are freely solvable for any value of n.*

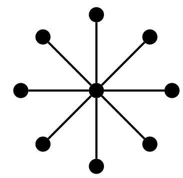
Proof. In K_n, all vertices are vertex transitive (all vertex labelings are equivalent), and all vertices are adjacent to all others. Hence, there is always a pair of adjacent vertices adjacent to a hole to allow jumping, until, there is only one peg remaining. Hence, K_n is solvable for any starting state, or freely solvable. ∎

Next, consider stars (S_n).

Theorem 2.0.6 *S_n is not solvable for any value of $n \geq 3$ and any starting state.*

Proof. For S_n, there are essentially two choices for the initial hole. You can either put the initial hole in the central vertex, or in one of the peripheral vertices (which are all vertex transitive). If the hole begins in the central

Figure 2.1: S_8



vertex, there are no adjacent pegs to jump with. If you put the hole in one of the peripheral vertices, you can jump from one of the other peripheral vertices across the central vertex into the hole. But now you are left with a hole in the central vertex, which leaves no adjacent pegs to jump with. Hence, S_n is not solvable. ■

We next consider paths and cycles. More specifically, we consider even paths, odd paths, even cycles, and odd cycles. In this investigation, we employ a very important concept which is a foundation to further investigations.

Theorem 2.0.7 *If a spanning subgraph H of G is k-solvable, then G is at worst k-solvable.*

Proof. Given graph H is k-solvable. Since H is a spanning subgraph of G, we know $V(H) = V(G)$ and $E(H) \subseteq E(G)$. Take G and solve it only on the edges $uv \epsilon E(H)$, which we know to be k-solvable since H is k-solvable on the edges $uv \epsilon E(H)$. The additional edges of G may or may not allow further solving. Hence, we are satisfied to say G is at worst $k-$solvable. ■

This theorem allows for multiple proofs of other results, such as Corollary 2.0.10.

Theorem 2.0.8 *P_{2n} is solvable with the initial hole in slot 2 or n for $n = 2$ and $n > 3$. For initial hole in slot $2n - 1$, the final peg ends in slot 2.*

Proof. Label vertices $1, 2, ..., 2n$ in the obvious way. Proceed by induction on n. For $n = 2$, place the initial hole in slot $2n - 1 = 3$. Jump from 1 to 3. Holes are now in slots 1 and 2. Now jump from 4 to 2 and we are done. See Figure 2.2.

Assume that P_{2n} is solvable with starting position $2n - 1$ for some n. Look at $P_{2(n+1)}$ with the starting hole in slot $2n + 1$. See Figure 2.3.

Jump from $2n - 1$ into $2n + 1$. Now we have holes in $2n$ and $2n - 1$. Jump from $2n + 2$ into $2n$, and we are left with holes in slots $2n + 2$, $2n + 1$,

Figure 2.2: Solving P_4

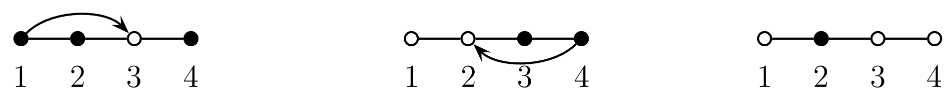
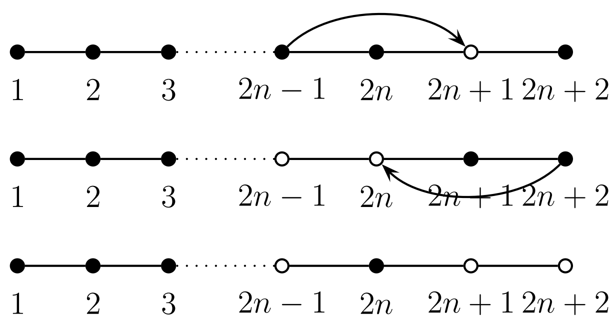


Figure 2.3: Solving P_{2n+2}



and $2n-1$. Ignoring slots $2n+2$ and $2n+1$, which are empty, we are left with P_{2n} with a hole in slot $2n-1$. This, though, is solvable by the inductive hypothesis. Thus, the result holds. ■

Notice that n must be greater than 1, or else we would consider the trivial case, P_2. For $n = 3$, P_6 is solvable only when the initial hole is in slot 2. For $n \geq 3$, when placing the initial hole in slot 2, the final peg ends up in slot $2n-4$ or $2n-1$. As for odd paths, trivially we know P_1 is solvable. Also, P_3 is solvable with initial hole at one of the two end vertices. We now define **distance 2-solvable** as a graph which is 2-solvable where $d(u, v) = 2$ where u and v are the final two pegs.

Theorem 2.0.9 *P_{2n+1} is distance 2-solvable with initial hole at slot $2n$, $n \geq 2$. Further, the final two pegs fall into holes 1 and 3.*

Proof. Proceed by induction on n. For $n = 2$, the initial hole is placed in position 4. First jump from slot 2 to 4, then from 5 to 3. Pegs are left in slots 1 and 3 with the remaining vertices left as holes. Clearly, $d(v_1, v_3) = 2$. See Figure 2.4.

Assume the result holds for some n. Now consider $P_{2(n+1)+1} = P_{2n+3}$. See Figure 2.5.

Figure 2.4: Solving P_5

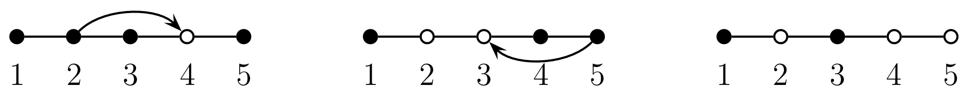
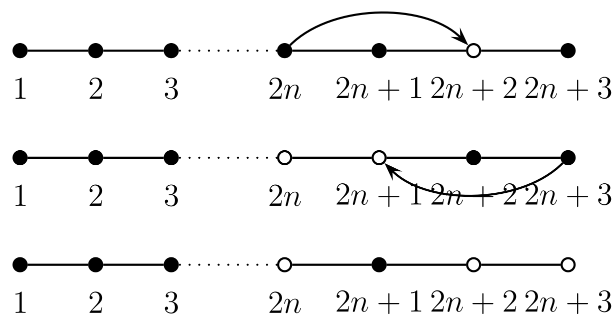


Figure 2.5: Solving P_{2n+3}



Jump from $2n$ to $2n+2$ (the initial hole). Then, jump from $2n+3$ to $2n+1$. The remaining graph is P_{2n+1} with a hole in slot $2n$. This is distance 2-solvable by the inductive hypothesis. Thus, the result holds. ∎

Corollary 2.0.10 *C_{2n} is freely solvable. C_{2n+1} is distance 2-solvable with $d(u,v)=2$, where u and v are the two remaining vertices with pegs.*

Proof. Take C_{2n} and delete one edge, resulting in P_{2n} which is solvable with the initial hole in slot $2n-1$ by the Theorem 2.4. Placing the edge back in, the labeling becomes irrelevant, as the vertices are vertex transitive. Hence, C_{2n} is freely solvable.

Similarly, delete one edge from C_{2n+1}, resulting in P_{2n+1}, which again, is distance 2-solvable by the above theorem, with the initial hole placed in slot $2n$. However, re-inserting the deleted edge makes the graph again vertex transitive, removing the importance of the labeling. Hence, C_{2n+1} is distance 2-solvable.

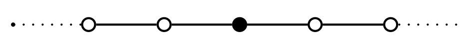
OR

Note that P_{2n} is a spanning subgraph of C_{2n}. Since P_{2n} is solvable for certain initial hole conditions, by Theorem 2.0.7, we know C_{2n} is at least solvable for these initial hole conditions. However, since C_{2n} is cyclic and vertices are vertex transitive, the labeling of the initial hole conditions becomes irrelevant. Therefore, C_{2n} is freely solvable.

Similarly, note that P_{2n+1} is a spanning subgraph of C_{2n+1}. Since P_{2n+1} is distance 2-solvable for certain initial hole conditions, by Theorem 2.0.7, we know C_{2n+1} is at worst distance 2-solvable for these initial hole conditions. However, since C_{2n+1} is cyclic and vertices are vertex transitive, the labeling of the initial hole conditions becomes irrelevant. Therefore, C_{2n+1} is distance 2-solvable. ∎

The question remains, however, of whether or not an odd path/cycle is solvable. We have proved they are both 2-solvable, but would a more clever strategy leave the graphs with only one peg? We look to diffuse this question now. We define an **empty bridge** as a bridge in which the two edge vertices have degree two and are holes. Observe that a single peg adjacent to nothing but empty bridges is stranded (Figure 2.6).

Figure 2.6: Stranded Peg



Theorem 2.0.11 *The path P_{2n+1}, $n \geq 2$, is not solvable.*

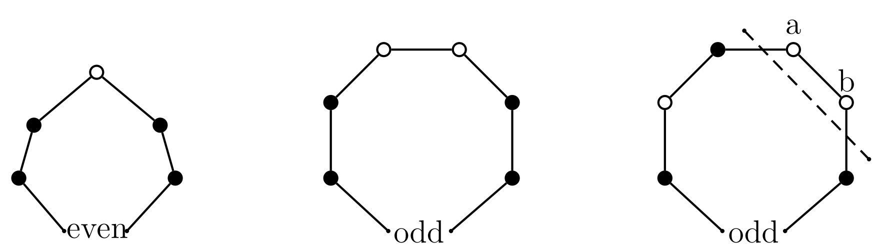
Proof. Note for the odd path in its starting state, there are an even number of pegs and a single hole. No matter where we choose to place the initial hole, we divide the $2n$ pegs into two groups. These groups must necessarily be both even or both odd. In both cases, the first move jumps with two pegs from one group, removing them and adding one peg to the other group. In the case in which the groups are both even, the group with the jumping pegs remains even, then there is an empty bridge, then an odd group of pegs. If both groups are odd, the first move keeps the jumping group odd while changing the other group to having an even number of pegs, connected by an empty bridge. In both cases, after the first jump, there is an empty bridge in between an odd number of pegs and an even number of pegs.

Observe that any jump requires two pegs. Also, on the path, note that each jump results in the formation of a new empty bridge. In order to solve the odd path, we attempt to link the set of even pegs with the set of odd pegs by jumping back across the bridge from both sides. Since we know one set of pegs is odd, let the number of pegs on this side of the empty bridge formed by the first move be represented by $2v + 1$ for some $v \epsilon \{0, 1, 2, ...\}$. After v moves, we will necessarily be left with one peg adjacent to only an empty bridge, which, by observation, represents a stranded peg. Hence, P_{2n+1} is not solvable. ■

Theorem 2.0.12 C_{2n+1} *is not solvable.*

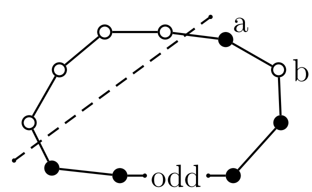
Proof. Similar to the odd path, the odd cycle, in its starting state, consists of an even number of pegs and a single hole. After the forced first move, there is an empty bridge (note here that the empty "bridge" deletion does not disconnect the graph) surrounded by an odd number of pegs. After the next move, which is forced, we are left with an empty bridge (call it ab) adjacent on one side to a single peg, a hole, and an odd number of pegs which are adjacent with the other side of the empty bridge. If we ignore empty bridge ab, we are left with an odd path, which we know is not solvable by Theorem 2.0.11 (See Figure 2.7).

Figure 2.7: Attempting to Solve C_{2n+1}



Assume we use hole a. We attack from the left by making two jumps ending in hole a. We are left with a series of four empty holes (two empty bridges) adjacent to, essentially, an odd path. If we ignore the four empty holes, we are left with an odd path, which we know is not solvable by Theorem 2.0.11. If we jump into one of the four empty holes, we will necessarily leave a single peg surrounded on both sides by empty bridges (since, like paths, each jump creates a new empty bridge), which is stranded (See Figure 2.8).

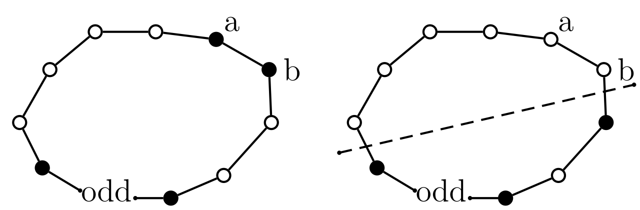
Figure 2.8: Attempting to Solve C_{2n+1}



Instead, assume we use hole b. By jumping into hole b, it is adjacent on the right by an empty bridge, hence it is "stranded" from this side. So, by deciding to use hole b, we must also use hole a, or else b is stranded. Hence, we attempt to jump into a from the left by a series of two moves as above. We now have pegs in a and b with holes adjacent to both a and b. If we jump from b over a, the resulting peg would be surrounded on both sides by empty bridges, and thus, stranded. Instead, we jump from a over b. This results in a group of six consecutive holes adjacent to an odd path. If we choose not to use the six holes, we are left with an odd path, which again, is not solvable by Theorem 2.0.11. If we jump into one of the six empty holes, we necessarily leave a single peg surrounded on both sides by empty bridges, which is stranded. See Figure 2.9. Hence, C_{2n+1} is not solvable.

■

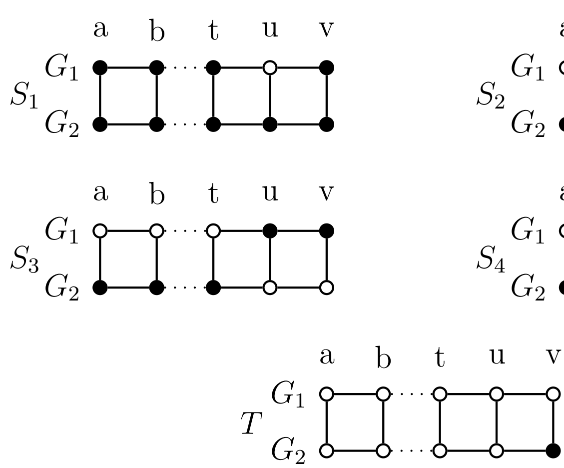
Figure 2.9: Attempting to Solve C_{2n+1}



Theorem 2.0.13 *Given freely solvable (solvable) graph G, the cartesian product $G \times K_2$ is freely solvable (solvable).*

Proof. Note that $G \times K_2$ is essentially two copies of G, call them G_1 and G_2, with corresponding vertices connected. Let (G_1, a) represent vertex a of G_1. Call the vertex in G in which the final peg ends v for initial hole u. See Figure 2.10.

Figure 2.10: Solving $G \times K_2$

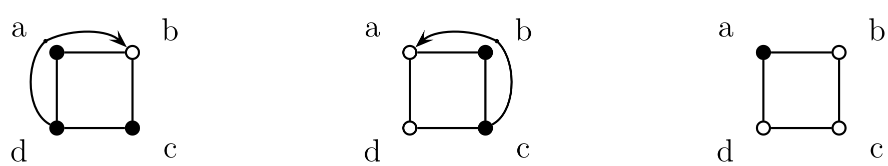


Let the initial hole be in (G_1, u), shown by S_1. Solve G_1, leaving the final peg in (G_1, v), shown by S_2. Jump from (G_2, v) over (G_2, u) into (G_1, u), shown by S_3. Next, jump from (G_1, u) over (G_1, v) into (G_2, v). This leaves G_2 with a hole in u, shown by S_4. Solve G_2, solving the graph, shown by T. ■

Corollary 2.0.14 *Q_n, $n \geq 2$, is freely solvable.*

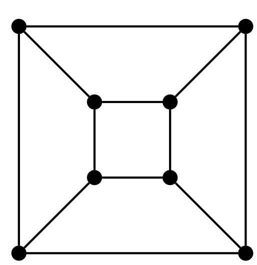
Proof. Note Q_n is given by $Q_{n-1} \times K_2$. We proceed by induction on n. Trivially, Q_0 and Q_1 are freely solvable. For $n = 2$, $Q_2 = Q_1 \times K_2$. Pick any vertex as the initial hole since vertices of Q_n are vertex transitive. Pick vertex b without loss of generality. See Figure 2.11. After two moves, Q_2 is solved

Figure 2.11: Solving Q_2



down to one peg. Therefore, the result holds for $n = 2$ (equivalently, note $Q_2 = C_4$, which is solvable by Corollary 2.0.10). Assume the result holds for some n. Hence, $Q_n = Q_{n-1} \times K_2$ is freely solvable. We look to see if the result holds for $n+1$. By definition, $Q_{n+1} = Q_n \times K_2$ by definition. And, Q_n is freely solvable by the inductive hypothesis. So, by Theorem 2.0.13 above, Q_{n+1} is also freely solvable. Hence, the result holds for $n \geq 2$. ■

Figure 2.12: Q_3



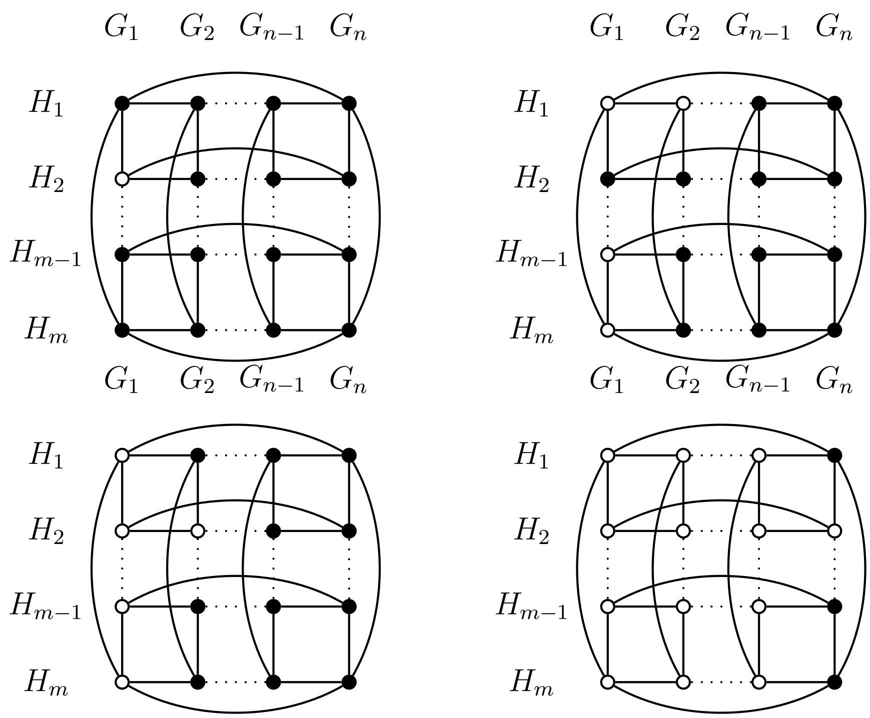
Theorem 2.0.15 *Given freely solvable graphs G and H, $G + H$ is freely solvable.*

Proof. Consider $G+H$ with one hole. Since any hole (vertex) can be viewed as being part of G or H, without loss of generality view the hole as part of G. Since G is freely solvable, proceed to solve G. Once G is solved, take a peg from H, jump over the last peg in G and into an adjacent hole in G. Now, we have one peg in G and one hole in H. Take the peg in G, jump over a peg in H which is adjacent to the lone hole in H, and into the lone hole in H. Now, you are left with graph H with one hole. Since H is freely solvable, you can solve the remaining portion of $G + H$. Hence, the result holds. ∎

Theorem 2.0.16 *If G and H are even cycles, $G \times H$ is freely solvable.*

Proof. Let G have order m and H have order n such that m and n are of the form $2 + 2k$ for some $k = \{0, 1, 2, ...\}$. Consider $G \times H$ with one hole. Since G and H are freely solvable, without loss of generality view the hole as if it belongs to G. Specifically, view the hole as it is in the first copy of G, call this G_1. Solve G_1. Take a peg from G_2 which corresponds to the final peg in G_1, jump the last peg in G_1, and land in an adjacent hole in G_1. Now, jump from G_1 into G_2 by landing in the lone hole of G_2. We are now left with $n-2$ copies of G completely filled with pegs, one copy of G completely empty (G_1), and one copy of G with one hole (G_2). Repeat this process $n-2$ times followed by a solving of the final copy of G, G_n. $G \times H$ is now solved down to one peg. Hence, the result holds. See Figure 2.13. ∎

Figure 2.13: Solving $G \times H$



Chapter 3

IMPROVED SOLVING

We now employ further knowledge of graph theory in analyzing the solvability of graphs as opposed to a more brute force, purely logical analysis as above. Using the above results, we present these new results which give more truths concerning the solvability of generic graphs given certain requirements.

Theorem 3.0.17 *i) If G is hamiltonian and has even order, it is freely solvable.*

ii) If G is hamiltonian and contains a triangle, it is solvable.

Proof. i) Since G is hamiltonian and of even order it has C_{2n} as a spanning subgraph. This is freely solvable by Theorem 2.0.7.

ii) We are done if G has even order by above. Without loss of generality, assume G is of odd order. Since G is hamiltonian, it has C_{2n+1} as a spanning subgraph. Solve on the subgraph in such a way that the two final pegs are on two vertices of a triangle. Then, jump along the triangle, solving the graph. ■

Corollary 3.0.18 *If $deg_G(v) > \frac{n(G)}{2}$ for all $v \epsilon V(G)$, then G is solvable.*

Proof. Given $deg_G(v) > \frac{n(G)}{2}$, which implies G is hamiltonian by Theorem 1.0.3. So, if G has even order, we are done by above. If not, the number of edges in G is

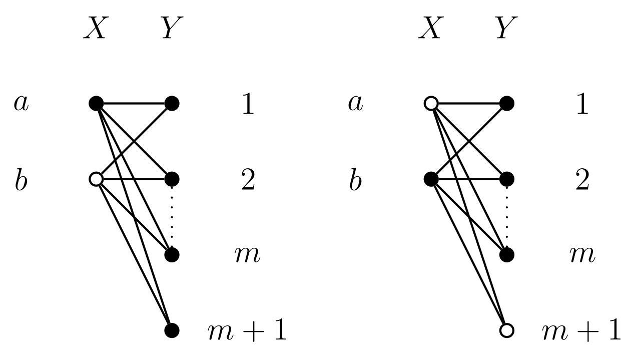
$$e(G) \geq \frac{n(G)\delta(G)}{2} > \frac{n(G)^2}{4}.$$

Therefore, G has a triangle by Mantel's Theorem. Thus, G is hamiltonian and contains a triangle, which is solvable by Theorem 3.0.17. ■

Theorem 3.0.19 *$K_{2,m}$ is freely solvable for $m \geq 2$ (solvable for $m = 1$).*

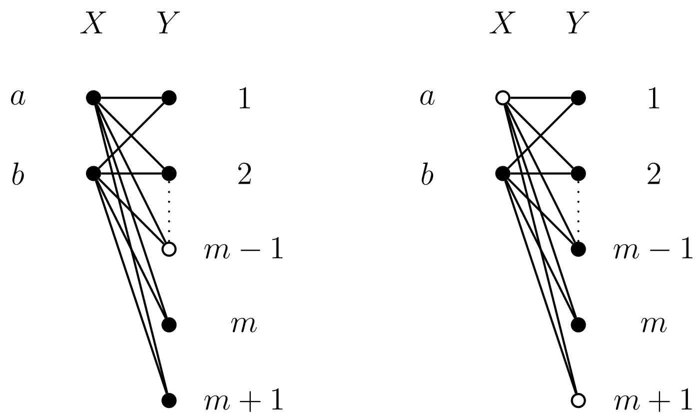
Proof. Let vertex set by $X \cup Y$ where $X = \{a, b\}$, $Y = \{1, 2, ..., m\}$. If initial hole is placed in X, proceed by induction on m. If $m = 1$, then $K_{2,m} = P_3$, which is solvable. If $m = 2$, then $K_{2,m} = C_4$, which is freely solvable. Suppose this is true for some m. Consider $K_{2,m+1}$. Jump from a over $m+1$ to b. Ignore hole at slot $m+1$. The remaining graph is $K_{2,m}$ with a hole in X, which is solvable by the inductive hypothesis. See Figure 3.1.

Figure 3.1: Solving $K_{2,n}$, initial hole in X



Now suppose the initial hole is in Y. Without loss of generality, let the initial hole be in slot $m-1$. Jump from $m+1$ over a into $m-1$. Ignore hole at $m+1$. Remaining graph is $K_{2,m}$ with a hole at a, which is solvable. See Figure 3.2.

Figure 3.2: Solving $K_{2,n}$, initial hole in Y



Hence, the result holds. ∎

Theorem 3.0.20 *$K_{n,m}$ is freely solvable ($n \geq 2$, $m \geq 2$).*

Proof. Proceed by induction on n. If $n = 2$, $K_{2,m}$ is freely solvable for all m by above. Suppose $K_{n,m}$ is freely solvable for some $n \geq 2$.

$$V = X \cup Y$$

$$|X| = n, |Y| = m$$

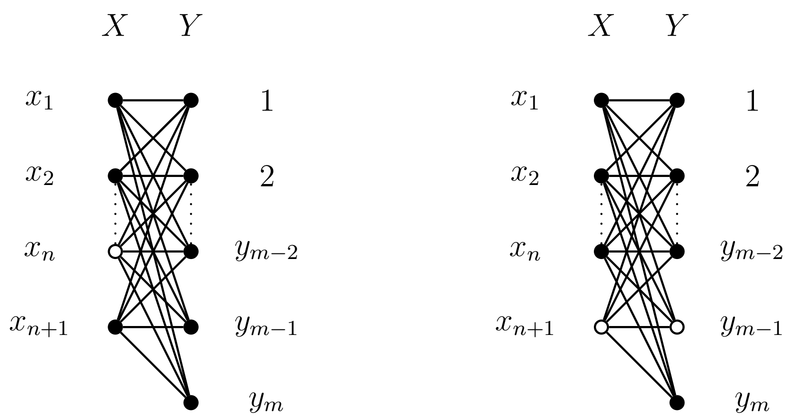
Consider $K_{n+1,m}$

$$X = \{x_1, ..., x_n, x_{n+1}\}$$

$$Y = \{y_1, ..., y_m\}$$

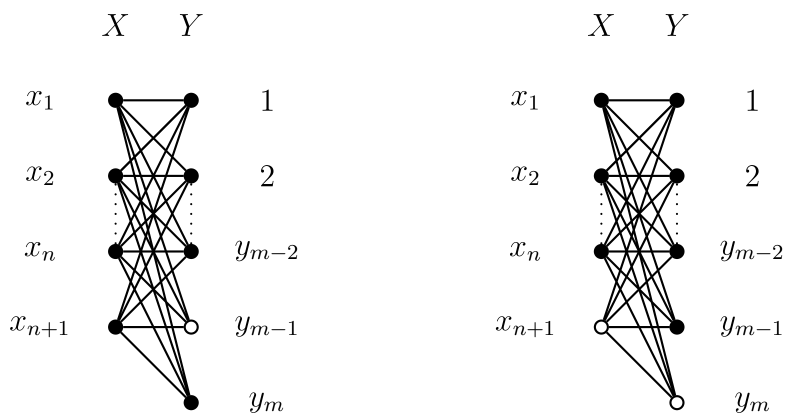
If initial hole is in X (say x_n), jump from x_{n+1} over y_{m-1} to x_n. Ignoring the hole in x_{n+1}, remaining graph is $K_{n,m}$, which is solvable by the inductive hypothesis. See Figure 3.3.

Figure 3.3: Solving $K_{n,m}$, initial hole in X



If initial hole is in Y (say y_{m-1}), jump from y_m over x_{n+1} to y_{m-1}. Ignoring the hole in x_{n+1}, the remaining graph is $K_{n,m}$ with a hole in Y, which is solvable by the inductive hypothesis. See Figure 3.4. Thus, the result holds. ■

Figure 3.4: Solving $K_{n,m}$, initial hole in Y



Corollary 3.0.21 *Let G, H be graphs with more than two vertices each. $G + H$ is freely solvable.*

Proof. $G+H$ has $K_{n(G),n(H)}$ as a spanning subgraph, which is freely solvable by above. ∎

Previously, we proved that the cartesian product of any two even cycles was freely solvable. We now prove that any 2-mesh, a spanning subgraph of the cartesian product of two even cycles, is solvable.

Theorem 3.0.22 *$M(m, n)$ is solvable.*

Proof. We must consider three cases for this proof. First, the case in which m and n are both even. Second, the case in which, without loss of generality, m is odd and n is even. Lastly, the case in which both m and n are odd. Let (x, y) represent the vertex generated by $x \epsilon V(P_m)$ and $y \epsilon V(P_n)$.

Case 1: Suppose m and n are both even. Begin with the initial hole in $(1, 3)$. Jump from $(2, 2)$ over $(1, 2)$ into $(1, 3)$. We now have holes in vertices $(1, 2)$ and $(2, 2)$. We know we can solve each of these paths independently, resulting in a single peg in each path at location $n - 1$ by Theorem 2.0.8. These two pegs are clearly adjacent, and we can jump them in any way if these are the last pair of two paths. However, if there is another pair of two paths, we must continue and reproduce this solving possibility. Jump from $(1, n-1)$ over $(2, n-1)$ into $(2, n-2)$. Then, jump from $(4, n-1)$ over $(3, n-1)$ into $(2, n-1)$. Next jump from $(2, n-2)$ over $(3, n-2)$ into $(3, n-1)$. Finally, jump from $(2, n-1)$ over $(3, n-1)$ into $(3, n-2)$. This leaves vertical paths three and four with holes in slots $n-1$, or equivalently, with different labeling, the second slots of each path. Again, we know we can solve these paths independently, leaving pegs adjacent in hole 2 (or $n-1$ depending on your labeling). If this was the last pair of paths, we have two adjacent pegs, which we can finish solving. If not, repeat the process for each pair of paths, eventually solving the 2-mesh. See Figure 3.5.

Case 2: Without loss of generality, suppose m odd and n even. Begin as in the same as Case 1. We can solve each pair of paths in the exact same way. However, when we no longer have a pair of paths to solve, and only a single path remaining, our plan for solving must deviate. In vertical path $m-1$, we end up with a peg in slot $n-2$ (or 3). Jump from $(m, n-2)$ over $(m-1, n-2)$ into $(m-1, n-1)$. Then jump from $(m-1, n-1)$ over $(m, n-1)$ into $(m, n-2)$. This leaves the last, even path with a hole in slot $n-1$ (or 2) which we know is solvable. Hence, the 2-mesh is solved. See Figure 3.6.

Case 3: Suppose m and n are both odd. Begin as in Case 1, with the initial hole $(1, 3)$. Jump from $(2, 2)$ over $(1, 2)$ into $(1, 3)$. We now have holes in vertices $(1, 2)$ and $(2, 2)$, which we know each path is independently distance 2-solvable with final pegs landing in locations n and $n-2$ by Theorem 2.0.9. Solve each path independently. Then solve the set of four pegs in such a way that the final peg of the four lands in $(2, n-1)$. Next jump from $(3, n-1)$ over $(2, n-1)$ into $(2, n-2)$. Finally, jump from $(2, n-2)$ over $(3, n-2)$ into $(3, n-1)$. This leaves the third vertical path with a hole in location $n-2$ (or 3, depending on labeling). We can repeat this process for each pair of vertical paths until there are only 5 paths remaining.

When there is a hole in $(m-4, 3)$, as similar to above, jump from $(m-3, 2)$ over $(m-4, 2)$ into $(m-4, 3)$. Again, 2-solve these two paths independently. Solve this set of four pegs and two holes such that the final peg of the four lands in vertex $(m-4, n-1)$. Then, jump $(m-1, n-1)$ over $(m-2, n-1)$ into $(m-3, n-1)$. Now jump from $(m-4, n-1)$ over $(m-3, n-1)$ into $(m-3, n)$. We next make a series of four jumps. First, jump from (m, n) over $(m, n-1)$ into $(m-1, n-1)$. Next, jump from $(m-1, n)$ over $(m-1, n-1)$ into $(m-2, n-1)$. Third, jump from $(m-2, n)$ over $(m-2, n-1)$ into

Figure 3.5: Solving the 2-mesh, m & n even

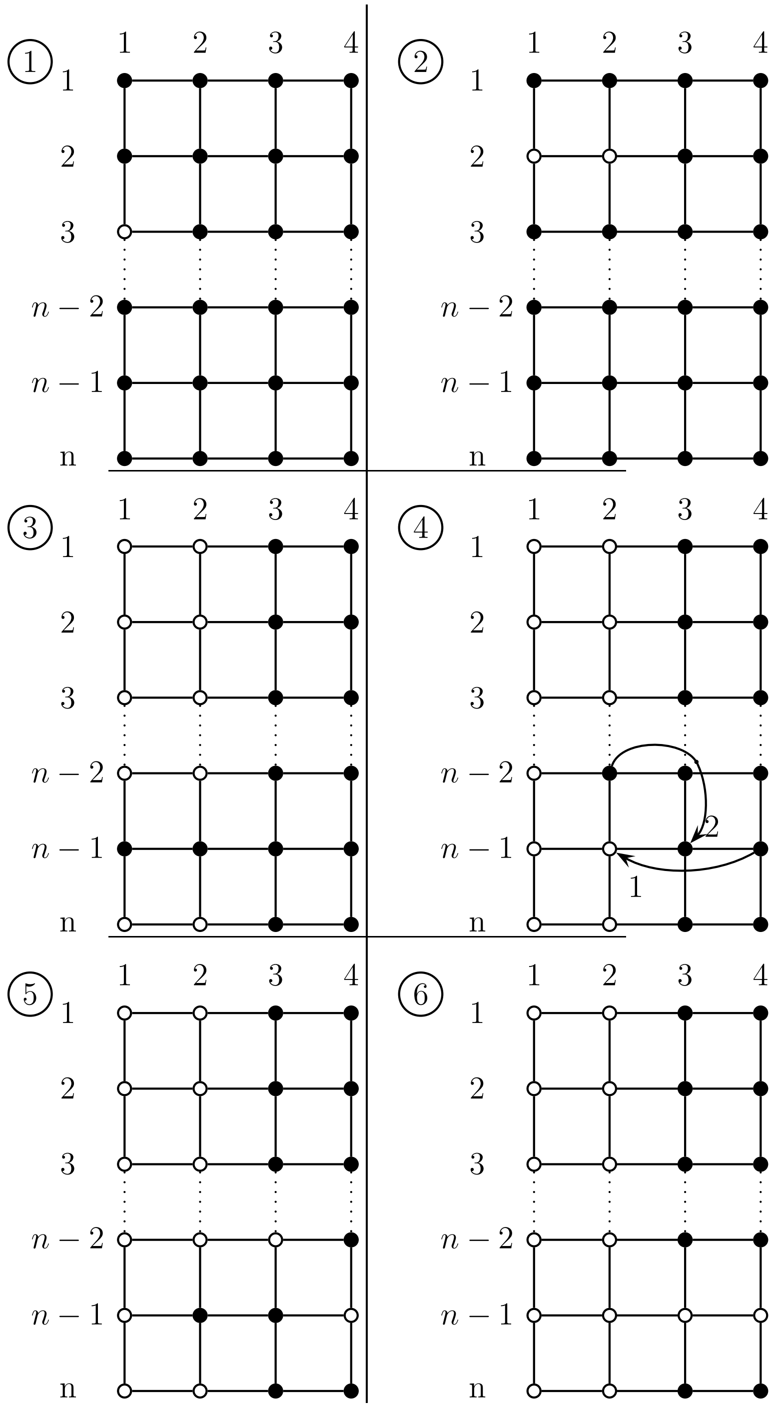
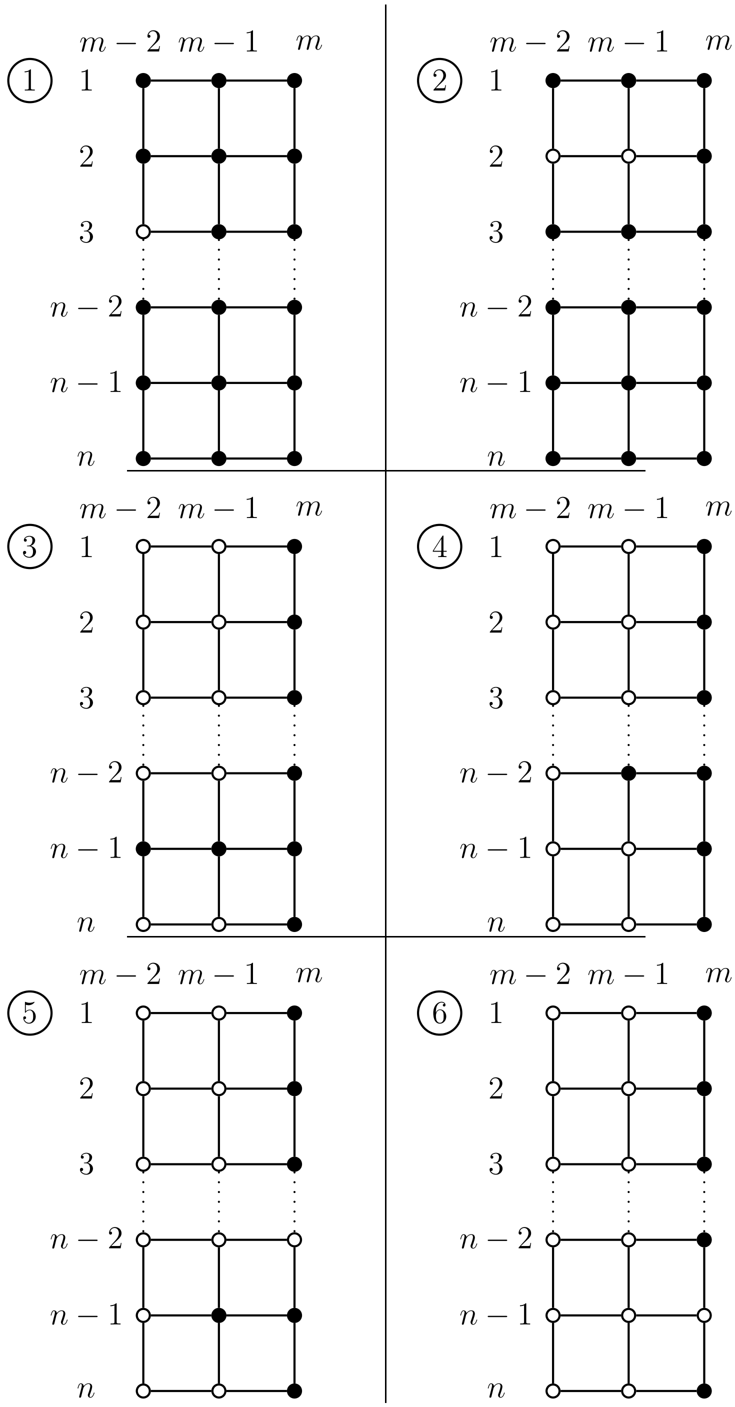


Figure 3.6: Solving the 2-mesh, m odd, n even



$(m-3, n-1)$. Finally, jump from $(m-3, n-1)$ over $(m-3, n)$ into $(m-2, n)$. This leaves vertical path $m-2$ with a hole in vertex $n-1$ and vertical paths $m-1$ and m with holes in vertices $n-1$ and n. Path $m-2$, we know, is 2-solvable with the final two pegs ending in slots 1 and 3. Ignoring the bottom holes, which are empty, of paths $m-1$ and m, these paths are of even order with holes in location "n". All even paths with this starting state will solve down to pegs in slots 1, 4, 6, 8, 10, ... n. If the "even" paths (we created by ignoring the hole in actual slot n) are of order $4k$, for some $k \epsilon \mathbb{N}$, jump left, from $(m, n-1)$ over $(m-1, n-1)$ into $(m-1, n-2)$. Then jump from $(m-1, n-3)$ over $(m-1, n-2)$ into $(m, n-2)$. Next, jump from $(m, n-2)$ over $(m, n-3)$ into $(m, n-4)$. Now, jump from $(m, n-5)$ over $(m, n-4)$ into $(m-1, n-4)$. Then jump from $(m-1, n-4)$ over $(m-1, n-5)$ into $(m-1, n-6)$. Continue this back-and-forth, up-and-down pattern of jumping until eventually there is a peg in vertex $(m-1, 3)$. If the "even" paths are of order $4k+2$, for some $k \epsilon \mathbb{N}$, jump right first, from $(m-1, n-1)$ over $(m, n-1)$ into $(m, n-2)$. Continue in the same pattern until there is a peg in vertex $(m-1, 3)$. Next, jump from $(m-1, 3)$ over $(m-2, 3)$ into $(m-2, 2)$, leaving just four pegs. We can proceed to make three jumps in a counterclockwise fashion to finish solving. Namely, jump first from $(m-2, 1)$ over $(m-2, 2)$ into $(m-1, 2)$. Then, jump from $(m-1, 1)$ over $(m-1, 2)$ into $(m, 2)$. Finally, jump from $(m, 2)$ over $(m, 1)$ into $(m-1, 1)$. See Figure 3.7 and Figure 3.8.

Figure 3.7: Solving the 2-mesh, m & n odd

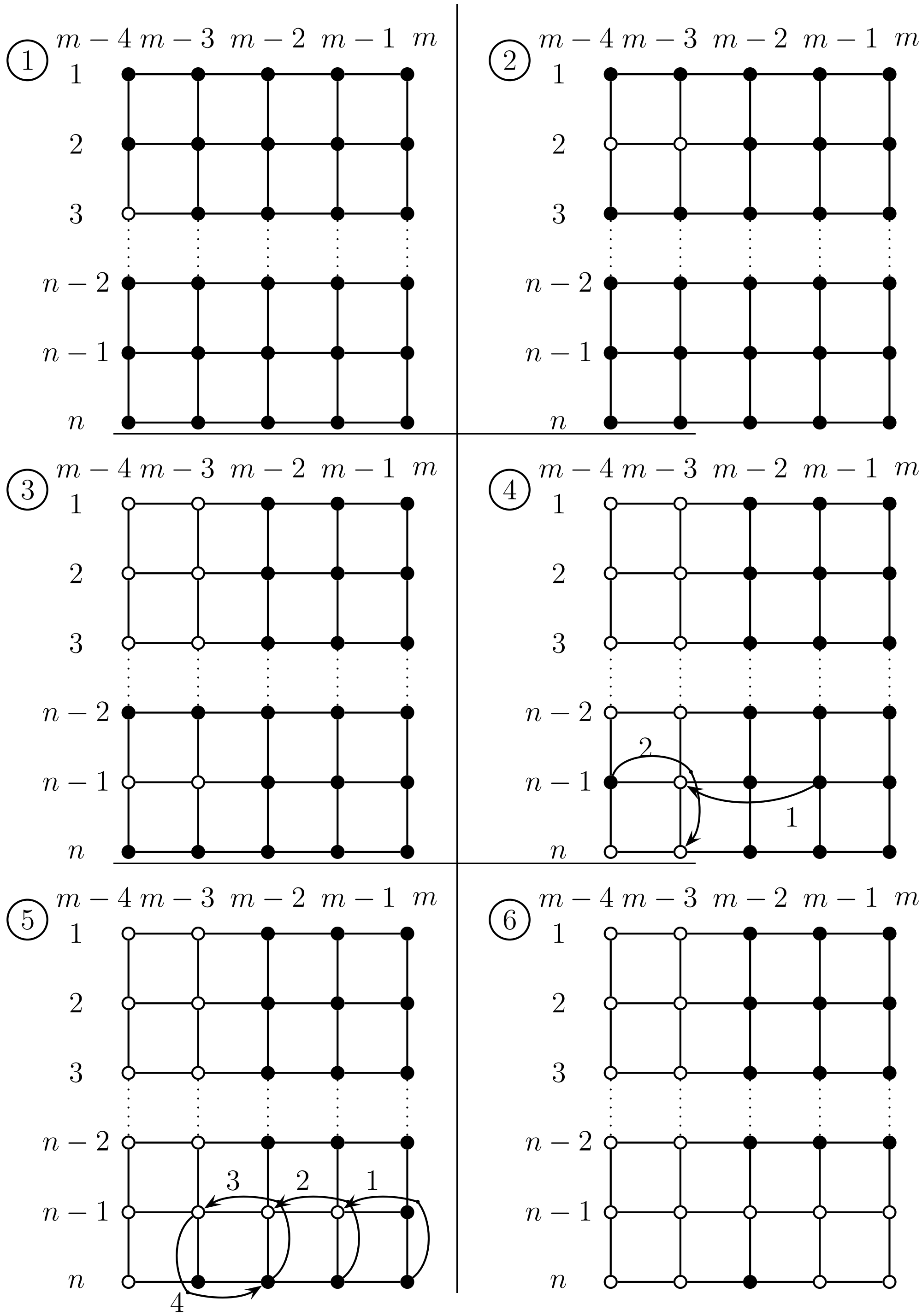
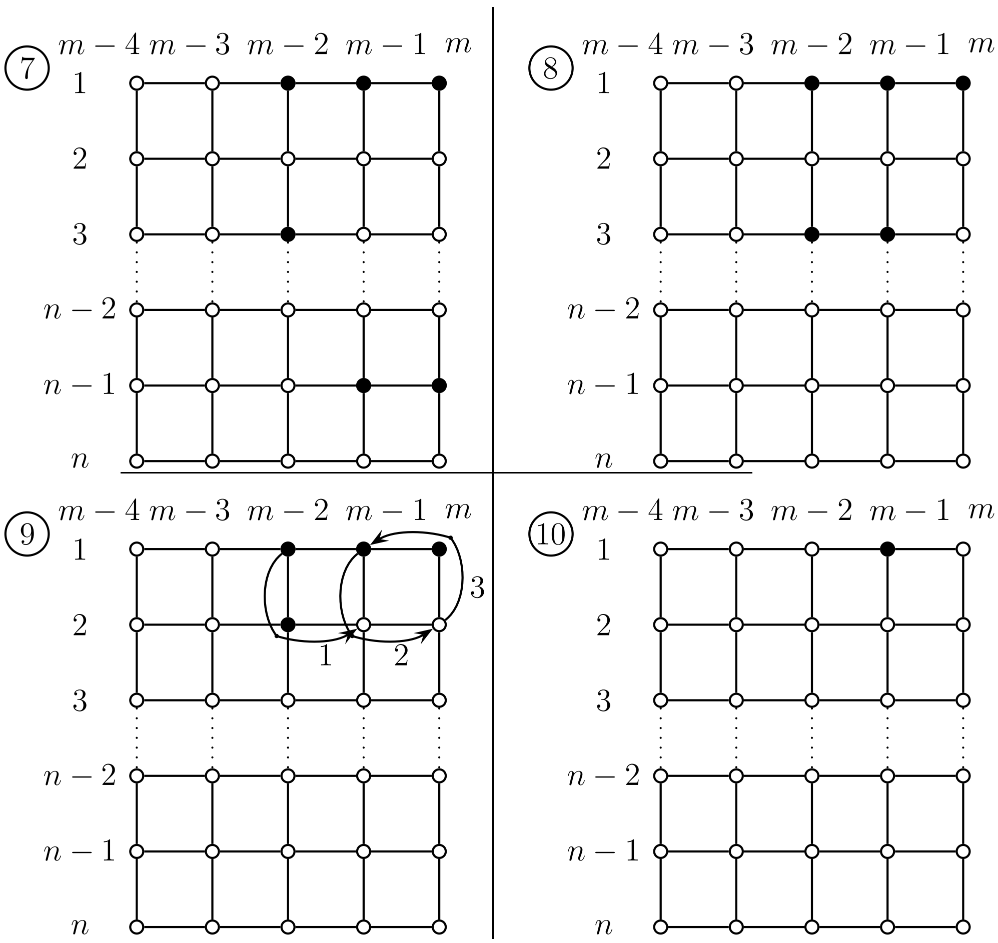


Figure 3.8: Solving the 2-mesh (continued), m & n odd



Hence, any 2-mesh is solvable. ■

Theorem 3.0.23 *Given solvable graphs G and H, the cartesian product $G \times H$ is solvable.*

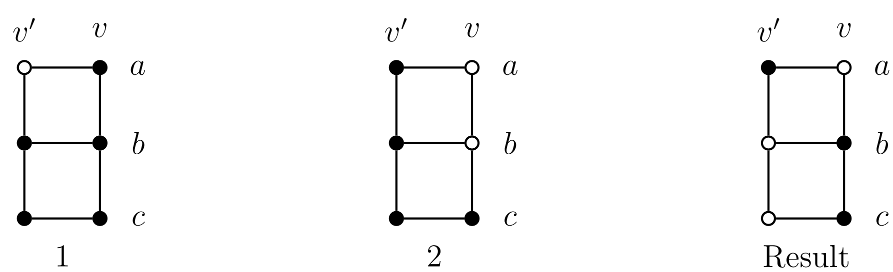
Proof. If $H = K_1$, $G \times H = G$ which is solvable by hypothesis. If $H = K_2$, we know $G \times H$ is solvable by Theorem 2.0.13. Without loss of generality, assume $n(G) \geq 3$, $n(H) \geq 3$. We know there exist some vertices a and z in which the G is solvable with a as its initial hole and z as the final peg location. We are also guaranteed vertices b, c, x, and y such that $ab, bc, xy, yz \epsilon E(G)$. The first move we have constructed is jumping c over b into a; the last move is made by jumping x over y into z. Similarly, we know H is solvable starting at u ending at location v. In H, u is adjacent to u' and v is adjacent to v'. Let G_h be the copy of G generated by $h \epsilon V(H)$ and H_g be the copy of H generated by $g \epsilon V(G)$. Also, let (g, h) be the vertex in $G \times H$ generated by $g \epsilon V(G)$ and $h \epsilon V(H)$.

Start with initial hole in (a, u). Since H is solvable starting at u, solve H_a ending at (a, v). Note all copies of G have a hole in a except G_v.

We now make the following series of moves (see Figure 3.9):

1. Jump from (b, v) over (a, v) into (a, v');
2. Jump from (c, v') over (b, v') into (b, v).

Figure 3.9: Solving $G \times H$



Result: $G_{v'}$ has holes in (b, v') and (c, v') with pegs elsewhere. This is the state of G after our stated initial jump of c over b into a. Hence, $G_{v'}$ is solvable with final peg landing in (z, v'). G_v is also solvable, since it has a single hole in vertex (a, v) with pegs elsewhere.

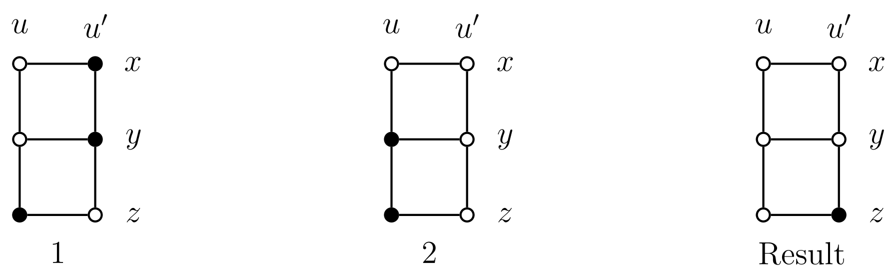
Now, all copies of G have holes in a (except $G_{v'}$). Solve them independently, except $G_{u'}$, ending in z each time. Also solve $G_{v'}$ ending in (z, v') by above.

As for $G_{u'}$, solve it in such a way that pegs end in locations (x, u') and (y, u').

We now make the following series of moves (see Figure 3.10):

1. Jump from (x, u') over (y, u') into (y, u);
2. Jump from (y, u) over (z, u) into (z, u').

Figure 3.10: Solving $G \times H$



Result: H_z has a hole in (z, u), pegs elsewhere. All other copies of H are empty.

Since H_z is solvable from (z, u), we solve it (ending in (z, v)), completing the proof.

■

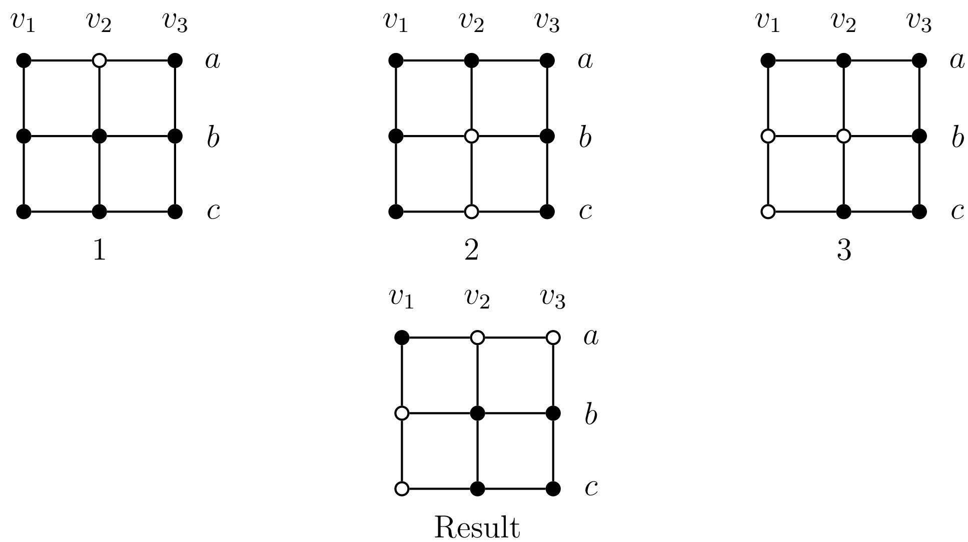
Theorem 3.0.24 *Suppose G is solvable (as described in Theorem 3.0.23) and H is distance 2-solvable starting in u ending in v_1, v_3. $G \times H$ is solvable.*

Proof. Start with initial hole in (a, u). Let u_1 and u_2 be vertices of H in the initial move. Since H_a is 2-solvable starting at u ending at v_1, v_3 (v_2 is adjacent to v_1 and v_3 in H), distance 2-solve H_a. Now all copies of G have a hole at a except G_{v_1} and G_{v_3}.

We now make the following series of moves (see Figure 3.11):

1. Jump from (c, v_2) over (b, v_2) into (a, v_2);
2. Jump from (b, v_1) over (c, v_1) into (c, v_2);
3. Jump from (a, v_3) over (a, v_2) into (b, v_2).

Figure 3.11: Solving $G \times H$ (H distance 2-solvable)



Result: Pegs in (a, v_1), (b, v_2), (b, v_3), (c, v_2), (c, v_3); holes in (a, v_2), (a, v_3), (b, v_1), (c, v_1).

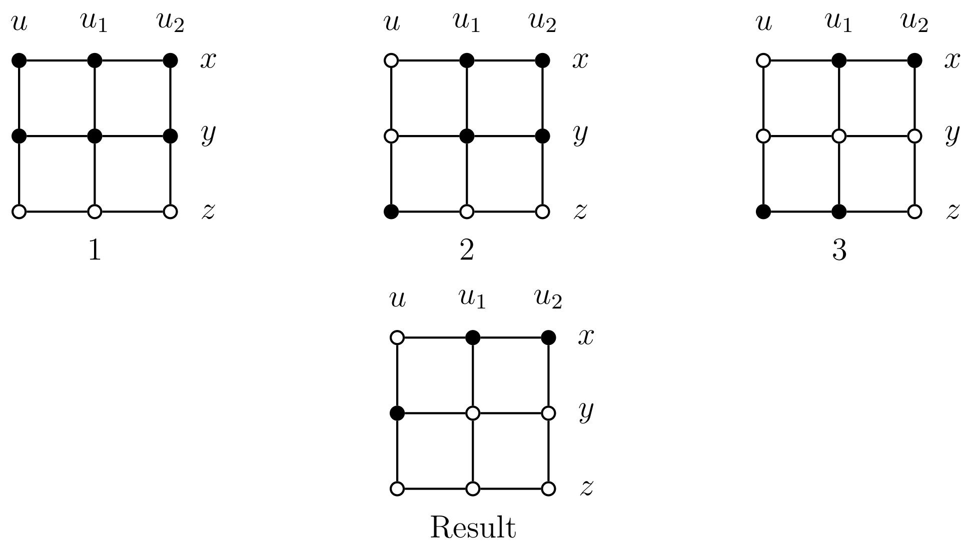
Now, all copies of G have holes in a (except G_{v_1}). Hence, they are solvable. G_{v_1} is similarly solvable, as it has holes in (b, v_1), (c, v_1) and pegs elsewhere.

Solve each copy of G independently, ending with the final two pegs in x and y, which are adjacent. The result is two copies of H (H_x and H_y) filled entirely with pegs. All other copies of H are completely empty, including H_z.

We now make the following series of moves (see Figure 3.12):

1. Jump from (x, u) over (y, u) into (z, u);
2. Jump from (y, u_2) over (y, u_1) into (z, u_1);
3. Jump from (z, u_1) over (z, u) into (y, u).

Figure 3.12: Solving $G \times H$ (H distance 2-solvable)



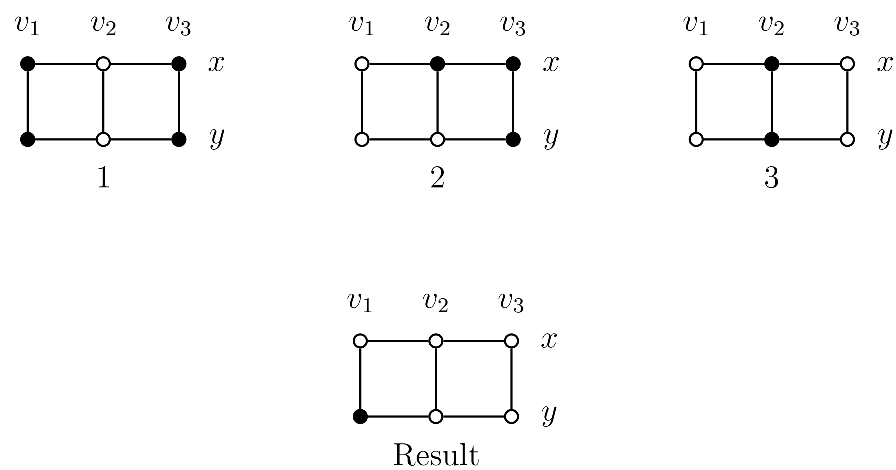
Result: H_x has a hole in (x, u), pegs elsewhere. It is distance 2-solvable ending in $\{(x, v_1),\ (x, v_3)\}$. Since H_y has holes in (y, u_1) and (y, u_2), it is distance 2-solvable ending at $\{(y, v_1),\ (y, v_3)\}$.

Now, final four pegs are in (x, v_1), (x, v_3), (y, v_1), and (y, v_3).

We now make the following series of moves (see Figure 3.13):

1. Jump from (y, v_1) over (x, v_1) into (x, v_2);
2. Jump from (x, v_3) over (y, v_3) into (y, v_2);
3. Jump from (x, v_2) over (y, v_2) into (y, v_1).

Figure 3.13: Solving $G \times H$ (H distance 2-solvable)



Result: One peg in (y, v_1). Hence, $G \times H$ is solved.

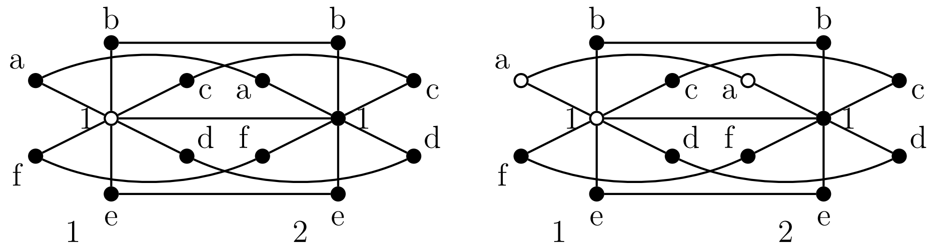
■

After realizing our inability to show the general result that the cartesian product of any solvable graph with any unsolvable graph is solvable, we next present an explicit example of the solvability of the cartesian product of a solvable graph and an unsolvable graph, namely $P_{2m} \times S_n$.

Theorem 3.0.25 *The cartesian product of the even path and the star ($P_{2m} \times S_n$) is solvable.*

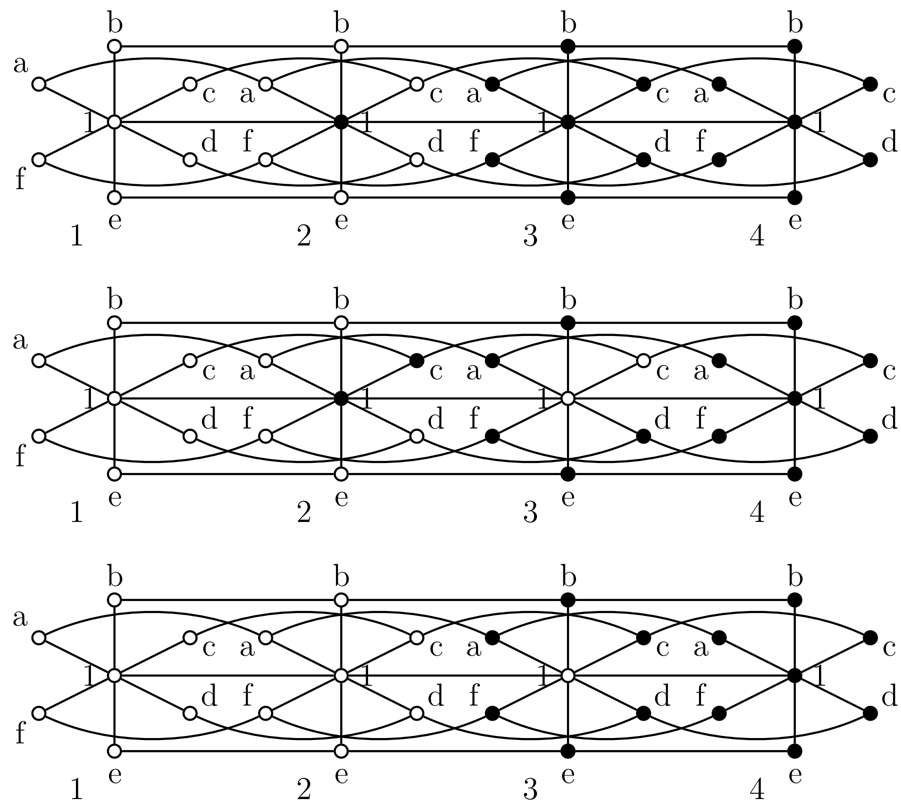
Proof. In order to solve $P_{2m} \times S_n$, we break up the graph into prisms of stars, solving two stars at a time. Let vertex $(a, 1)$ represent vertex a of the first star. Let $(1, i)$ represent the central vertex of the i^{th} star. Begin with the initial hole in vertex $(1, 1)$. Jump from $(a, 2)$ over $(1, 2)$ into $(1, 1)$. Similarly, jump from $(a, 1)$ over $(1, 1)$ into $(1, 2)$. This series of two jumps takes the first two copies of S_n and turns them both into S_{n-1}. After n series of these two jumps, the first two stars are empty entirely except for a lone peg $(1, 2)$.

Figure 3.14: Solving $P_{2m} \times S_n$



Next, jump from $(1, 3)$ over $(c, 3)$ into $(c, 2)$. Then jump back from $(1, 2)$ over $(c, 2)$ into $(c, 3)$. This leaves the first two stars completely empty and a single hole in the central vertex of the third star, vertex $(1, 3)$. From here, we can follow the same algorithm to eliminate pegs completely from the third and fourth stars. Since we crossed the star with an even path, we can continue to solve the stars in sets of two, eventually ending with a single peg in the last star. Hence, $P_{2m} \times S_n$ is solvable. ■

Figure 3.15: Solving $P_{2m} \times S_n$



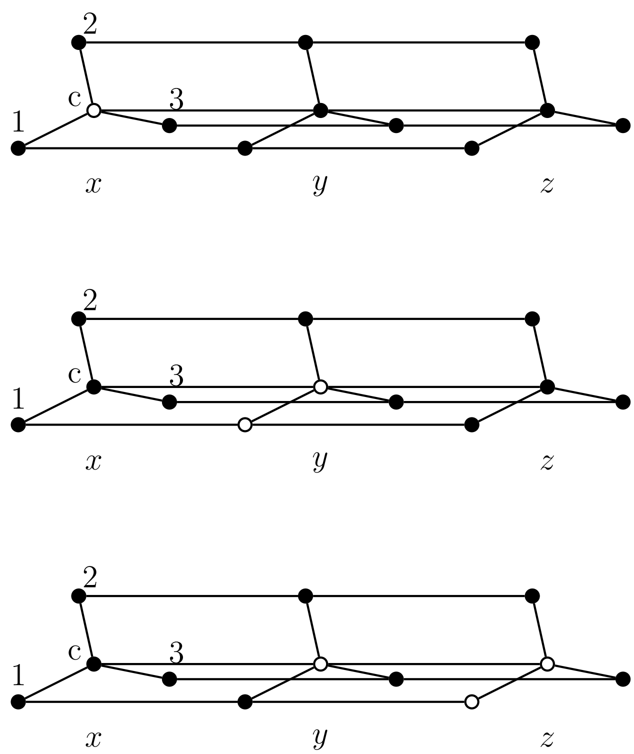
We next present the solvability of the cartesian product of, essentially, two distance 2-solvable graphs, namely $P_n \times S_3$.

Theorem 3.0.26 *The cartesian product of any path and S_3, $P_n \times S_3$, is solvable, $n \geq 3$.*

Proof. Observe first that if n is even, then $P_n \times S_3$ is solvable by Theorem 3.0.25 above. Hence, we will examine the case in which n is odd. Let $(n, 1)$ denote a vertex of the first star. Let (c, i) represent the central vertex of the i^{th} star. Again, begin with the initial hole in the central vertex of the first star, vertex $(c, 1)$. Solve pairs of stars exactly as above. When there are only three stars remaining (solving the same way as above will produce a hole in the central vertex of the third-to-last star), our solving strategy deviates.

Call the three remaining stars X, Y, and Z. By this labeling, we are left with $n-3$ copies of S_3 completely empty, X with a hole in the central vertex, and both Y and Z completely filled with pegs. Jump from $(1, y)$ over (c, y) into (c, x). Next jump from (c, z) over $(1, z)$ into $(1, y)$. See Figure 3.16.

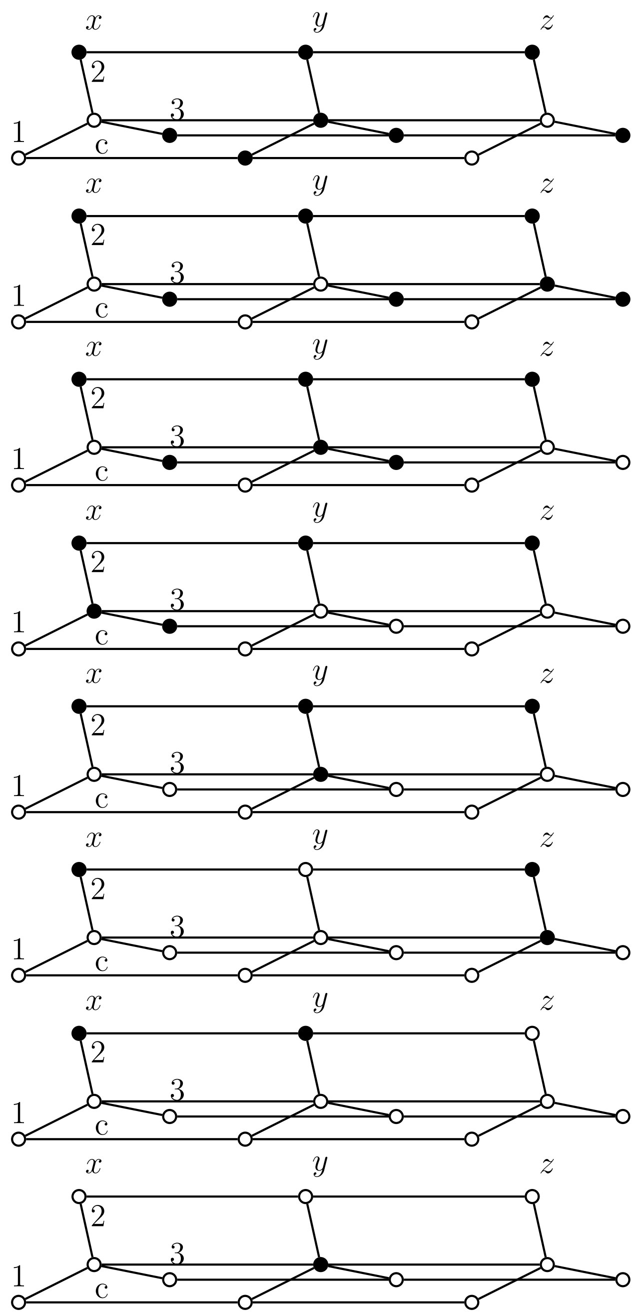
Figure 3.16: Solving $P_n \times S_3$



We now begin making a series of two jumps first from left to right, then right to left. More explicitly, first jump from $(1, x)$ over (c, x) into (c, y). Then, jump from $(1, y)$ over (c, y) into (c, z). Next, make similar jumps back right to left, from $(3, z)$ over (c, z) into (c, y) then from $(3, y)$ over (c, y) into (c, x). One more time, make similar jumps back across left to right. First, jump from $(3, x)$ over (c, x) into (c, y). Then, jump from $(2, y)$ over (c, y) into (c, z). Next, jump from (c, z) over $(2, z)$ into $(2, y)$. Finally, jump from $(2, x)$ over $(2, y)$ into (c, y). See Figure 3.17. Hence, the graph is solved and the result holds.

$\blacksquare$

Figure 3.17: Solving $P_n \times S_3$

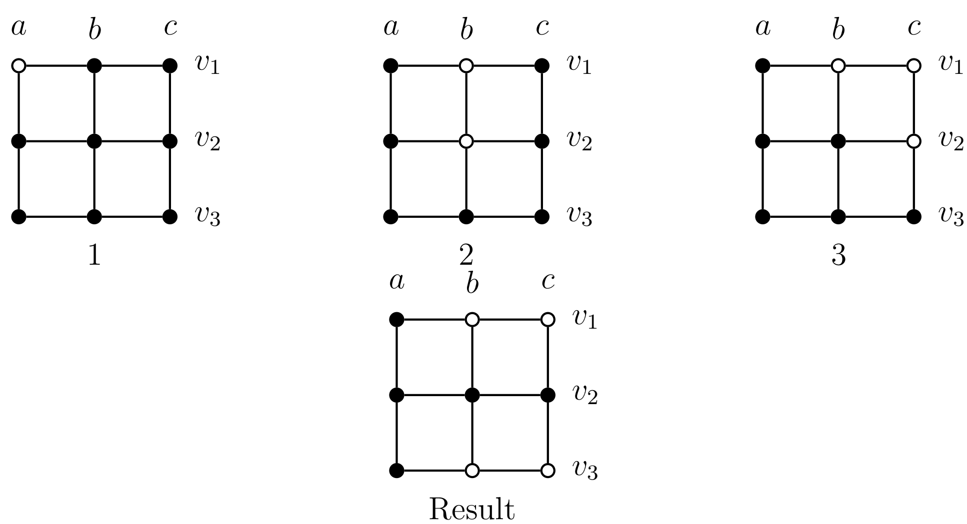


Lastly, we present the solvability of any two distance 2-solvable graphs.

Theorem 3.0.27 *Suppose G and H are both distance 2-solvable graphs. G is distance 2-solvable for initial hole a, leaving pegs in x_1 and x_3. H is distance 2-solvable for initial hole u_1, leaving pegs in v_1 and v_3.*

Proof. Let vertex (x, y) represent the vertex generated by $x \epsilon G$ and $y \epsilon H$. Begin with initial hole in (a, v_1). Make the following series of moves (see Figure 3.18):
1. Jump from (b, v_2) over (b, v_1) into (a, v_1);
2. Jump from (c, v_1) over (c, v_2) into (b, v_2);
3. Jump from (b, v_3) over (c, v_3) into (c, v_2).

Figure 3.18: Solving $G \times H$ (G & H distance 2-solvable)

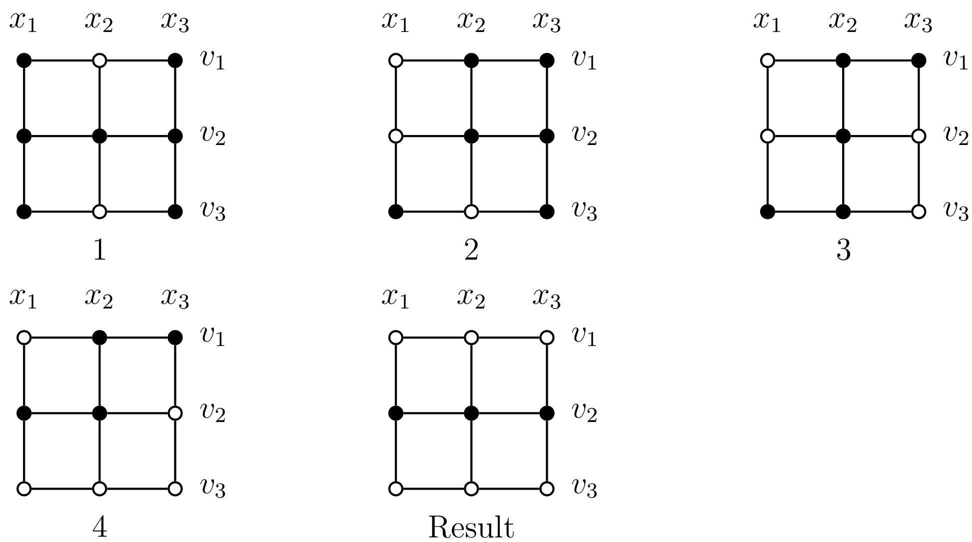


Result: Holes in (b, v_3), (c, v_3), (b, v_1), and (c, v_1).

Now, there are two consecutive holes in G_{v_1} and G_{v_3} representing the two holes formed after the first jump into a of a solving of G. Solve G_{v_1} and G_{v_3} independently leaving pegs in (x_1, v_1), (x_3, v_1), (x_1, v_3), and (x_3, v_3).

We now make the following series of moves (see Figure 3.19):
1. Jump from (x_1, v_2) over (x_1, v_1) into (x_2, v_1);
2. Jump from (x_3, v_2) over (x_3, v_3) into (x_2, v_3);
3. Jump from (x_2, v_3) over (x_1, v_3) into (x_1, v_2);
4. Jump from (x_2, v_1) over (x_3, v_1) into (x_3, v_2).

Figure 3.19: Solving $G \times H$ (G & H distance 2-solvable)

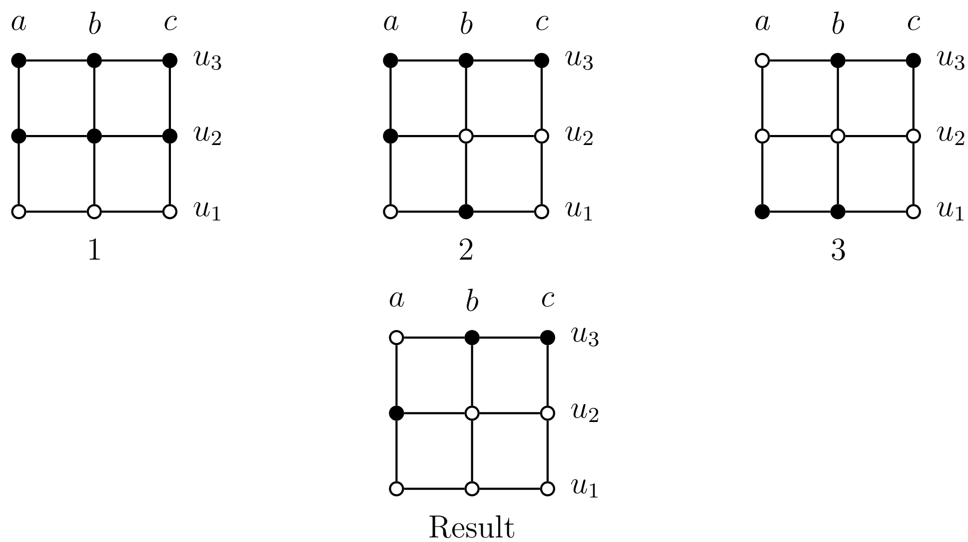


Result: All copies of H have holes in v_1 and v_3. By duality, we can solve these copies ending in u_1. Instead, we solve them leaving the final jump to be made, leaving holes in u_2 and u_3. (u_1 adjacent to u_2 adjacent to u_3)

We now make the following series of moves (see Figure 3.20):

1. Jump from (c, u_2) over (b, u_2) into (b, u_1);
2. Jump from (a, u_3) over (a, u_2) into (a, u_1);
3. Jump from (b, u_1) over (a, u_1) into (a, u_2).

Figure 3.20: Solving $G \times H$ (G & H distance 2-solvable)



Result: The only pegs remaining are in G_{u_2} and G_{u_3}. G_{u_3} has a single hole in a, vertex (a, u_3). G_{u_2} has two holes in b and c, vertices (b, u_2) and (c, u_2), representing the state of G after the initial jump.

Proceed to distance 2-solve G_{u_2} and G_{u_3}. This leaves only four pegs, in vertices (x_1, u_2), (x_1, u_3), (x_3, u_2), and (x_3, u_3).

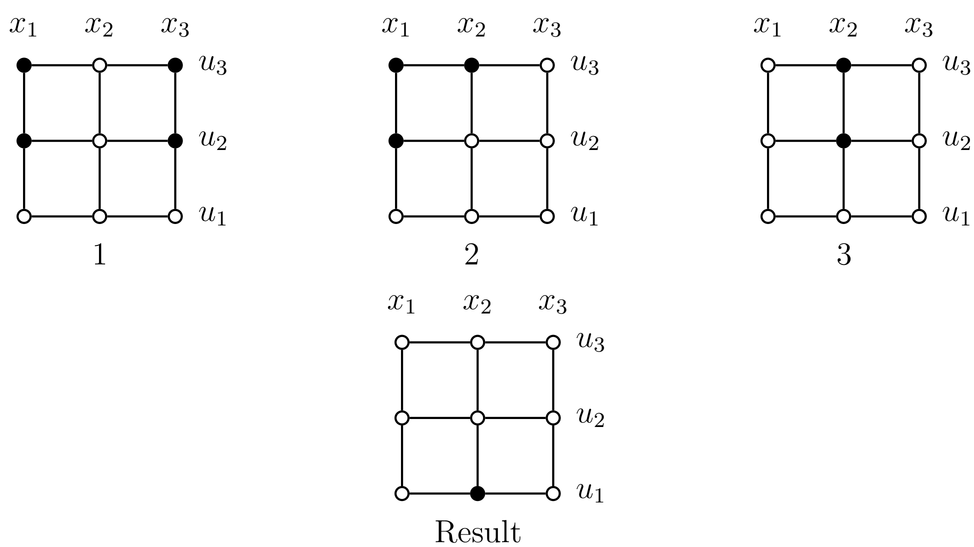
We now make the following series of moves (see Figure 3.21):

1. Jump from (x_3, u_2) over (x_3, u_3) into (x_2, u_3);
2. Jump from (x_1, u_3) over (x_1, u_2) into (x_2, u_2);
3. Jump from (x_2, u_3) over (x_2, u_2) into (x_2, u_1).

Result: The graph $G \times H$ is solved.

■

Figure 3.21: Solving $G \times H$ (G & H distance 2-solvable)

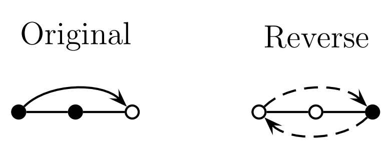


Chapter 4

RELATED GAMES

We define the *Reverse Game* as the game in which the exact opposite of the typical peg solitaire play occurs. In other words, a peg will jump over a *hole* into a hole, turning the jumped hole into a peg. In this game, with each move, the number of pegs goes up and the number of holes goes down, until at best there are $n-1$ pegs on a graph of order n and one hole. On the other hand, in the original game, with each move, the number of pegs went down and the number of holes went up until, at best, there was one peg and $n-1$ holes. The reverse game is equivalent to viewing the empty holes as pegs and the pegs as holes and playing in the original way, only with opposite titles. It can also be thought of as playing the original game backwards, starting with a single peg and building back up the board. Through this thinking, we can see a clear bijection from the reverse game to the original game. This leads us to the big result of this section, showing the equivalence of starting and terminal states. In other words, if a graph is solvable starting at u ending at v, it is also solvable starting at v ending at u. Figure 4.1 shows the equivalency of the reverse game to the original game in P_3.

Figure 4.1: Dually Solving P_3



Though the following theorem is both stated and proved in the introduction, it is also explicitly stated here for emphasis and relevance reasons.

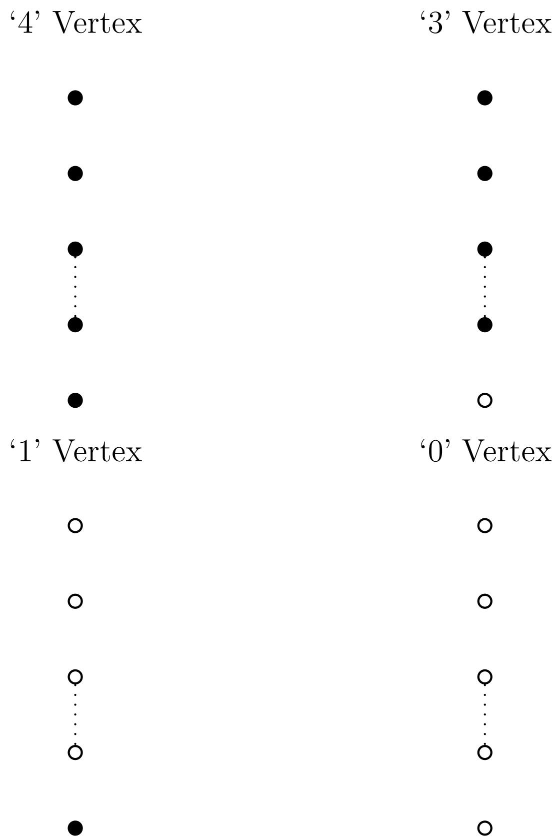
Theorem 4.0.28 *Suppose that S is a starting state of graph G with associated terminal state T. Define the sets S' and T' by reversing the roles of "pegs" and "holes" in S and T, respectively. It follows that T' is a starting state of G with associated terminal state S'.*

Proof. Consider a game that is played in reverse. We begin the *Reverse Game* with a set of pegs in T. In the Reverse Game, if $xy, yz \epsilon E(G)$ with a peg in x and holes in y and z then x can jump over y into z, placing a new peg in y. Clearly, if S is a starting state in the original game with associated terminal state T, then the reverse game has terminal state S associated with T. The Reverse Game is equivalent to the original game. Reversing the roles of "pegs" and "holes" obtains the desired result. ∎

This result is very important and useful in the tree progenation that comes in the next section.

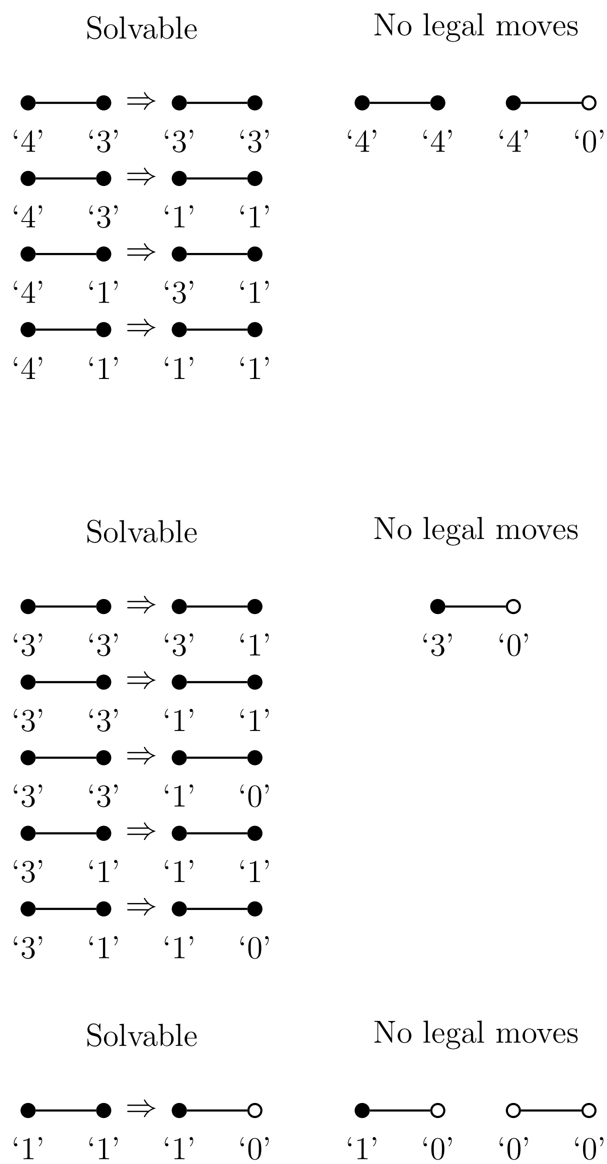
We now invent another game, the *Bipartite Game*. In this game, we have four distinct types of vertices, labeled 4, 3, 1, and 0. The '4' vertices represent the set of independent vertices filled entirely with pegs. The '3' vertices represent the set of independent vertices with one hole, and the remaining vertices filled with pegs. The '1' vertices represent the set of independent vertices with one peg, and the remaining vertices holes. Finally, the '0' vertices represent the set of independent vertices consisting entirely of holes. See Figure 4.2.

Figure 4.2: Types of Vertices in the Bipartite Game



The starting state for this game consists of one vertex labeled ‘3’ and the remaining vertices labeled ‘4.’ The desired ending state is to have one vertex labeled ‘1’ and the remaining vertices labeled ‘0.’ Essentially, each vertex in the game is a partition of a complete bipartite graph. Expanding the labeled vertex idea into actually drawing the graphs as they are, we have multiple partitions of complete bipartite graphs. The above desired ending state is simply the expanded graph with one peg and the rest holes. We must first consider the relationship between adjacent vertices of all possible labels. See Figure 4.3. “Solvable” means, as it originally did, the graph is reduceable to one peg and the remaining vertices holes, which in the Bipartite Game, is equivalent to one ‘1’ vertex and the remaining vertices ‘0’ vertices. “No legal moves” means there are no two pegs adjacent to a hole.

Figure 4.3: Relationships Between Vertex Types in the Bipartite Game



Essentially, the rules in the Bipartite Game correspond directly with how two halves of a complete bipartite graph interact in the old peg game.

Theorem 4.0.29 *Every connected graph can be freely solved with the final '1' vertex ending in any position in the Bipartite Game.*

Proof. It suffices to show the result holds for trees, as every connected graph has a spanning tree. Starting with a '3' on any vertex, it is possible to replace every '4' on the graph with a '1' since $4-3$ and $4-1$ both reduce down to $1-1$ by the rules outlined above. Propagate '1's throughout the graph. Further, we can replace $3-1$ by $1-1$ by the rules. Once all vertices are labeled '1,' choose any vertex to be the '1' at the end. Designate this vertex as the root, r. Begin with the vertices furthest away from r. Relabel these vertices as '0's using the rule $1-1 \rightarrow 1-0$. Remove all vertices labeled '0,' and repeat the process on the remaining graph until only r is labeled with a '1' and all other vertices are labeled '0.' ■

Take a labeled graph G from the Bipartite Game and expand it according to its rules. If it is a '4' vertex, fill all of its constituent vertices with pegs. If it is a '3' vertex, fill all but one of its constituent vertices with pegs. If it is a '1' vertex, fill one of its constituent vertices with a peg. If it is a '0' vertex, make all of its constituent vertices holes. Connect all constituent vertices of one vertex to all constituent vertices of all vertices adjacent to it in G of the Bipartite Game. We now have an ordinary graph formed from a *labeled* graph, call it G_L.

Theorem 4.0.30 *Any graph G_L is freely solvable in the original game.*

Proof. Reduce G_L back down to G with labels in the Bipartite Game with one '3' vertex and the rest '4' vertices. We know this can be freely solved by Theorem 4.0.29. We also know this can be solved with the final '1' vertex wherever we want it by Theorem 4.0.29. Hence, the result holds. ■

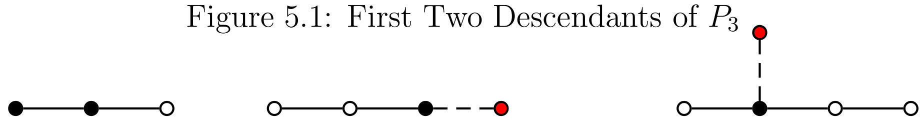
Chapter 5

PROGENATION OF SOLVABLE TREES

We now turn towards creating solvable graphs. In particular, we look at creating solvable trees. There is a theorem which says that all connected graphs have spanning trees as subgraphs. When spanning subgraphs are solvable, the original graphs are solvable. Thus if a spanning tree is solvable, the graph is solvable. Hence, we work on progenating solvable trees. For example, consider P_3, a solvable tree. Solve it, resulting in a peg in one of the end vertices. Append an edge with a peg onto this remaining vertex, resulting in P_4 with two adjacent pegs and two adjacent holes. This graph, expectedly, is also solvable. We continue this process resulting in all solvable trees. See Figure 5.1.

In order to improve this method of progenation, we note that at each level (i.e., a tree on 3 vertices, 4 vertices, etc.) we have only a specific solving of the tree. We choose to consider all solvings on each overlying tree framework (such as P_4). We check each vertex in the overlying tree framework to see if we can solve from that starting state. We also use the idea in the Reverse Game that if you can solve given a certain starting state, you can also solve

Figure 5.1: First Two Descendants of P_3



leaving the final peg in that starting state's initial hole location in order to make sure we find all possible terminal states for a given tree from which to append edges and progenate from. We recognize that all even paths (P_{2n}), $n \geq 3$, will not be produced from this progenation on P_3 since P_{2n+1}, $n \geq 2$ would have to progenate these even paths and we know from Theorem 2.0.11 that P_{2n+1}, $n \geq 2$ is not solvable. Hence, if we progenate from P_3 and each successive even path starting with P_6, this progenation creates almost all solvable trees of order 9 or less. More specifically, this progenation generates 100% of the solvable trees on 3, 4, 5, 6, and 7 vertices and approximately 91% of the solvable trees on 8 and 9 vertices.

The following table consists of all solvable trees with their ancestors and descendants of order 9 or less obtained from the progenation method defined above. See the following tables. The red holes/pegs correspond to all the locations on each graph in which the initial hole can be placed and, equivalently, the final peg can land (up to isomorphisms). For trees of order 9, the red holes/pegs are still legitimate starting/ending locations. However, we no longer guarantee that these locations are the *only* legitimate starting/ending locations on the graph. The final table shows the only three graphs (stemming from one root graph) of solvable trees not found by procreating from P_3, P_6, and P_8 (up to order 9).

Table 5.1: Solvable Trees Found Through Progenation ($n \leq 6$)

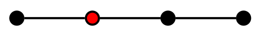
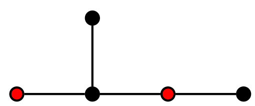
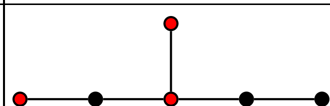
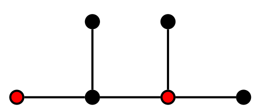
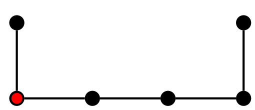
Order	Label	Solvable Tree	Ancestor	Children
3	3a		None	1
4	4a		3a	1
5	5a		4a	2
6	6a		5a	3
6	6b		5a	2
6	6c		None	1

Table 5.2: Solvable Trees Found Through Progenation ($n = 7$)

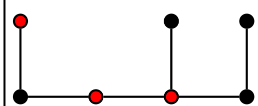
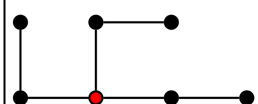
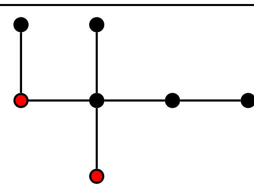
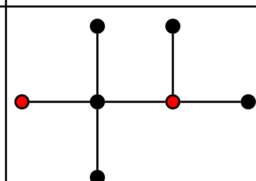
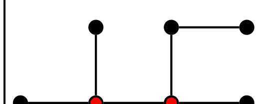
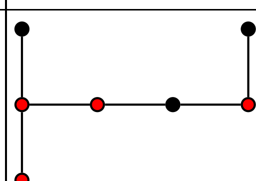
Order	Label	Solvable Tree	Ancestor	Children
7	7a		6a	3
7	7b		6a	1
7	7c		6a	1 1 common child w/7b
7	7d		6b	1 1 common child w/ 7c
7	7e		6b	(1 in common with children of 7c/7d & 7a)
7	7f		6c	3 (1 in common with 7a)

Table 5.3: Solvable Trees Found Through Progenation ($n = 8$)

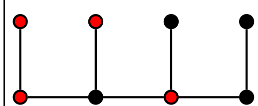
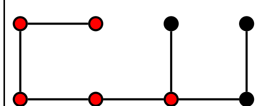
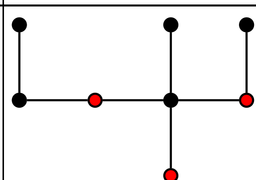
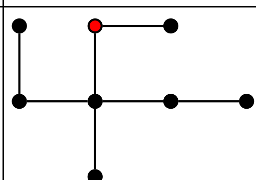
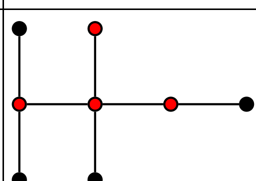
Order	Label	Solvable Tree	Ancestor	Children
8	8a		7a/7e	4
8	8b		7a/7f	4 1 in common with 8a
8	8c		7a	3
8	8d		7b/7c	1
8	8e		7c/7d/7e	3 1 in common w/ 8a & 8d

Table 5.4: Solvable Trees Found Through Progenation ($n = 8$, cont)

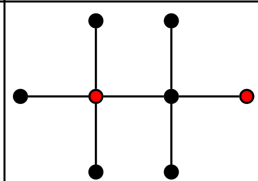
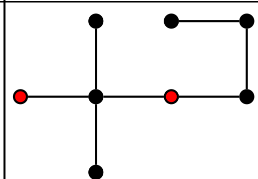
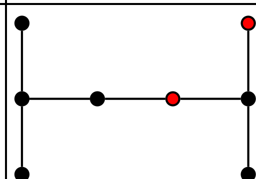
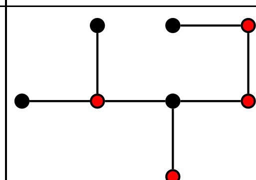
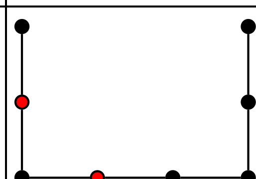
Order	Label	Solvable Tree	Ancestor	Children
8	8f		7d	1 1 in common w 8e
8	8g		7f	2
8	8h		7f	1 1 in common w/ 8b
8	8i		7f	1 1 in common w/ each of 8a, 8g, & 8h
8	8j		None	2

Table 5.5: Solvable Trees Found Through Progenation ($n = 9$)

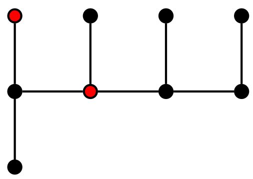
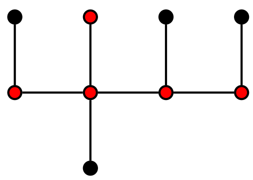
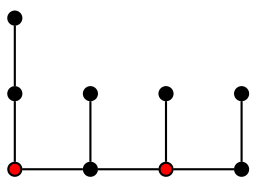
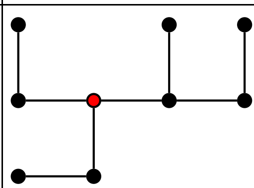
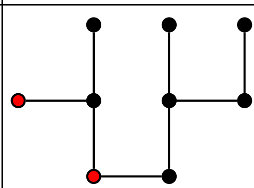
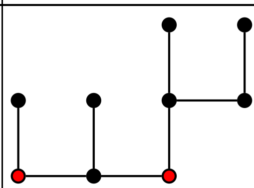
Order	Label	Solvable Tree	Ancestor	Children
9	9a		8a/8i	?
9	9b		8a/8c/8e	?
9	9c		8a/8b	?
9	9d		8a	?
9	9e		8b/8h	?
9	9f		8b	?

Table 5.6: Solvable Trees Found Through Progenation ($n = 9$, cont)

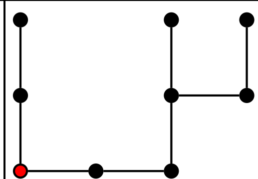
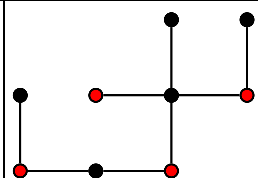
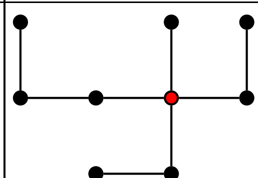
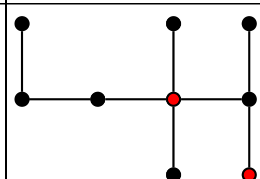
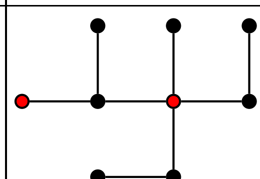
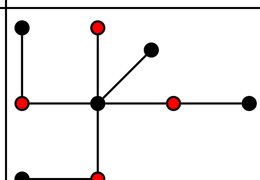
Order	Label	Solvable Tree	Ancestor	Children
9	9g		8b	?
9	9h		8b/8g	?
9	9i		8c	?
9	9j		8c	?
9	9k		8d/8e	?
9	9l		8d	?

Table 5.7: Solvable Trees Found Through Progenation ($n = 9$, cont)

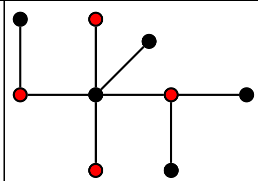
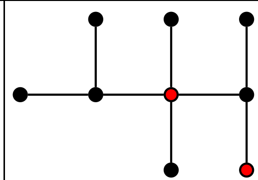
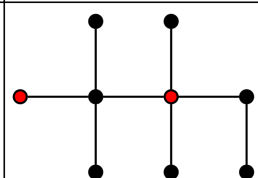
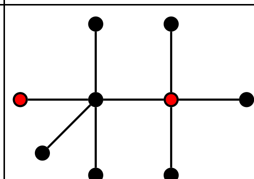
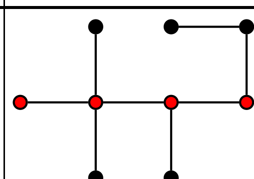
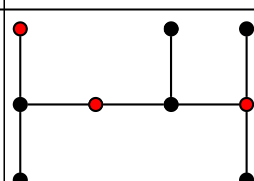
Order	Label	Solvable Tree	Ancestor	Children
9	9m		8e	?
9	9n		8e	?
9	9o		8e/8f	?
9	9p		8f	?
9	9q		8g/8i	?
9	9r		8h/8i	?

Table 5.8: Solvable Trees Found Through Progenation ($n = 9$, cont)

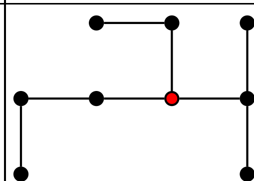
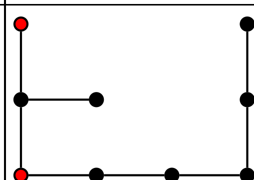
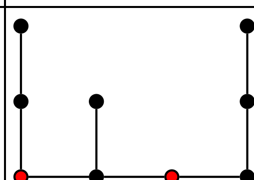
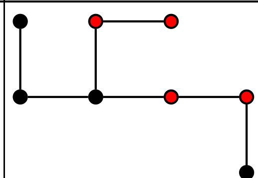
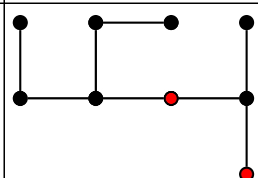
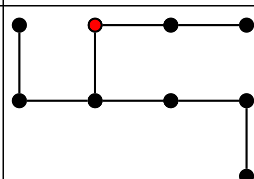
Order	Label	Solvable Tree	Ancestor	Children
9	9s		8i	?
9	9t		8j	?
9	9u		8j	?

Table 5.9: Remaining Solvable Trees up to Order 9

Order	Label	Solvable Tree	Ancestor	Children
8	8k		None	2 1 common w/ 8a & 8i
9	9v		8k	?
9	9w		8k	?

Chapter 6

OPEN PROBLEMS

How many of the 106 trees on 10 vertices are solvable? What are their ancestors?

Which, if any, of the graphs presented as solvable are actually freely solvable?

What are the necessary and sufficient conditions which determine a graph's solvability/unsolvability?

Given a graph G with n vertices and m edges, what is the probability that G is solvable?

Consider the set of all graphs with n vertices and m edges. What is the smallest m such that all graphs in this family are solvable? Freely solvable?

Given a solvable graph G, what is the minimum number of moves required to solve G?

Consider a variation in which we must make jumps when available. Given a graph G, what is the maximum number of pegs we can leave in this variant?

Given graph G with starting state S. Determine the set of terminal states associated with S.

Given two graphs G and H, both distance 2-solvable. What can be said about $G \times H$?

Given a solvable (or freely solvable) graph G. What can be said about $G \times H$ where H is some arbitrary connected graph?

Consider the solvability of other graph products (such as direct, lexicographic, graph composition, etc.).

Suppose we require our final peg to end in the hole we started from. What graphs are solvable now?

What about the solvability of directed graphs, when you can only jump corresponding with the directions given?

Suppose pegs are labeled $\{1, 2, 3, ..., n-1\}$. We require that pegs can only jump those of lower rank. Give a labeling of G that will induce a solution.

Bibliography

[1] Avis, D., & Deza, A. (2001). "On the solitaire cone and its relationship to multi-commodity flows", *Mathematical Programming* **90** (1): 27-57.

[2] Beasley, J. D. (1985). *The Ins and Outs of Peg Solitaire*. Oxford.

[3] Bell, G. I. (2008). "Solving triangular peg solitaire", *Journal of Integer Sequences* **11**: Article 848.

[4] Berlekamp, E., Conway, J., & Guy, R. (1982). *Winning ways for your mathematical plays*. (Vol. 2). New York, NY: Academic Press.

[5] de Bruijn, N. G. (1972). "A solitaire game and its relation to a finite field", *Journal of Recreational Mathematics* **5**: 133-137.

[6] Buckley, F., & Lewinter, M. (2003). *A friendly introduction to graph theory*. Upper Saddle River, New Jersey: Pearson Education.

[7] Cross, D. C. (1968). "Square Solitaire and variations", *Journal of Recreational Mathematics* **1**: 121-123.

[8] Eiss, H. E. (1988). *Dictionary of mathematical games, puzzles, and amusements*. Westport, Connecticut: Greenwood Press.

[9] Gardner, M., "Mathematical games", *Scientific American* **206** (6): 156-166.

[10] Kiyomi, M., & Matsui, T. (2001). "Integer programming based algorithms for peg solitaire problems", *Proc. 2nd Int. Conf. Computers and Games (CG 2000)*. Lecture Notes in Computer Science, **2063**, 229-240.