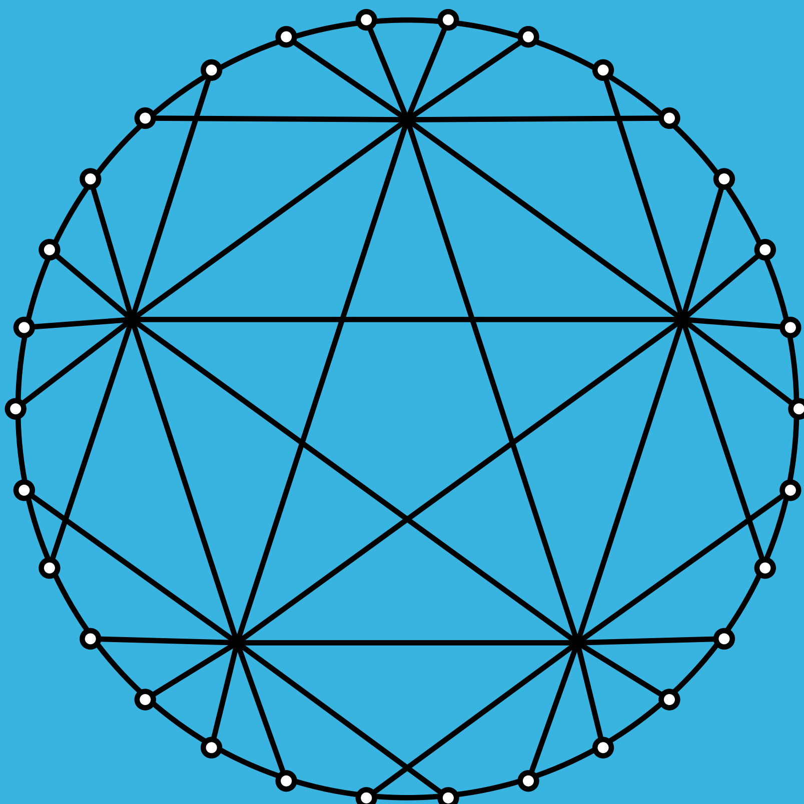
BULLETIN of The INSTITUTE of COMBINATORICS and its APPLICATIONS

Volume 96
October 2022

Editors-in-Chief:
Marco Buratti, Donald Kreher, Ortrud Oellermann, Tran van Trung



Duluth, Minnesota, U.S.A.

ISSN: 2689-0674 (Online)
ISSN: 1183-1278 (Print)

BULLETIN OF THE ICA
Volume **96** (2022), 107–118

Peg solitaire on corona products

Martin Kreh and Jan-Hendrik de Wiljes*

Abstract. In 2011, Beeler and Hoilman generalised the game of peg solitaire to arbitrary connected graphs. Since then, peg solitaire has been considered for quite a few graph classes as well as graph operations, such as the Cartesian product and line graphs. In this paper, we consider peg solitaire on corona products. We give some sufficient criteria for solvability and present a class of graphs that are not solvable. Using some of our results, we can completely answer a question posed by Beeler, Gray, and Hoilman in 2012 concerning corona products of solvable graphs.

1 Introduction

In [3], Beeler and Hoilman introduced the game of peg solitaire on graphs as a generalisation of the classical peg solitaire game:

Given a connected, undirected graph G with vertex set $V(G)$ and edge set $E(G)$, we can put pegs in the vertices of G. Given three vertices u, v, w with pegs in u and v and a hole in w such that $uv, vw \in E(G)$, we can jump with the peg from u over v into w, removing the peg in v (cf. Figure 1). This jump will be denoted as $u \cdot \vec{v} \cdot w$. In figures, vertices with pegs will be drawn filled while vertices without pegs are drawn unfilled.

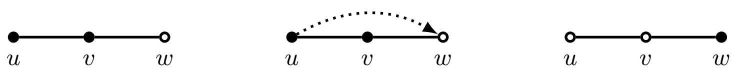


Figure 1: A jump in peg solitaire.

In general, we begin with a *starting state* $S \subseteq V(G)$ of vertices that are empty (i.e., without pegs). A *terminal state* $T \subseteq V(G)$ is a set of vertices that have pegs at the end of the game such that no more jumps are possible. A *terminal state* T is *associated* to a starting state S if T can be obtained

*Corresponding author: jan.dewiljes@math.fu-berlin.de
Key words and phrases: peg solitaire, corona product
AMS (MOS) Subject Classifications: 05C57, 05C76, 91A43

Received: 22 October 2021
Accepted: 29 April 2022

from S by a series of jumps. We will always assume that the starting state S consists of a single vertex. A *configuration* is a set A of vertices such that all elements in A contain a peg and the ones in $V(G) \setminus A$ have a hole.

The goal of the original game is to remove all pegs but one. This is not possible for every graph. If this is indeed possible, then we call G *solvable.* There are more variants of solvability that have been considered in the literature (see for example [3,10]), but we will not deal with those here. For a solvable graph G, a *solution process* of G is a sequence of configurations $A_i, i = 1, \dots, |V(G)| - 1$ with $A_0 = V(G) \setminus \{v\}$ for some $v \in V(G)$ such that A_{i+1} can be reached from A_i using a single jump.

The main goal is the characterisation of solvable graphs. To this end, peg solitaire has been considered for quite a few classes of graphs, including path graphs, complete graphs, star graphs, double stars, and caterpillars (for results see [2–4, 6]). Additionally, the game was examined on graphs obtained via certain graph operations, such as Cartesian products, joins, and line graphs (see [9–11]), leading to the construction of new solvable graphs and graph classes.

Motivated by an open problem in [1], we consider peg solitaire on corona products of graphs. This graph operation has been introduced by Frucht and Harary [8] in 1970 with the goal of defining a graph product such that the automorphism group of this product is (under certain assumptions) connected to the wreath product of the automorphism groups of the respective graphs. The *corona product* $G \circ H$ is defined as follows: Let n be the number of vertices of G. Then $G \circ H$ consists of the graph G and n copies of H, where the i-th vertex of G is joined to each vertex of the i-th copy of H. Denote the copy of H in $G \circ H$ which is joined to vertex $v \in V(G)$ by H_v. Figure 2 shows the graph $C_{10} \circ K_5$, where C_n denotes the *cycle graph* and K_n the *complete graph* on n vertices. In general, we have $G \circ H \neq H \circ G$, i.e., $\circ$ is not commutative.

Note that $G \circ H$ is connected whenever G is connected, hence it makes sense to consider peg solitaire on $G \circ H$ even if H is not connected.

In this paper, we prove the solvability of $G \circ H$ (for arbitrary H) given a solvable graph G and some extra condition involving solution processes of G (Proposition 2.2). We also show, in our main result Theorem 2.3, that $G \circ H$ is solvable when G is solvable (without further assumptions) and H fulfils certain conditions. As an application, we prove that $G \circ G$ is solvable for solvable G, answering the open question from [1] mentioned above. Finally, we obtain infinite families of (connected) graphs G and (unconnected) graphs H such that $G \circ H$ is not solvable.

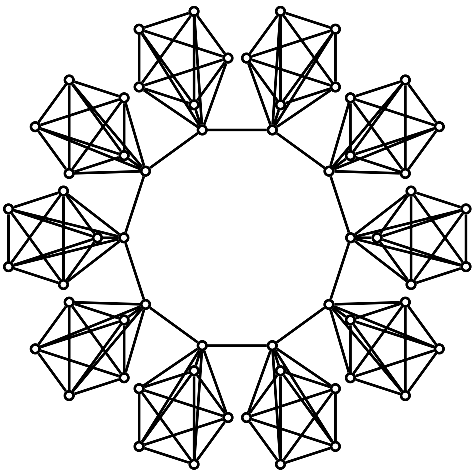


Figure 2: The graph $C_{10} \circ K_5$.

The following notations and notions will be used throughout this paper. The *union* of graphs is understood as their disjoint union in this paper, i.e., the union of G and H has vertex set $V(G) \cup V(H)$, where $V(G) \cap V(H) = \{\}$, and edge set $E(G) \cup E(H)$, where $E(G) \cap E(H) = \{\}$. For a graph G and a set of vertices $W \subseteq V(G)$, the *subgraph* of G *induced* by W, denoted by $G[W]$, is the graph with vertex set W and edge set $\binom{W}{2} \cap E(G)$, where $\binom{W}{2}$ denotes the set of all subsets of W with exactly two elements, i.e., all possible edges between vertices in W. A subgraph H of G is *spanning* if $V(H) = V(G)$. The *path graph* on n vertices is denoted by P_n and the *complete bipartite graph* such that one part has m vertices and the other n vertices is denoted by $K_{m,n}$, where the special case $K_{1,n}$ is called a *star graph*. Vertices of degree 1 are called *pendant*. A *double star* $\mathrm{DS}(L, R)$ consists of two adjacent vertices c_L, c_R, where additionally c_L is adjacent to L pendant vertices and c_R is adjacent to R pendant vertices. A matching M of a graph G is called *perfect matching* if $2|M| = |V(G)|$.

We will often employ the concept of double star purges, cf. [4, 5], which are basically shortcuts defined via combining several consecutive jumps. Consider the double star $\mathrm{DS}(L, R)$. If $L = R$, then we choose the starting state $\{c_L\}$ (choosing $\{c_R\}$ as the starting state works similarly). Repeating the process of alternately jumping from a pendant adjacent to c_R into c_L and from a pendant adjacent to c_L into c_R solves this graph such that the final peg is in c_R, i.e., the centre vertex that contained a peg at the beginning. This process is called a *double star purge*.

2 Results

As a foretaste of what is to come, we start with a general result using known facts about the solvability of even path graphs. Recall that a *Hamiltonian path* in a graph G is a path that includes all vertices of G (exactly once).

Proposition 2.1. *If G has an even number of vertices and contains a Hamiltonian path, then $G \circ H$ is solvable for all (not necessarily connected) graphs H.*

Proof. It has been shown in [3] that any graph G with an even number of vertices that has a Hamiltonian path is solvable. To solve G, take the Hamiltonian path $P = v_1, v_2, \ldots, v_n$, start with a hole in v_2, and jump alternating from right to left and from left to right. This results in a configuration where the final peg is in the vertex v_{n-1}. Note that for any $i \in \{1, 2, \ldots, \frac{n}{2}\}$ at some point of this process there is exactly one peg and one hole among the vertices v_{2i-1}, v_{2i}.

Now we can solve $G \circ H$ as follows: Start solving the graph G (as a subgraph of $G \circ H$) using the Hamiltonian path P of G. For each $i \in \{1, 2, \ldots, \frac{n}{2}\}$, whenever we reach a configuration such that exactly one of v_{2i-1}, v_{2i} has a peg, pause solving G. Consider the subgraph of $G \circ H$ induced by these two vertices and the vertices of the two copies of H attached to them. This subgraph contains a double star as a spanning tree, which we solve as described above, ignoring the additional edges. At the end of this process, all vertices in the two copies of H will be empty and the situation on the Hamiltonian path P remains the same. Hence, we can continue solving G using P and eliminate the pegs in copies of H as soon as we reach these copies in the solution process, using double star purges as described above. $\square$

This result can be extended in the following two ways: by weakening the conditions on G but not the ones on H (Proposition 2.2) or by allowing (even) more graphs G and restricting H (Theorem 2.3). The statement corresponding to the first extension can be proven analogously, while the second one needs additional ideas.

Proposition 2.2. *Let G be a solvable graph for which a perfect matching M and a corresponding solution process exist such that for each $uv \in M$ there is a configuration in the solution process such that exactly one of u and v contains a peg. Then $G \circ H$ is solvable for all (not necessarily connected) graphs H.*

For the second extension, we use an idea from [7] to decompose the vertex set of a tree H in the following way: Let u be a vertex of degree 1 and v the neighbour of u. Then $H[\{v\} \cup N_v]$, where N_v denotes the set of neighbours of v with degree 1, is a star graph. Iterating this process on the connected components of $H[V(H) \setminus (\{v\} \cup N_v)]$ yields vertex sets $V_1, V_2, \ldots, V_\ell$ (for some positive integer ℓ), where $V_i \subseteq V(H)$ and $H[V_i]$ contains a spanning star graph. This procedure can be applied to arbitrary graphs component-wise, i.e., choose a spanning forest of H with the same number of connected components as H; additionally to using vertices of degree 1 (as described above), each isolated vertex of H forms one V_i. Every sequence of vertex sets obtainable in this way is called a *vertex star decomposition*[1] of H. The largest possible ℓ in such a decomposition $V_1, V_2, \ldots, V_\ell$ is the *vertex star decomposition number* $\mathrm{vsd}(H)$ of H and a corresponding sequence of V_i is called a *maximum vertex star decomposition* of H. Note that $\mathrm{vsd}(H) = 1$ if and only H is a star graph or $H = K_3$. Also note that the number of V_i of size 1 coincides with the number of isolated vertices in H.

Vertex star decompositions of H induce, when considering $G \circ H$, in some sense (see Subcase 2.3.1 of the upcoming proof) the following graph class, which plays a major role in the proof of our main result. A *windmill variant* $\mathrm{W}(P, B)$ is a graph with a universal vertex u, i.e., a vertex adjacent to every other vertex, P pendant vertices, which are only adjacent to u, and B blades consisting of two vertices each, such that these two vertices are adjacent. Furthermore, $\mathrm{W}(B) = \mathrm{W}(0, B)$ is called a *windmill graph*. An example, the graph $\mathrm{W}(2, 3)$, is displayed in Figure 3.

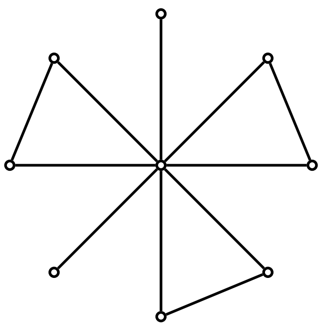


Figure 3: The windmill variant $\mathrm{W}(2, 3)$.

[1]This is not to be confused with the star decomposition of a graph, which works with a partition of the edge set.

Theorem 2.3. *Let G be solvable and H be a graph with at most $\frac{1}{2}|V(H)|-1$ isolated vertices such that H is not one of the following:*

- *K_2 or K_3,*
- *union of $K_{1,n}$ and mK_1 ($m \in \mathbb{Z}_{\geq 0}$) such that $n-m$ is even,*
- *union of an odd number of K_1 and an arbitrary number of K_2.*

Then $G \circ H$ is solvable.

Proof. Fix a solution process of G. We characterise the vertices of G in the following way: Let $x_1, x_2, \ldots, x_q$ be those vertices which do not have a neighbour containing a peg whenever they are empty. For each x_i there is a unique vertex y_i for which $x_i \cdot \vec{y_i} \cdot w_i$, for some vertex w_i, is the last jump involving either one of x_i, y_i. Note that we even have $y_i \neq y_j$ if $i \neq j$. Let Z denote the set of vertices of G that do not occur as x_i or y_i.

Start solving G (as a subgraph of $G \circ H$). We pause this process in one of the following two cases to eliminate all pegs from certain copies of H.

<u>Case 1:</u> Suppose the next jump would be $x_i \cdot \vec{y_i} \cdot w_i$ and this would be the last jump involving either one of x_i, y_i. Then we instead jump from a vertex of H_{y_i} over y_i into w_i. Next we perform a double star purge on the graph induced by $V(H_{x_i}) \cup V(H_{y_i}) \cup \{x_i, y_i\}$ until there are only pegs in some $a \in V(H_{x_i})$, some $b \in V(H_{y_i})$, and y_i. Jump $w_i \cdot \vec{y_i} \cdot x_i, a \cdot \vec{x_i} \cdot y_i, b \cdot \vec{y_i} \cdot w_i$ and continue solving G.

<u>Case 2:</u> Now we deal with the situation when some $z \in Z$, for the first time in the solution process, does not contain a peg but one of its neighbours, say u, does. In each of the forthcoming subcases, we empty H_z completely while keeping the peg situation in G as it was before. After that, proceed solving G.

<u>Subcase 2.1:</u> If H_z is a star graph, then denote by c its centre and by p an arbitrary pendant and jump $p \cdot \vec{c} \cdot z, u \cdot \vec{z} \cdot c$. As the number $2r$ of pegs in $H_z \setminus \{p, c\}$ is even and at least 2, the configuration in $G[\{z\}] \circ H$ can be reduced (using a double star purge: the centres are c and z, the vertices from $H_z \setminus \{p, c\}$ are viewed as $r+1$ of them belonging to c and $r-1$ belonging to z) such that only two pegs remain, namely in z and some neighbour v of c. Jump $v \cdot \vec{z} \cdot u$ to empty H_z.

<u>Subcase 2.2:</u> Suppose H_z is the union of one star graph $K_{1,n}$ ($n \geq 1$), denote its centre by c, and $m \geq 1$ graphs K_1. If $m \leq \frac{1}{2}|V(H)| - 2$, then we get

$m \leq n-3$. In this case, we can remove all pegs from vertices which are isolated in H_z by successively jumping with a peg from $K_{1,n}$ into z and with a peg from such an isolated vertex over z into c. Since $n-m$ is odd and at least 3, we can continue as if H_z was a star graph (see previous subcase). If $m = \frac{1}{2}|V(H)| - 1$, then we get $m = n-1$. In this case, we use the same jumps as above until we arrive at the situation depicted in Figure 4, which can be solved such that the final peg is in u.

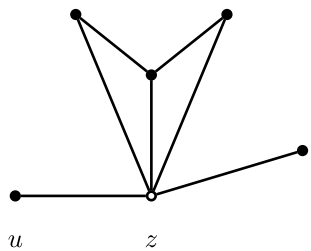


Figure 4: Base case for Subcase 2.2 when $m = n-1$.

Subcase 2.3: For any other H_z, fix a maximum vertex star decomposition $V_1, V_2, \ldots, V_{\mathrm{vsd}(H_z)}$ of it. Without loss of generality we may arrange the V_i such that some integer $k > 1$ with $|V_i| \geq 2$ for all $i \leq k$ and $|V_i| = 1$ for all $i > k$ exists (note that we can choose $k = \mathrm{vsd}(H_z)$ if H_z has no isolated vertices). Furthermore, in case H_z has a vertex of degree at least 2, say v, assume $v \in V_1$. Denote the isolated vertices of H_z by $b_1, b_2, \ldots, b_{\mathrm{vsd}(H)-k}$. Our main goal is to reach one of the situations displayed in Figures 5 and 6, which are all solvable with final peg in u.

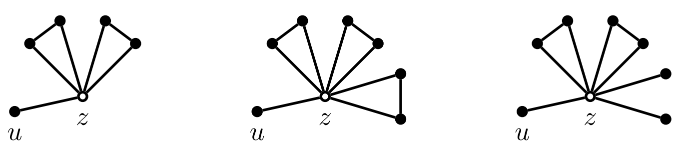


Figure 5: Base cases for Subcase 2.3 when P is even.

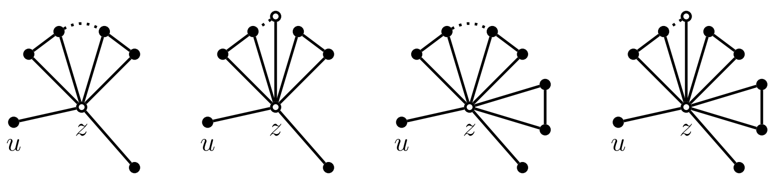


Figure 6: Base cases for Subcase 2.3 when P is odd.

To this end, we first eliminate as many pegs from isolated vertices of H_z as possible while still keeping at least two pegs in each V_i with $i \leq k$. As long as some V_i contains more than two pegs and some b_j is not empty, jump from a pendant in V_i over its centre into z and with the peg from b_j back into the centre of V_i.

We reach one of two situations: Either at least one b_i contains a peg or all b_i are empty.

Subcase 2.3.1: In the first case, the subgraph of $G \circ H$ induced by $\{v \in (V(H_z) \setminus V_1) : v \text{ contains a peg}\} \cup \{z\}$ contains a subgraph which is isomorphic to some windmill variant $\mathrm{W}(P, k-1)$, where $P \leq 2k-2$ holds by the restriction on the number of isolated vertices of H. Using the ideas from [4], namely removing all pegs from one blade and two pendants or blades pairwise as soon as only one or no b_i contains a peg (cf. Figure 7), this reduces to one of the solvable situations given in Figures 5 and 6.

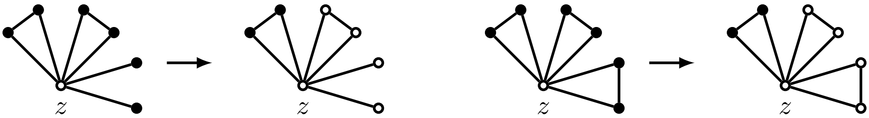


Figure 7: Reduction of blades and pendants.

Note that we keep exactly one or two blades full (i.e., they contain two pegs), and the additional blade shown in Figures 5 and 6 comes from V_1. Also note that we have to use v and the additional edge, when, after the reduction, exactly one of the b_i still contains a peg; hence we had to exclude the graph unions given in the statement of Theorem 2.3.

Subcase 2.3.2: In the latter case, we can successively remove pegs from the V_i until we reach a solvable situation given in Figure 5 in the following way. Start by removing pegs from every V_i containing at least four pegs by repeatedly jumping from the centre of V_i into z and back into the centre (compare top part of Figure 8) until only two or three pegs remain in V_i.

Then, for every $i, j \geq 1$ with $i \neq j$ and the number of pegs in V_i, V_j being three each, remove one peg from V_i and V_j using a double star purge given on the bottom part of Figure 8. Now at most one V_i contains three pegs. As before, using the results from [4], we can successively empty blades pairwise until we reach one of the solvable situations given in Figure 5 (if no V_i contains three pegs) or Figure 9 (if some V_i contains three pegs). $\square$

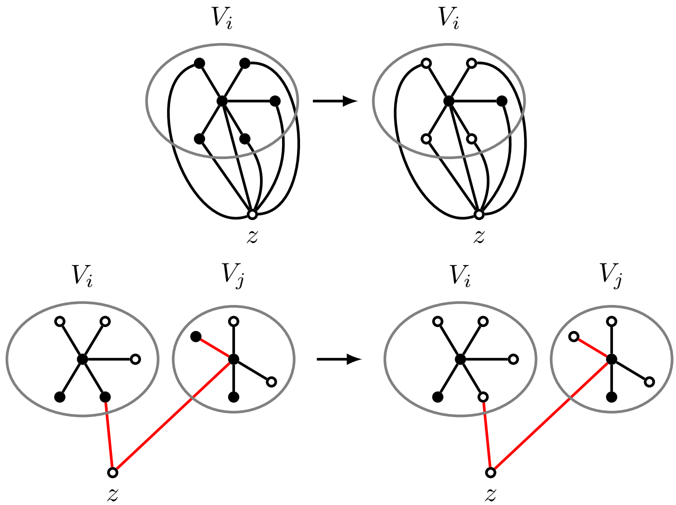


Figure 8: The top picture displays the reduction of pegs in V_i to 2 or 3. The bottom picture illustrates the iterative reduction of the pegs in V_i, V_j, where $i \neq j$, to only 2. The red edges form the double star DS$(1,1)$ that is used for the reduction.

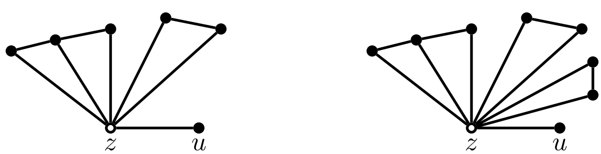


Figure 9: Base cases for Subcase 2.3.2 when there is a V_i that contains three pegs.

The graph $G \circ G$, i.e., the special case of $G \circ H$ where $G = H$, is called the *corona* of G in the literature. The following result, which is a straightforward corollary of Theorem 2.3 when realising that $K_2 \circ K_2, P_3 \circ P_3$, and $K_3 \circ K_3$ are solvable, resolves a problem posed in [1].

Theorem 2.4. *The corona $G \circ G$ of a solvable graph G is also solvable.*

For the remaining cases of connected H in Theorem 2.3, i.e., $H = K_1, H = K_2, H = K_3$ or $H = K_{1,n}$ with even n, the arguments used in the proof do not work. At least for $H = K_2$ however, we believe the statement in the theorem still to be true. This conjecture is supported by the following proposition which shows that $G \circ K_2$ is even solvable for star graphs G, i.e., for graphs that are in some sense furthest away from being solvable.

Proposition 2.5. *1. $K_{1,n} \circ K_2$ is solvable for all n.*

2. $P_n \circ K_2$ is solvable for all n.

Proof. 1. Denote the vertex in $K_{1,n} \circ K_2$ corresponding to the centre of $K_{1,n}$ by c, the pendants by x_i, the vertices corresponding to K_2 adjacent to x_i by y_i and z_i, and the vertices corresponding to K_2 adjacent to c by c_1 and c_2. For $n = 1$, Proposition 2.1 tells us that $K_{1,1} \circ K_2$ is solvable. If $n = 2$, start with a hole in c and jump

$$y_1 \cdot \vec{x}_1 \cdot c, x_2 \cdot \vec{c} \cdot x_1, z_1 \cdot \vec{x}_1 \cdot c, y_2 \cdot \vec{z}_2 \cdot x_2, x_2 \cdot \vec{c} \cdot x_1, c_1 \cdot \vec{c}_2 \cdot c, x_1 \cdot \vec{c} \cdot c_1,$$

ending with a peg in c_1.

If $n = 3$, use the same jumps as for $n = 2$ after which the only pegs left are in c_1, x_3, y_3, z_3. Jump

$$y_3 \cdot \vec{x}_3 \cdot c, c_1 \cdot \vec{c} \cdot x_3, z_3 \cdot \vec{x}_3 \cdot c$$

to solve the graph.

Let now $n \geq 4$ and start again with a hole in c. Then the jumps

$$c_1 \cdot \vec{c}_2 \cdot c, x_n \cdot \vec{c} \cdot c_1, y_{n-1} \cdot \vec{x}_{n-1} \cdot c, c_1 \cdot \vec{c} \cdot x_{n-1}, z_{n-1} \cdot \vec{x}_{n-1} \cdot c, z_n \cdot \vec{y}_n \cdot x_n, x_n \cdot \vec{c} \cdot c_2$$

followed by

$$y_{n-2} \cdot \vec{x}_{n-2} \cdot c, x_{n-3} \cdot \vec{c} \cdot x_{n-2}, z_{n-3} \cdot \vec{y}_{n-3} \cdot x_{n-3}, z_{n-2} \cdot \vec{x}_{n-2} \cdot c, x_{n-3} \cdot \vec{c} \cdot c_1$$

empty four copies of $K_1 \circ K_2$ corresponding to pendants in $K_{1,n}$. Iterating this reduction eventually yields one of the base cases $K_{1,n'} \circ K_2$ with $n' \in \{1, 2, 3, 4\}$. The first three cases have already been discussed, while if $n' = 4$, then we finally jump $c_1 \cdot \vec{c}_2 \cdot c$ to solve the graph.

2. If n is even, then the statement follows from Proposition 2.1, so let n be odd from now on. If $n = 3$, then we have $P_3 = K_{1,2}$, so $P_3 \circ K_2$ is solvable due to the first statement of this proposition.

If $n > 3$, then we can proceed as described in Proposition 2.1 – the difference being that we cannot empty all vertices of P_n and not all vertices attached to them. We are left with a subgraph of $P_n \circ K_2$ isomorphic to $P_3 \circ K_2$ which can be solved as described above. □

Proposition 2.5 does not answer the question of whether $G \circ K_2$ is solvable for solvable G, but it might help in the following way: Suppose G is solvable, then we can consider a spanning tree T of G and decompose T into path and star graphs which may lead to a way, incorporating Proposition 2.5, to solve $G \circ K_2$. A similar idea has been used in [10] to show that certain Cartesian products are solvable. To this end, it was necessary to prove that certain base cases, which for corona products are the ones mentioned in Proposition 2.5, are super freely solvable (meaning that we can freely choose the position of the starting hole and the final peg). It is unclear whether the corona products given in Proposition 2.5 have this property.

Although it might be reasonable to suspect that all corona products $G \circ H$ with connected G are solvable, this is not the case. There is at least one family of counterexamples, which, not surprisingly, involves star graphs. Recall that $\overline{G}$ denotes the *complement* of the graph G.

Consider the graph $K_{1,n} \circ \overline{K_k}$ which is a tree of diameter four if $n > 1$, i.e. for any two vertices a path between them of length at most 4 exists. The solvability of these trees has been examined in [6]. In the notation of that article, $K_{1,n} \circ \overline{K_k}$ is the graph $K_{1,n}(k; k, \ldots, k)$. This graph is solvable if and only if $0 \leq k + n - kn \leq n + 1$ and $k \geq 2$. Since $kn > k + n$ for $n, k \geq 2, (n, k) \neq (2, 2)$, it is only solvable if $(n, k) = (2, 2)$. Thus, we have established the following result.

Proposition 2.6. *Let n and k be integers greater than 1. The graph $K_{1,n} \circ \overline{K_k}$ is not solvable unless $(n, k) = (2, 2)$.*

3 Concluding remarks

The following two aspects should be part of future research. It would be desirable to settle the cases in Theorem 2.3 where H is one of the forbidden graphs. At least for all H without isolated vertices we believe the statement still to be true. Note however that, in view of Proposition 2.6, the statement cannot be true for arbitrary H as $G = K_{1,2} = P_3$ is solvable but $P_3 \circ \overline{K_k}$ is not for $k > 2$. Moreover, is it possible to construct more unsolvable corona products $G \circ H$ with connected G, e.g., if G is a star graph? We suspect such graphs to be rare.

References

[1] R.A. Beeler, A.D. Gray, and D.P. Hoilman Constructing solvable graphs in peg solitaire, *Bull. Inst. Combin. Appl.*, **66** (2012), 89–96.

[2] R.A. Beeler, H. Green, and R.T. Harper, Peg solitaire on caterpillars, *Integers*, **17** (2017), Art. G1.

[3] R.A. Beeler and D.P. Hoilman, Peg solitaire on graphs, *Discrete Math.*, **311 (20)** (2011), 2198–2202.

[4] R.A. Beeler and D.P. Hoilman, Peg solitaire on the windmill and the double star graphs, *Australas. J. Combin.*, **52** (2012), 127–134.

[5] R.A. Beeler and C.A. Walvoort, Packages and purges for peg solitaire on graphs, *Congr. Numer.*, **218** (2013), 33–42.

[6] R.A. Beeler and C.A. Walvoort, Peg solitaire on trees with diameter four, *Australas. J. Combin.*, **63 (3)** (2015), 321–332.

[7] J.-H. de Wiljes and M. Kreh, Making graphs solvable in peg solitaire, *preprint*.

[8] R. Frucht and F. Harary, On the corona of two graphs, *Aequationes Math.*, **4** (1970), 322–325.

[9] M. Kreh and J.-H. de Wiljes, Peg solitaire on line graphs, *preprint*.

[10] M. Kreh and J.-H. de Wiljes, Peg solitaire on Cartesian products of graphs, *Graphs Comb.*, **37 (3)** (2021), 907–917.

[11] S. Loeb and J. Wise, Fool's solitaire on joins and Cartesian products of graphs, *Discrete Math.*, **338 (3)** (2015), 66–71.

Martin Kreh
University of Hildesheim, Hildesheim, Germany
kreh@imai.uni-hildesheim.de

Jan-Hendrik de Wiljes
Freie Universität Berlin, Berlin, Germany
jan.dewiljes@math.fu-berlin.de