The Angel Problem, Positional Games, and Digraph Roots

Strategies and Complexity

Dissertation zur Erlangung des Doktorgrades

vorgelegt am

Fachbereich Mathematik und Informatik
der Freien Universität Berlin

2004

von

Martin Kutz

Institut für Mathematik
Freie Universität Berlin
Arnimallee 2
14195 Berlin
Germany
kutz@math.fu-berlin.de

Betreuer und erster Gutachter:

Prof. Dr. Martin Aigner
Freie Universität Berlin
Arnimallee 2
D-14195 Berlin
Germany
aigner@math.fu-berlin.de

zweiter Gutachter:

Prof. Dr. Gyula O. H. Katona
Alfréd Rényi Institute of Mathematics,
Hungarian Academy of Sciences
H-1053 Budapest, Reáltanoda u. 13–15
Hungary
ohkatona@renyi.hu

Tag der Disputation: 7. Juli 2004

Preface

This thesis is about combinatorial games—mostly. It is also about graphs, directed graphs and hypergraphs, to a large extent; and it deals with the complexity of certain computational problems from these two areas. We study three different problems that share several of the above aspects, yet, they form three individual subjects and so we treat them independently in three self-contained chapters:

The angel-devil game. In the first chapter, we present improved strategies for an infinite game played on an infinite chess board, which has been introduced by Berlekamp, Conway, and Guy [**8**]. The *angel*, a chess person who jumps from square to square, tries to escape his opponent, the *devil*, who intends to strand the angel by placing obstacles on the board.

The open question about this game is, whether some angel who is allowed to make sufficiently large but bounded steps in each move, will be able to escape forever. Conway [**11**] has shown that certain quite natural escape attempts are bound to fail.

We attack this problem from the devil's perspective, trying to improve upon a result from [**8**], which established that the ordinary chess king, who can be considered as an angel of minimal power, cannot escape. A reformulation of the game which focuses on the angel's speed as the crucial quantity, allows us to show that certain faster "chess kings" can still be trapped. A second part of this chapter deals with angels on a three-dimensional board. We show that the new dimension grants the angel enough freedom to escape forever.

Weak positional games on hypergraphs. The games in the second chapter are very general versions of the well-known game of Tic-Tac-Toe. Two players alternately claim vertices of a hypergraph, the first player trying to get all vertices within some edge, his opponent striving to prevent this from happening.

Such *weak positional games* are known to be PSPACE-complete, but the respective hardness result from [**39**] utilizes edges of size up to 11. We analyze the restricted class of hypergraphs whose edges contain no more than three vertices each, trying to find optimal strategies for both players. We almost succeed. Under the additional restriction of almost-disjointness, that is, any two edges may share at most one vertex, we obtain a classification of such hypergraphs into those that yield a first player win and those who don't, which immediately leads to efficiently computable optimal strategies for either player. Eventually, a new framework is introduced for describing values that individual parts of a hypergraph contribute to a game that is played on the whole hypergraph.

The complexity of digraph root computation. The final chapter is not about games. A kth *root* of a square Boolean matrix A is some other matrix R with $R^k = A$. Interpreting A as a the adjacency matrix of a directed graph (digraph), we get an induced notion of powers for digraphs: the digraph D^k has an arc from a to b iff there is a walk of length exactly k from a to b in the digraph D.

The computational complexity of deciding whether a given Boolean matrix or, equivalently, a given digraph has a kth root, has been an open problem for twenty years. We answer this question by proving the problem NP-hard for every single integer $k \geq 2$. Our NP-completeness proof takes the graph-theoretic view, using basic concepts like paths, cycles, and vertex neighborhoods.

Besides the phenomena that make root finding hard, we discover a relation between digraph roots and graph isomorphism which materializes in form of an isomorphism-completeness result: For a special class of digraphs defined through arc subdivisions, root finding is of the same complexity as deciding whether two digraphs are isomorphic. This may come as a surprise since all problems known to be of this complexity are more or less obviously isomorphism problems. In abridged form, the results from this chapter have already appeared in [**28**].

Acknowledgments

I want to thank my advisor Martin Aigner. I am grateful for his support and guidance but also for granting me the freedom to find and pursue my own way. Many thanks to everybody at the mathematics department of Freie Universität Berlin who made the past three years such an amazing time. Especially to Mark de Longueville, Carsten Schultz, Stefan Geschke, Elmar Vogt, and Dennis Epple for endless talks about math and a thousand other things; and to Andrea Hoffkamp, simply for being there.

I thank the people from the computer-science department of Freie Universität Berlin, in particular Laura Heinrich-Litan for being supportive and especially Günter Rote for sharing his knowledge and for always being interested. Throughout my time at Freie Universität Berlin I was a member of the European graduate school "Combinatorics, Geometry, and Computation" supported by the Deutsche Forschungsgemeinschaft and I want to thank the faculty of the program for providing an excellent work environment. Also thanks to Manuel Bodirsky for fruitful cooperation, and to Vincenzo Marra and Jörg Zimmermann for long discussions.

I am grateful to the hospitable people at KAM in Prague, foremost to Martin Loebl, for his advice and his interest in my work; and to Emo Welzl's group at ETH Zürich, in particular to Bernd Gärtner and Kaspar Fischer for a very productive time and to Ingo Schurr for sharing his office.

I also want to thank my teacher Arnold Schönhage, who influenced my mathematical thinking more than anybody else. Eventually, my thanks go to Sven Ehlert, Tim Bärmann, and Hans-Peter Jacobs for being very supportive; and to my girlfriend Petra Oyen for that and much more.

Berlin,
May 2004 Martin Kutz

Contents

CHAPTER 1

The Angel Problem

1. Angels, Kings, and Fools

Two players, the *angel* and the *devil*, play a game on an infinite chess board whose squares be indexed by pairs of integers. The angel is an actual "person" moving across the board like some chess piece, while his opponent does not live on the board but only manipulates it. In each move, the devil blocks an arbitrary square of the board such that this location may no longer be stepped upon by the angel. The angel in turn, flies in each move from his current position $(x, y) \in \mathbb{Z}^2$ to some unblocked square at distance at most k for some fixed integer k, i.e., to some position $(x', y') \neq (x, y)$ with $|x' - x|, |y' - y| \leq k$. Note that devil moves are not restricted to the angel's proximity or limited by any other distance bounds; he can pick squares at completely arbitrary locations.

The devil wins if he can stop the angel, that is, if he manages to get him in a position with all squares in the $(2k + 1) \times (2k + 1)$ area around him blocked. The angel wins simply if he succeeds to fly on forever. The open question is, whether for some sufficiently large integer k the angel with distance bound k, called the *k-angel*, can win this game.

First variants of this game were discussed by Martin Gardner [**17**], who names D. Silverman and R. Epstein as original inventors. Though his article deals mainly with finite configurations, i.e., the question whether a chess king (which is simply a 1-angel) can reach the boundary of a given rectangular board, he also asks for a strategy against a chess knight on an infinite board, possibly with a devil who gets to play more than just one block per move. In it's present form the angel game first appeared in Berlekamp, Conway, and Guy's classic [**8**] (Chapter 19). Amongst detailed analyzes of games with kings and other chess pieces on finite boards against devils with certain additional restrictions, the authors coin the names "angel" and "devil" for the two competitors and give a thorough proof that the chess king can be caught on a 33 by 33 board. Then Conway [**11**] focused entirely on the infinite angel game, trying to explain possible pitfalls with certain natural escape attempts and pointing out the hardness of the problem. Besides all variants, the central open question remains whether some angel of sufficient power can escape forever. In his overview article [**14**], Demaine cites it as a difficult unsolved problem of combinatorial game theory.

In this present work, we present modest advances on the current best known devil strategy. Therefore we introduce a slight reformulation of the original game, which allows us to focus on speed as the important parameter. In a further part, we treat a higher-dimensional analog of the angel game, showing that an angel of sufficiently large power can escape in 3D.

Catching the chess king. The only case for which the k-angel problem is solved is $k = 1$, the ordinary chess king. We like to sketch a winning strategy for the devil, which is motivated by the analysis in [**8**]. This shall get us some feel for the game and make us familiar with some basic principles that will turn up every now and then. The basic ideas are quite simple. Maybe the reader likes to stop reading here for a while and enjoy figuring out such a strategy on his own.

Assume the devil wants to prevent the king from crossing a certain horizontal line. With three squares above the king already blocked on that line, like in Figure 1, this is easily achieved. The devil simply answers a king's move a to the right by an extension of that triple block by a play at u. A further move to b is countered by v and likewise, any left movement to a' is blocked at u'. Pushing along in this simple fashion ensures that wherever the king goes, the three squares above him will always be blocked, making a crossing impossible.

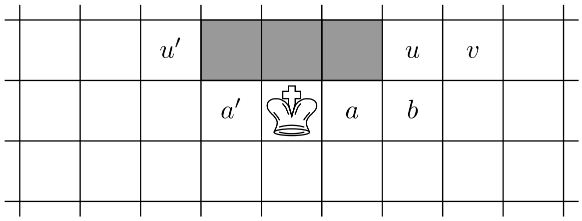


FIGURE 1. Pushing the chess king along a line.

It is not difficult to get the three initial blocks placed on a blank line when a king is just approaching. In the left drawing of Figure 2, the king is only five steps away from the desired line along the upper rim, where the devil has just played his first block. We claim that however the king now approaches that line, the devil will always manage to get his triple block in place.

If the king makes one step forward to a, the devil replies at u. After that the moves b' and b'' both lead directly to a triple block by the devil answers v' and v'', respectively. So we only have to consider a king's move to b. The devil plays at v, after which the king's moves c' and c'' are both blocked by v'. This leaves only a step to c, which is countered by w. Now the king's right-most option d can be blocked at x and the moves to d' or d'' again lead to a triple by v',

The second option for the king's first move is a in the right drawing of Figure 2. (A move to a' being symmetric to this case.) The devil plays at u. Against a step to b', the devil immediately forms a triple block by playing v'. The two moves to b and b'' lead to symmetric configurations, so we need only consider the remaining option b. The devil replies v, after which c is countered by w, and c' and c'' can both be blocked at v'.

So five preparation steps suffice for the devil to get his triple block in place against an approaching king. Figure 3 shows in a slightly non-proportional drawing how to turn the above wall-pushing argument into a successful devil strategy for catching the chess king. With his first 44 moves, the devil blocks some squares in the four corners of an imaginary box around

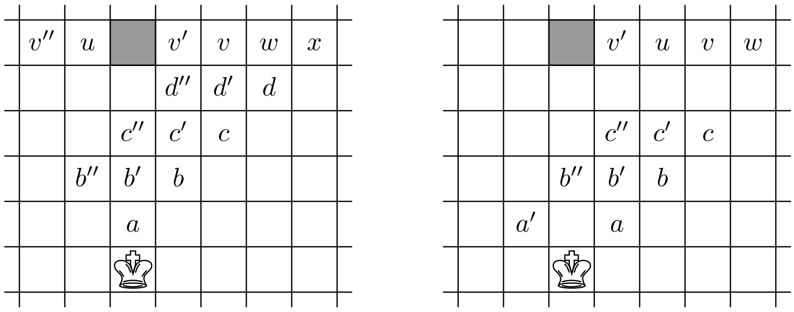


FIGURE 2. Getting the triple block in place.

the king. The box must be chosen large enough to ensure that during this preparation phase the king does not get too close to the boundary of that box. After that, the devil plays the above wall-pushing strategy along the dotted lines whenever the king approaches such a line. The four corners are there to ensure that the devil can never be forced to play on two fronts at the same time.

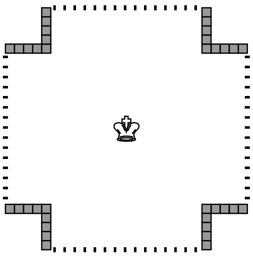


FIGURE 3. Catching the chess king.

We leave the argument at this informal state, hoping that the reader has grasped the idea. We are headed for a stronger result, which we shall then prove in full detail. As we already said, a deeper analysis of the chess king, very similar to the above discussion, can be found in Chapter 19 of [**8**].

The fool argument. The first general idea for an escape with a k-angel might be to run away in one direction. If the power k is large enough, shouldn't the angel somehow be able to go faster than the devil putting any serious obstacles in his way? Maybe the angel can simply run away in one direction. The answer is *no*! Conway defines a *k-fool* to be a k-angel who commits himself to strictly increasing his y-coordinate in every move. He shows that a fool of any power k can be caught [**11**]. The argument is simple, so we take the time to recapitulate it here.

The restriction on the y-movement implies that from a fixed position the k-fool can reach only squares within a cone of slope n. The devil sets out to build a long barrier at a far distance across this cone so that the fool, once he gets there, will stand in front of an impenetrable wall. Note that in order to block a k-fool effectively, we need a thick wall of k consecutive lines. We clearly cannot build such a solid wall across the whole width of the cone because already a 2-fool would arrive at the construction site much earlier than the devil could finish his work. Conway's trick is the following dynamic refinement strategy.

Say, our desired barrier shall be h units to the north. There the cone of possible future fool positions has width $2hk+1$, so that a complete wall of thickness k at that distance would consist of about $2hk^2$ squares. The devil begins filling this wall partly. With his first $h/(2k)$ moves, while the angel gets half way to the distant line, he blocks about 1 out of $4k^3$ squares there, distributing his moves evenly over the full width. Once the fool reaches the center line, the devil determines the new cone of potential fool positions, which by simple geometry, covers only half of the original wall. The devil then spreads his next $h/(4k)$ moves evenly on that segment of the construction site that can still be reached by the fool. See Figure 4.

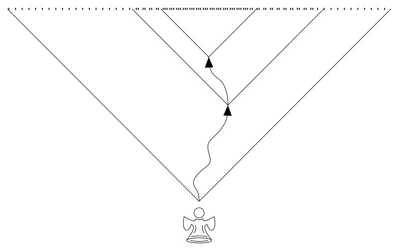


Figure 4. Catching a fool.

He obviously gets the same proportion of about 1 out of $4k^3$ squares blocked there until the fool has reduced his distance to the wall to $h/4$. If the initial distance h was chosen large enough, we can iterate this process often enough (about $4k^3$ times) to finish the relevant part of our barrier before the fool arrives.

This argument generalizes to non-strict fools, i.e., angels who are not allowed to make a step in negative y-direction, and it is not limited to one direction. There is also a radial variant where the angel never decreases his distance to the origin. The detailed arguments are given in [**11**].

Conway's fool counter already indicates that devising an escape strategy for some angel might be a very difficult task. By a simple dove-tailing argument this result can even be turned into the following surprising fact [**11**].

1. Theorem ("Blass-Conway diverting strategy"). *There is a strategy for the devil with the following property. For each point p of the plane and each distance d, no matter how the angel moves there will be two times $t_1 < t_2$ such that at time t_2 the angel will be d units nearer to p than at time t_1.*

This diverting strategy does not imply, however, that the angel must run in a wild zig-zag across the board. Concrete bounds on t_1 and t_2 are astronomical, so that the angel has plenty of time to comply with those requirements. But Theorem 1 can be used to immediately disqualify a variety of ad-hoc angel strategies, like refinements of the fool approach, that do not allow for sufficient freedom of movement in all directions. After all, Conway himself believes that some angel can escape. He awards \$100 for an escape strategy for an angel of some sufficiently high power k.

2. From Finite to Infinite Games

Before we go on to devise strategies for angel and devil, let us pause a while to discuss some fundamental aspects of infinite games in general. Such games may behave a little weird: It may be that neither player can force a win, i.e., there exist no winning strategies, even though the game does not allow for draws.

Formally, an infinite *game* is simply a subset $\mathcal{A}$ of $\mathbb{N}^{\mathbb{N}}$. A *play* is an infinite sequence $\tau = (x_0, y_0, x_1, y_1, \ldots)$ of natural numbers where Players 1 and 2 choose the x_i and y_i, respectively, in turns, and Player 1 wins iff $\tau \in \mathcal{A}$. A *strategy* is a mapping from all possible finite initial segments of a play to the next move, i.e., a mapping from the set $\mathbb{N}^*$ of finite words to $\mathbb{N}$ and it is a *winning strategy* if it wins against all possible opponent plays.

It is well-known that the axiom of choice allows the construction of games in which neither player has a winning strategy [**23**, Sec. 43]; but Martin [**32**] proved that for games that are Borel sets this cannot happen, such games are *determined*: one of the two players must have a winning strategy. This result covers essentially all games that can be defined in simple ways. Any "reasonable" game will be determined. And so is the angel-devil game. However, we do not need the full power of Martin's deep theorem. The following Lemma is an easy adaptation of an earlier, simpler result of Gale and Stewart [**16**]. I want to thank Stefan Geschke for intoducing me to these set-theoretic foundations of infinite games and for helpful discussions about the arguments in this section.

2. Lemma. *The angel-devil game is determined. That is, either the angel or the devil has a winning strategy.*

Proof. Assume the devil has no winning strategy. The angel can play as follows. In each turn he makes a move after which the devil does not have a winning strategy. By induction, such a move must always exist since otherwise the devil would have a winning strategy. The resulting angel strategy is obviously a winning strategy, simply because it allows the angel to play forever. □

Of course, one could define the above strategy for any given infinite game. The decisive point is that usually such a strategy does not automatically yield a win as is the case with the angel-devil game.

A further observation, which is useful when thinking about our game, is that in a sense it is infinite only from the point of the angel. If the devil wins, the game ends, by definition, after finitely many moves. So it seems that if the devil can win at all against the k-angel, there should exist some constant N_k such that the devil can catch the k-angel in at most that many moves. Equivalently, if some angel should be able to survive M moves, for any arbitrarily large number M that is fixed at the beginning of the game, then he should also be able to escape forever.

These seemingly obvious implications bear a subtlety. It could in principle be possible that the angel would have to choose his strategy dependent on the given M, so that he can in deed escape for M moves as required but will be caught a little later. If he had wanted to survive longer he might have had to choose a different strategy. Ultimately, there might not exist a strategy that works for all M at the same time. Seen from the devil's perspective this would mean that while he is sure to catch the angel after a finite number of moves, there might not be a universal bound on the time that is required to catch him. Fortunately, our concerns are needless.

3. Lemma. *If the devil has a winning strategy against some angel then there exists a bound N such that the devil can stop that angel in at most N moves. Conversely, if the angel can survive for any arbitrarily large, previously given number of steps then he can escape forever.*

Proof. Assume that the devil has a winning strategy. Consider the game tree of all possible plays under such a devil strategy σ. It has a bounded number of options at each angel node (no more than $(2k+1)^2$) and just one option at each devil node, namely the one prescribed by σ. The leaves are exactly those positions in which the angel cannot move anymore and thus has lost. This tree contains no infinite paths because such a path would directly give the angel an infinite sequence of moves, in contradiction to our assumption that σ is a winning strategy.

Since the degree of the tree is bounded and it contains no infinite paths, it is finite by König's lemma and therefore has finite depth, N, say. This means that the strategy σ allows no more than N moves before the angel is stuck, independent of how the angel plays.

The second statement is equivalent to the first. If the angel can escape as long as required by the beginning of the game, the devil cannot have a strategy that catches him after a fixed number of moves. Hence, the devil has no winning strategy at all, which means by Lemma 2 that the angel can escape forever. $\square$

3. The Need for Speed

There is pretty little known about even very weak angels. Already the destiny of the 2-angel is not settled and even more, it is unknown whether a chess knight, i.e., a piece that jumps in each move to one of the eight squares at Euclidean distance exactly $\sqrt{5}$, can be caught.

We do not have a solution for the 2-angel, either, but we make a first step in this direction by devising devil strategies against opponents whose power lies somewhere between that of a 1-angel and the strength of 2-angel. The improvement is rather modest but the new concepts we need to introduce in order to obtain them or even state them, reveal details of the game that seem to lie hidden with Conway's original angels.

Let us take a closer look at what happens when we upgrade the original chess king to a 2-angel. This is already a large step. The improvement is actually two-fold. Not only does the 2-angel move at twice the speed, any barriers must also be twice as thick to hold him back. In a sense, the 2-angel can be said to be 4 times stronger than the 1-angel. We focus on the first aspect: *speed*! We would like to suppress the ability to jump over obstacles as an undesired side effect. Define a *k-king* as a player who in each turn makes exactly k ordinary king's moves, while the devil still gets to place one block per turn. The point is that now every single king's move must be valid, the k-king cannot fly across obstacles.

If we want to use kings for the study of the angel problem, they should, in some qualitative sense at least, be equivalent to angels. Obviously, a k-angel is stronger than a k-king. An escape strategy for a king can be used for an angel of the same power as well. The converse is, of course, not true—not for trivial reasons at least—but we can show that if you can catch kings of arbitrary power k then you can also catch any angel. Before we come to this reduction, let us first remark on a subtlety in the above argument.

A k-king could in principle use a sequence of k steps to run a circle and return to his starting position, thereby simulating a pass between two consecutive devil moves. An angel is formally not allowed to pass. So our trivial transformation from above had a little flaw. The following basic no-return lemma by Conway [**11**] repairs this defect. It works for k-kings as well as for k-angels and will be needed once more later on.

4. LEMMA. *If the k-angel or k-king can escape then he can also escape without ever visiting any square twice; where in the case of the king we only consider the last step of a sequence of k steps between two devil moves.*

The restriction to the last step in a sequence of k king steps is natural because that final location is always the one that the devil sees when it's his turn. For the intermediate positions the following argument would not work.

PROOF OF LEMMA 4. We assume that we have a winning devil strategy σ against a non-returning k-angel or k-king and derive from that a winning strategy against the non-restricted versions. The idea is simple. When the angel/king revisits a location, the situation is always worse than at his first visit. The set of blocks has only grown. We turn this observation into a formal proof.

The devil plays according to σ until the angel/king lands on a square p he has already visited before. In this case the devil blocks an arbitrary square from the $(2k+1) \times (2k+1)$ area around p. Now he simply forgets all moves since his opponent first visited p and resumes the strategy σ from that position. The point is that his reply to the angel's/king's move to p has already been played when he had answered to p the first time, so that his intermediate move was really for free and he did not fall behind with σ.

We must be precise about what we mean by "forget." The intermediate moves, since the first play of the revisited square p, are really erased from the devil's memory. So when the angel steps on a square he had been before but the devil has forgotten about that move, he plays on without backtracking. Otherwise we would have to show how forward jumps in σ, that is, jumps to a location that has been visited in the forgotten future, should be treated consistently.

The result of the described devil play is, of course, that the angel/king cannot return more than $(2k+1)^2$ times to the same location because then he would not be able to leave it again due to its barred environment. Consequently, the derived strategy wins just as σ does. $\square$

Lemma 4 also shows that a little inaccuracy in our definition of the game is inconsequential. We have not said explicitly whether the devil should be allowed to block the square on which the angel currently sits. Since we may assume that the angel may never return to that position anyway, the devil will never need to make such a move.

We now establish the announced equivalence of angels and kings. Of course, the reduction from angels to kings requires an increase in speed.

5. Proposition. *If the k-angel can escape then so can the $99k^2$-king.*

Proof. We derive an escape strategy for the $99k^2$-king from an escape strategy for the k-angel. While the king plays against the "real" devil, we set up an additional, imaginary board with an imaginary k-angel, where we simulate the action on the king's board through appropriate transformations.

The king's board is partitioned into a regular grid of sidelength-$18k^2$ boxes. Likewise, the angel's board is segmented into blocks of sidelength k. The boxes of the two worlds are in one-to-one correspondence with each other, in the obvious fashion: the box containing the king's starting point corresponds to the angel's initial box and further all adjacencies are preserved. These partitions and the correspondences are fixed once and for all at the beginning of the game.

We play as follows. When the devil blocks some square in the king's world, we cross out an arbitrary empty square from the corresponding box in the angel's world or from one of the eight adjacent boxes there.

When it's the king's turn, we use our escape strategy for the angel to get a move in the imaginary world. This move is then translated into the king's plane by a movement of the king into the corresponding box there. For example, in Figure 5 the angel jumps from his current box into the next box to the north; then the king runs into the northern box in his world, too. The precise position within that box is completely independent of the angel's position in his box, however. It will depend on the following technical details.

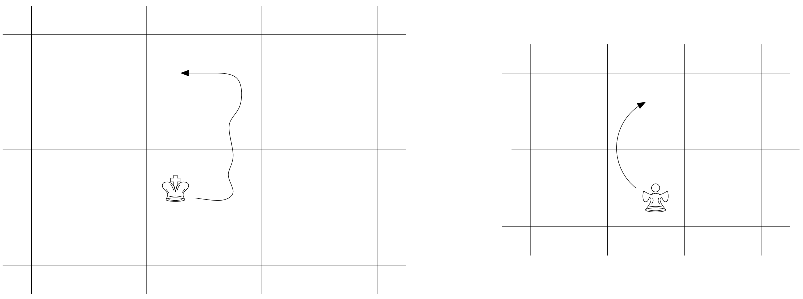


Figure 5. Simulating a king by an angel.

We have to describe precisely how the king should run and also prove that the required movement will always be possible. Observe that when the angel can jump into some box, the devil cannot have blocked all k^2 of its squares. From our simulation rule for devil moves, we conclude that the corresponding box in the king's world and also the eight surrounding boxes there contain less than $9k^2$ devil blocks each.

Knowing that a target box in the king's world contains less than $9k^2$ blocks, we can now find a route for the king. We introduce an invariant for king positions: the king only stops at locations from where the four lines into the four axis parallel directions within the current box are completely free. We maintain this invariant to always ensure a free passage for the king into his target box. Assume the king needs to go one box to the east. The density bound of $9k^2$ guarantees that in both orientations, vertical and horizontal, strictly more than half of the $18k^2$ lines are completely empty, in each box. This implies that there are at least two free long horizontal lines through both boxes. It is easy to see that from his good position with free roads in at least three directions (the fourth direction possibly blocked by the last devil move) the king can reach such a line in less than $9k^2 + 18k^2 = 27k^2$ steps, as shown in Figure 6. (The additional $9k^2$ moves result from a possible detour, which won't take longer, because there are at most $9k^2$ blocks in the whole square.) Then it takes no more than another $36k^2$ steps to reach a good position in the target box. If the king is headed for one of the four diagonally connected boxes, he first makes a stop-over in the horizontal or vertical direction and proceeds from there with at most another $36k^2$ steps. This gives a total of less than $99k^2$ steps.

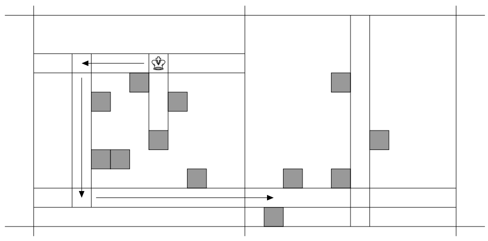


FIGURE 6. The king runs into a neighboring box.

Again a remark on passing. Since the king is forced to use up all his 99 moves in each turn, he might in principle get in troubles when he arrives at his destination too early. However, zugzwang is not really an issue here because the king finds enough empty squares along the side of his road to waste arbitrary numbers of moves by running in little circles. □

We emphasize again that the quantitative proportion of the above reduction is not our main concern. The purpose of Proposition 5 is only to establish the qualitative equivalence between angels and kings, as a legitimation to use kings as a tool to attack the angel problem.

Preparing fences. Let us have a closer look at the devil strategy against the 1-king from the beginning. It seems we wasted some potential there. After the preparation of the corners, the devil simply sits and waits for the king to arrive at one of the four sides. Couldn't he perhaps use this time for some further preparations so that he can catch a faster king, a 2-king, maybe.

The basic idea for the king counter was our dynamic-wall argument, where we had the king pushing along a line without ever letting him break through. Can we extend this method to the 2-king? Since the 2-king makes two steps for each devil move, it would suffice to have every second square along the desired frontier already in place. Starting from the initial position in Figure 7 with only two additional squares blocked, the devil can push along with the 2-king by answering the double move a_1, a_2 at u, then b_1, b_2 at v, and so on.

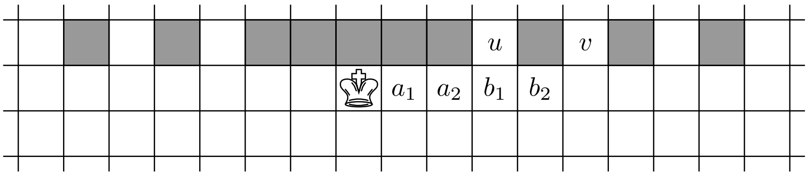


FIGURE 7. A wall against the 2-king.

How long would it take the devil to prepare such a density-1/2 wall against the 2-king? Since he needs to block 1 square out of 2, he can set up such a wall at an absolute speed of 2, which is exactly the speed of the 2-king. In other words, the devil can build such fences against the 2-king at the same speed the 2-king runs. For example, a 2-king who sits at the bottom of a square box of sidelength R with solid walls to the east, west, and south but completely open to the north is lost. We just learnt that the devil can build a fence of density 1/2 across that open gate in the north, in just the time it takes the 2-king to get there. Hence, the 2-king can never leave that box.

Can we extend these ideas to encircle the 2-king completely? The answer is *yes*—almost. We shall present successful devil strategies against any king of speed $2-\varepsilon$ for any fixed real $\varepsilon > 0$. First, of course, we have to say what such a statement shall mean. We need a definition of fractional, or even irrational speed.

Real kings. What is a 3/2-king? On average he should get to make three king's steps for 2 devil steps, which we could realize by a move sequence like $KKKDDKKKDD\ldots$, which shall mean that the king makes 3 steps, then the devil blocks 2 squares, and so on. However, such a concept would depend on the actual representation of a rational number. The 6/4-king would get a different sequence. We could get around this by demanding reduced fractions but then a 1001/8-king would behave completely different from a 1000/8-king, who should simply be the 125-king. What's worse, the grouping of devil moves can be lethal for the king. For example, the eight consecutive devil moves in the sequence $K^{1001}D^8K^{1001}D^8\ldots$ could be used to encircle the king completely, even though his average speed would be greater than 125.

What we want are move sequences that approximate a given speed $\alpha \in \mathbb{R}^+$ as fair as possible, avoiding unnecessarily large chunks of moves for either side. The sequence $(u_n)_{n\in\mathbb{N}}$ defined by [**2**]

$$(1)\qquad u_n = \lfloor (n+1)\gamma + \phi \rfloor - \lfloor n\gamma + \phi \rfloor \in \{0,1\} \qquad \text{with } \gamma = \frac{\alpha}{\alpha+1} \in (0,1)$$

and some constant offset $\phi \in \mathbb{R}$ shows this behavior—if we interpret 1's in the sequence as the king's and 0's as the devil's moves.

This sequence (u_n) is easy to understand; it simply compares consecutive elements of the arithmetic progression $(n\gamma + \phi)$. Whenever there lies an integer between the nth and the $(n+1)$st element of $(n\gamma + \phi)$, we have $u_n = 1$, otherwise, when the two elements fall in a common integer gap, (1) evaluates to $u_n = 0$. We conclude that the frequency of 1's in (u_n) is γ, hence the frequency of 0's is $1-\gamma$ and we get (cf. [**2**])

$$\lim_{n\to\infty} \frac{|\{i \le n : u_i = 1\}|}{|\{i \le n : u_i = 0\}|} = \frac{\gamma}{1-\gamma} = \alpha. \tag{2}$$

The sequences (u_n) are called *Sturmian sequences* if α is irrational, and they are well-studied. See [**2**] for a broad treatment and for historic references.

6. Definition. For $\alpha \in \mathbb{R}^+$ we define the *α-king* to be a king whose move sequence is given by (1) with $\phi = 0$. This means that in the n-th time step the king moves by one square if $u_n = 1$ and the devil gets to block a new square if $u_n = 0$.

The choice of the offset ϕ looks arbitrary. For a natural definition it would be desirable that the chances of the α-king in the game do not depend on this parameter. And in fact, they don't.

7. Lemma. *Any two kings with move sequences generated by* (1) *with the same speed parameter α but different ϕ's either can both escape or can both be caught.*

Proof. Let (u_n) be a sequence defined by (1) with offset ϕ and let (u'_n) be another sequence defined with some other offset ϕ', both with the same α, though. We distinguish rational and irrational α.

For $\alpha \in \mathbb{Q}$ we write $\gamma = p/q$ in reduced form. Obviously both sequences are periodic with period q. All we have to do is to align them in the right way. Partition the unit interval into congruent half open intervals of length $1/q$ and let $r \in \{0, \dots, q-1\}$ be the index of the residue class with

$$\phi \in \left[m + \frac{r}{q}, m + \frac{r+1}{q}\right) \qquad \text{for some } m \in \mathbb{Z}.$$

We look for some index j, where the sequence $(jp/q + \phi')$ hits the same residue class; i.e.,

$$j\,\frac{p}{q} + \phi' \in \left[m' + \frac{r}{q}, m' + \frac{r+1}{q}\right) \qquad \text{for some } m' \in \mathbb{Z},$$

which clearly exists because p and q are coprime. It is easy to see that

$$u_n = u'_{n+j} \quad \text{for all } n \in \mathbb{N}.$$

If the devil has a winning strategy on the move sequence (u_n), he can therefore also win on (u'_n) by simply waiting j time steps and then starting to play according to the strategy for (u_n). By exchanging (u_n) and (u'_n) we get the converse simulation.

For irrational α the sequences (u_n) and (u'_n) are non-periodic. We use a deeper result from the theory of such Sturmian sequences: The set of contiguous subwords of the sequence (u_n) depends only on α and not on the offset ϕ [**2**]. (Even more, there are exactly $n+1$ different subwords of length n and each of them occurs infinitely often in u_n.) Therefore, any initial segment of (u_n) can also be found somewhere in (u'_n).

Since Lemma 3 tells us that if the devil can win on the move sequence (u_n), he can do so in a bounded number of N steps, say, he can use a strategy on (u_n) to win on (u'_n) by simply waiting until a copy of the N-prefix of (u_n) starts in (u'_n) and then pursuing this strategy. $\square$

For $\alpha \in \mathbb{N}$, the above definition of an α-king obviously coincides with the previous one that was restricted to integral speed. For $\alpha = k \in \mathbb{N}^+$ the defining sequence (1) produces exactly k 1's between any two consecutive 0's, just as expected. It is also clear that our notion of an α-king fulfills our wish for fairness, large chunks of devil moves cannot occur. One easily checks that for $\alpha \geq 1$, the devil never gets to block two squares at a time. On the other hand, we can guarantee that not only in the long run but also locally, the devil always gets his share of moves.

8. Definition. A 0/1-sequence is (s,t)-*bounded*, $s,t \in \mathbb{N}^+$, if every contiguous subword that contains strictly more than s occurrences of 1's contains at least t occurrences of D. We call a king with a given move sequence (s,t)-*bounded* if the sequence is (s,t)-bounded. (Where we interpret 1's as king's moves and 0's as devil moves.)

9. Lemma. *An α-king, $\alpha \in \mathbb{R}^+$, is (s,t)-bounded for every pair $s,t \in \mathbb{N}^+$ with $\alpha \leq s/t$.*

The "strictly" in the definition appears for a technical reason; it does not mean that we get only t devil moves per $s+1$ king moves on average. Namely, starting from any 1 in the sequence, we count 0's until we reach the $(s+1)$st 1. By then we have passed at least t 0's. When we read on until the $(2s+1)$st 1 shows up, we are sure to have counted at least $2t$ 0's. And so on. Before the $(rs+1)$st 1 appears, we are guaranteed to read at least rt many 0's.

Proof of Lemma 9. Assume we have $s+1$ many 1's between two positions a and b (inclusively) in the sequence (u_n). Telescoping (1) yields

$$s+1 \leq \sum_{a \leq i \leq b} u_i = \lfloor (b+1)\gamma + \phi \rfloor - \lfloor a\gamma + \phi \rfloor < (b-a+1)\gamma + 1,$$

where the terminal 1 accounts for the error that might result from the deletion of the floors. For the number of 0's in this interval we thus get

$$b-a+1-(s+1) > \frac{s}{\gamma} - (s+1) = \frac{s-\alpha}{\alpha} \geq t-1. \qquad \square$$

4. Catching a $(2-\varepsilon)$-King

In this section we develop a devil strategy to catch all kings of speed less than 2. The following main theorem emerged from joint work with Attila Pór.

10. Theorem. *The devil can catch any α-king with $\alpha < 2$.*

Have a look at Figure 7 again, where the devil pushed a 2-king along a line of density 1/2. With every second square already in place, the 2-king could never break through. We generalize this idea to kings of arbitrary speed.

11. Definition. An *infinite* (s,t)*-fence* is an infinite horizontal or vertical strip in the plane with some squares blocked such that when an (s,t)-bounded king enters the strip from one side, the devil can play in a way that prevents the king from leaving it on the other side. Formally, such a fence is just a map $F\colon \mathbb{Z}\times[1\,..\,w]\to\{0,1\}$, where $F^{-1}(1)$ is the set of blocked squares. The integer w is called the *width* of F.

We call such a fence *periodic* if there exists some integer λ such that $F(x,y)=F(x+\lambda,y)$ for all $x\in\mathbb{Z}$. Call the minimal such λ the *period* of F. In this case we also define the *density* of the fence, as the ratio

$$\frac{1}{\lambda}\left|\left\{(x,y)\mid 1\le x\le\lambda, 1\le y\le w, F(x,y)=1\right\}\right|.$$

Note that density is measured with respect to length, not area. Width is not the crucial quantity, it appears for merely technical reasons.

12. Lemma. *Against an (s,t)-bounded king, $s/t\le 2$, there exists a periodic infinite fence of density $1-t/s$ and width $10s+1$.*

Proof. We provide a periodic map $F\colon\mathbb{Z}\times[1\,..\,10s+1]\to\{0,1\}$ with the desired properties. Let F be everywhere zero except for those points (x,y) with

$$0\le x \bmod s < s-t \quad\text{and}\quad y=5s+1.$$

In other words, we group the central horizontal line $y=12+1$ into segments of s squares and place $s-t$ blocks in each segment. See Figure 8. The density of this pattern is obviously the claimed $(s-t)/s$.

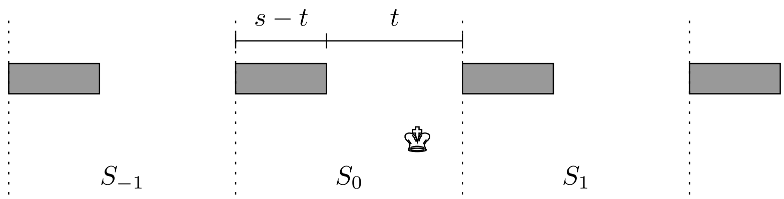


Figure 8. An infinite (s,t)-fence.

We now show how the devil keeps the king from crossing F by making sure that he can never step on the central line. By symmetry we may assume that the king enters the strip from the bottom.

Like in the case of the 2-king, we make sure that in the proximity of the king the central line is always filled completely. Precisely, if the segment S_0 above the king's current position has already been filled completely and the one to the left and right, S_{-1} and S_1, too, then the devil acts as follows. As soon as the king steps into the area below the segment S_{-1} to the left, the devil uses his next t moves to fill up the segment S_{-2}, further to the left. By (s,t)-boundedness, this is finished before the king gets to play his $(s+1)$st move (counting the move that entered S_{-1} as the first). Hence, by that time the king must be somewhere below the segments S_{-1} and S_0, and because S_{-2} is now filled we are in the situation as before: the three segments directly above the king are blocked. If the king started running to the right, the devil would have filled the segment S_2, of course. The devil can iterate this recipe forever, never letting the king step on the central line.

To obtain the above configuration, we reuse the procedure for the 1-king from section 1, where we managed to get a block of three consecutive squares in the king's way. Interpreting a whole segment S_i as a single square in which we must play t moves, we immediately see that 5 such meta moves suffice to get three segments prepared. Since any sequence of t devil moves yields no more than s king's moves, this gives a total of $5s$ approach moves, which is just the width of the strip below the central line. $\square$

The devil cannot build infinite structures in finite time. Infinite fences serve as a mere theoretical concept, which is easier to handle than finite fences, whose existence can be easily derived from the infinite ones.

13. Definition. A *finite (s,t)-fence* is a rectangular box of size $\ell \times w$ in the plane with some squares blocked, such that when an (s,t)-bounded king enters through one of the length-ℓ sides he can only leave through that side again and such that all squares along the two length-w sides blocked. Formally, such a fence is a map $F\colon [1\,..\,\ell] \times [1\,..\,w] \to \{0,1\}$, where $F^{-1}(1)$ is the set of blocked squares. The integers ℓ and w are called the *length* respectively *width* of F. The *density* of the fence is the ratio

$$\frac{1}{\ell}\left|\left\{(x,y) \mid 1 \le x \le \ell, 1 \le y \le w, F(x,y) = 1\right\}\right|.$$

The following lemma provides the trivial transformation of an infinite fence into a finite fence.

14. Lemma. *If there exists a periodic infinite (s,t)-fence of density σ then there exist finite (s,t)-fences of the same width and of density no more than*

$$\sigma + \frac{2w}{\ell}$$

for any length $\ell \ge 1$.

Proof. The basic idea is obv: we cut a length-ℓ segment out of the infinite fence S. We only face a little inconvenience. Unless the desired length ℓ is a multiple of the period λ of S, our chosen segment might contain more than the average density due to local inhomogeneities. This problem is easily overcome by looking at a sequence of λ aligned length-ℓ segments of S. Since their total length is an exact multiple of λ, the total mass in all of them is exactly $\sigma\lambda\ell$. Now at least one of those segments contains no more than the average $\sigma\ell$ blocks. To turn this segment into a finite fence, we have to fill the length-w sides up completely, which costs the additional $2w$ squares. $\square$

Lemma 12 provides us with an infinite fence of density $1 - t/s$, which is strictly smaller than $1/2$ for an α-king with $\alpha < 2$. This does not seem to suffice to catch any such king, yet, but for $\alpha < 9/8$ we already get a devil win as follows. By Lemma 9 this speed bound grants us (s,t)-boundedness with $s/t < 9/8$. So there exist infinite fences of density $\sigma \leq 1 - t/s < 1/9$. Choosing sufficiently long finite subfences of such an infinite strip, we can make the additional cost of $2w/\ell$ in Lemma 14 arbitrarily small, so that it gets absorbed by the small gap between σ and $1/9$. Altogether there exist finite (s,t)-fences of density at most $1/9$. This is all we need against our α-king. We simply build a square box of four such fences around him; in such a way that these fences touch but don't overlap. For a sidelength of ℓ this takes $4\ell/9$ devil moves, which in turn yield less than $9/8 \cdot 4\ell/9 = \ell/2$ king's moves. That means, all four fences will be finished by the time the king reaches the boundary of the box. Hence, he will be caught.

In prospect of the proof of Theorem 10, we forgo more formal details of this argument because the 9/8-king is covered by that result. The strategy behind Theorem 10 starts of just like the 9/8 case, by obtaining some fence of density below 1/2. The trick then is to assemble many such fences into a huge new fence of slightly smaller density. Iterating this process, we will eventually produce fences of arbitrarily small density. The key tool is the following lemma, whose proof describes this construction. (Observe that the bound $(s/t)\sigma^2$ in this lemma is strictly less than σ, which means that the density is really decreased.)

15. Lemma. *If there exist finite (s,t)-fences, $s/t \leq 2$, of any length above some value ℓ_0, all of the same width w and with density bounded by a common $\sigma < 1/2$, then there also exists a periodic infinite (s,t)-fence with density below*

$$\frac{s}{t}\,\sigma^2.$$

Proof. The basic idea is to assemble infinitely many identical vertical finite density-σ fences to a wide horizontal fence of the desired density. As the length ℓ of those finite fences we pick any multiple of s larger than ℓ_0 and w. (Actually ℓ_0 should be much bigger than w anyway, but let us demand $\ell \geq w$ here for the sake of rigor.) As the distance between those fences we choose

$$m := \left\lfloor \frac{t\ell}{s\sigma} \right\rfloor \geq \ell.$$

Let the width of the infinite fence L we want to construct be 7ℓ, i.e.,

$$L\colon \mathbb{Z} \times [1 \mathinner{..} 7\ell] \to \{0,1\}.$$

Figure 9 shows how the vertical fences of length ℓ and width w are placed in the central ℓ-strip of L. Precisely, the region

$$\big[\nu(w+m)+1 \mathinner{..} \nu(w+m)+w\big] \times \big[3\ell+1 \mathinner{..} 4\ell\big]$$

forms a fence for each $\nu \in \mathbb{Z}$.

Before we start to play on L, let us compute its density. The period is $w+m$ and each segment of this length receives no more than $\sigma\ell$ blocks, so we can bound L's density by

$$\frac{\ell\sigma}{m+w} \leq \frac{\ell\sigma}{\left\lfloor \frac{t\ell}{s\sigma} \right\rfloor + 1} \leq \frac{s}{t}\sigma^2,$$

which is what we claimed.

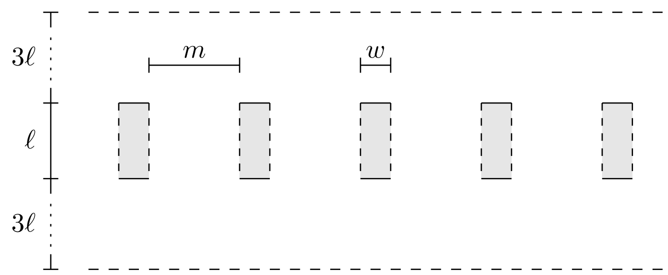


FIGURE 9. Assembling many finite vertical fences into one big infinite horizontal fence.

We turn to the more difficult part: showing that L is indeed an (s,t)-fence. Assume the king enters L from the south, so we have to keep him from reaching the upper border. The basic idea is to build a horizontal fence between the upper ends of two vertical fences whenever the king runs north between them. Such a horizontal fence will be of length m to make it fit nicely in the gap. It will be placed in the rectangle

$$\big[\nu(w+m)+w+1 \mathinner{..} (\nu+1)(w+m)-1\big] \times \big[4\ell \mathinner{..} 4\ell+w-1\big]$$

for the respective $\nu \in \mathbb{Z}$. This arrangement is displayed in Figure 10 (the shaded area between the fences will soon be addressed). Note the vertical one-point overlap with the vertical fences on line 4ℓ. To avoid confusion: Those horizontal fences will be created *dynamically* by the devil when necessary, they are not part of the original strip L when the king enters.

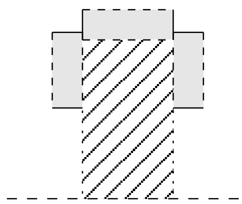


FIGURE 10. A horizontal fence between two vertical fences and the shaded slot between them.

We now describe the essential aspect of the devil strategy starting from a standard situation, postponing the matter how to reach that situation for later. Therefore we give the shaded area between two vertical fences and below the (potential) horizontal fence a name: call such a rectangle of the form

$$\big[\nu(w+m)+w+1 \mathinner{..} (\nu+1)(w+m)-1\big] \times \big[1 \mathinner{..} 4\ell-1\big], \quad \nu \in \mathbb{Z},$$

a *slot*. We say that the king is in *standard position* if he is located within a slot whose upper border is already closed with a horizontal fence or he sits between two such blocked slots, perhaps within the vertical fence between them.

Let us assume the king is in standard position. We claim that if he leaves the slot then the devil can force him into standard position again by playing as follows. When the king enters one of the three surrounding fences, he follows the strategy of that respective fence to make sure that the king does not break through to the other side of that fence. Note that we use the fact here that those fences do not overlap so that the devil is not forced to play in two fences simultaneously. Since there are no gaps where the three fences touch, this play guarantees that the king cannot leave the current slot above line 3ℓ without rebouncing from the fences.

If the king leaves the slot that way below, to the left, say, the devil starts constructing the horizontal fence across the slot to the left. This takes no more than

$$m\sigma = \left\lfloor \frac{t\ell}{s\sigma} \right\rfloor \sigma \leq \frac{t\ell}{s} \tag{3}$$

devil moves. During this time the devil completely ignores the king play. In particular, he does *not* respond to the possible king's crossing of any fences, thus rendering them ineffective. Where can the king get while the devil is off at work? By (s,t)-boundedness the king gets no more than s steps per t devil moves. Counting the step out of the slot as the king's first move, we reckon that until the $(\ell+1)$st king move, the devil has made at least $t\ell/s$ moves, i.e., the $(\ell+1)$st king move comes after that many devil moves. Since this figure is just what we have computed in (3), the king gets no more than ℓ moves before the new fence is finished. Note that we used that ℓ is a multiple of s when we applied (s,t)-boundedness.

A look back at Figure 9 reveals how far the king can have run in ℓ moves. Since the first move lead strictly below the $(3\ell+1)$st line, he cannot have reached the $(4\ell+1)$st line, where the horizontal fences start. Neither can he have crossed completely the slot to the left, nor the original slot because those areas are each $m \geq \ell$ points wide. Consequently, the king ended up somewhere inside the old slot or the new slot, which is fine because both now have a fence above them, or he sits somewhere in between them. That means he is in standard position again.

It remains to show how to reach a standard position from the initial situation when the king enters the unmodified strip on line 1. The argument is again very similar to the respective part of the strategy against the 1-king. Here it is actually even simpler because we need only one horizontal fence instead of a triple block. However, the notion of standard position requires a little extra attention.

Call the slot in which the king's first position lies S_0, in case he enters just between two slots, just pick any of them; and label the four neighboring slots correspondingly S_{-2}, S_{-1}, S_1, and S_2, from left to right. The devil first constructs the horizontal fence above slot S_0. We already know from the previous computation that such an endeavor grants the king at most ℓ steps. So afterwards, the king sits in one of the slots S_{-1}, S_0, S_1 or in one of the two gaps between them. Inside S_0 he is already in standard position. If he sits in a gap, the one between S_0 and S_1, say, then the devil builds a fence above S_1, after which the king can only be in S_0, S_1 or the gap where he already was before. So we have reached standard position. It remains to consider a king in slot S_{-1} or S_1 with S_0 blocked; in S_1, say, by symmetry. Then the devil builds the fence above S_2, squeezing the king between S_0 and S_1. Blocking S_1 with the next m moves, then leads into standard position.

Altogether, the total number of devil moves is in no case larger than $3m$, so that by the time we attain standard position, the king will not have reached the $(3\ell+1)$st line, yet. □

Proof of Theorem 10. Pick positive integers s and t with $\alpha \leq s/t < 2$, so that the α-king is (s,t)-bounded by Lemma 9. Then Lemma 12 provides us with an infinite periodic (s,t)-fence of density $\sigma < 1/2$.

For an application of Lemma 15 we have to fix a suitable lower bound ℓ_0 on the length of the finite fences we allow for the construction of the new fence. Therefore we write $s/t = 2/(1+\delta)$ with some, possible very small $\delta > 0$ and choose ℓ_0 large enough to ensure that the densitiy of the finite fences longer than ℓ_0, as obtained by Lemma 14, is bounded through

$$\sigma + \frac{2w}{\ell_0} \leq \sqrt{1+\delta}\,\sigma.$$

Now Lemma 15 gives us an infinite (s,t)-fence of density

$$\sigma' \leq \frac{s}{t}\left(\sigma + \frac{2w}{\ell_0}\right)^2 \leq \frac{2}{1+\delta}\big(\sqrt{1+\delta}\,\sigma\big)^2 = 2\sigma^2.$$

Repeated application of this procedure yields a sequence $\sigma_0, \sigma_1, \sigma_2, \ldots$ with $\sigma_n \leq 2\sigma_{n-1}^2$ and $\sigma_0 < 1/2$. The resulting bound

$$\sigma_n \leq \tfrac{1}{2}(2\sigma_0)^{2^n}$$

is easily verified, so that we see that the sequence (σ_n) converges to 0.

In a game against the α-king, the devil can now arrange four finite (s,t)-fences of density smaller than $1/16$, say, along the four sides of a huge square around the king. With α bounded by 2, the devil builds such fences more than 8 times faster than the α-king runs and thus finishes them before the king reaches any of them. Hence, the king will never leave that big box. Note that the fences have to be arranged in a non-overlapping way to ensure that the devil can play in each of them independently. And maybe we should also remark that the king cannot run around in his cage forever. After some time, when the fences are filled to the rim with devil moves, the devil simply starts flooding the central region with blocks until the king eventually gets stuck. □

As the proof has shown, the 2 in Theorem 10 maxes out the potential of our fences. We have already indicated in our discussion of the 2-king on page 10 that a speed of 2 can be considered “fair” with respect to fence building. If one used fences in the described way against faster kings, their construction would be more expensive than the gain through the resulting king's detour. This can perhpas be seen as some very weak indication that a $(2+\varepsilon)$-king cannot be caught anymore, but fences could, of course, just be one technical tool, without any deeper meaning for the game.

Anyway, since Conway's article of 1996, there has apparently not been any progress on the angel problem. Maybe Theorem 10 stimulates interest in this game again, since the concept of α-kings allows for arbitrarily small improvements in devil strategies. Perhaps we can learn something new about our two antagonists from the voyages of very slow kings. Join the game!

5. An Escape into Space

So we do not have any escape strategy for any k-angel in the plane. Maybe we can obtain some positive result in higher dimensions, where an escape should be potentially easier. And in fact we can. 3D-angels live in a 3-dimensional world of cubes, indexed by coordinates in $\mathbb{Z}^3$. Just like in the plane, in every move the k-angel jumps from his current position (x, y, z) to some other cube (x', y', z') with $|x' - x|, |y' - y|, |z' - z| \leq k$, and in turn, the devil blocks some cube of his choice. We prove the following.

16. Theorem. *On the three-dimensional board, the 13-angel can escape forever.*

The problem at hand has only been mentioned once in the literature, also in [**8**], where the authors actually report to know escape strategies for angels in higher dimensions. However, a proof has apparently never been published.

Theorem 16 should be seen in the proper light. It is not a breakthrough on the way towards a solution of the two-dimensional case but rather confirms that the original question by Conway, Berlekamp, and Guy addresses the right problem. Moreover, it will become clear that our solution, a density-sensitive path-search method based on a hierarchical space partition, will *not* carry over to the two-dimensional game—not without major modifications, at least. So we not only provide a first constructive escape strategy for a variant of the angel problem but also want to point out the intrinsic obstacles for similar strategies in dimension two, to further emphasize the hardness of the original angel problem.

The box hierarchy. Our escape strategy divides the world into an infinite hierarchy of larger and larger boxes. The angel will have to make sure that on each level, his current box contains not too many devil blocks. This shall then guarantee his free travel.

A remark on terminology. Our usage of the word "cube" might get a little confusing when we speak about our hierarchy, since higher-level boxes will themselves be cubes—of cubes of cubes of cubes, etc. We shall use the expression *elementary cube* to emphasize that we mean the basic locations of the board, while the term *box* be reserved for collections of such objects. With other expressions the intended meaning should in general be clear from the context.

On the first level, the world is regularly partitioned into cubes of sidelength 13, such that the origin $0 \in \mathbb{Z}^3$, where the angel starts, lies at the very center of one of these boxes. Formally, the first level H_1 is the collection of all boxes

$$H_1^{(u,v,w)} := \{ (x, y, z) \in \mathbb{Z}^3 \mid \begin{array}{rcccl} 13u - 6 & \leq & x & \leq & 13u + 6, \\ 13v - 6 & \leq & y & \leq & 13v + 6, \\ 13w - 6 & \leq & z & \leq & 13w + 6 \end{array} \},$$

with $u, v, w \in \mathbb{Z}$, where we reference elementary cubes of the world via their coordinates $(x, y, z) \in \mathbb{Z}^3$.

The sidelength 13 corresponds to the power of the 13-angel. From level 2 on, sidelengths grow by a factor of 29 per step, where there is no deeper reason for the choice of this particular value except that it makes the forthcoming computations work. On each level we again demand that the origin lie at the very center of the one box that contains it. Technically, for $j \geq 2$ the jth level H_j of our hierarchy is the collection of all boxes

$$\begin{aligned} H_j^{(u,v,w)} := \big\{ H_{j-1}^{(a,b,c)} \mid\ & 29u - 14 \leq a \leq 29u + 14, \\ & 29v - 14 \leq b \leq 29v + 14, \\ & 29w - 14 \leq c \leq 29w + 14 \big\}, \end{aligned}$$

with $u, v, w \in \mathbb{Z}$.

So any box on level $j \geq 2$ contains 29^3 boxes on level $j-1$ and the whole hierarchy is symmetric to the origin. Note that formally the elements of a higher-level box are again boxes, which is what we want. But with a certain laxness we shall also consider a level-j box simply as the set of the $(13 \cdot 29^{j-1})^3$ elementary cubes that lie inside it. In this vein we define the *level-j box of a cube $a \in \mathbb{Z}^3$* to be the unique box in H_j that "contains" the elementary cube a and denote it by

$$Q_j(a).$$

Further we define a mass function μ for all boxes A on all levels of our hierarchy, letting

$$\mu(A)$$

count the number of elementary cubes inside A that have already been blocked.

Clear roads ahead. Globally, the angel's route through our hierarchy of boxes will be guided by simple mass constraints, in a quite elegant way. The basic step, the transition between two adjacent boxes, however, requires some dirty work. We need to introduce a few technical notions to ensure that locally the angel does not get stuck in unfortunate arrangements of blocks. The ideas are similar to the invariant from the proof of Proposition 5 from page 8.

17. Definition. Let E be a quadratic grid of 29×29 cubes with some cubes marked *forbidden*. We say that a cube a of E *lies clear in E* if

- no more than 12 of the $29^2 = 841$ cubes in E are forbidden,
- a lies in the central 13 by 13 square of E,[1] and
- the two axis-parallel lines through a in E contain no forbidden points.

See the left-hand side of Figure 11.

Let C be a cubic grid of $29 \times 29 \times 29$ cubes with some cubes marked *forbidden*. We say that a cube a of C *lies clear in C* if

- no more than 333 of the the $29^3 = 24{,}389$ cubes in C are forbidden and
- a lies clear in one of the three axis-parallel 29×29 planes through a in C.

See the cube in Figure 11.

[1] The occurrence of the number 13 here is coincidental. This is a "different" 13 than the one from Theorem 16.

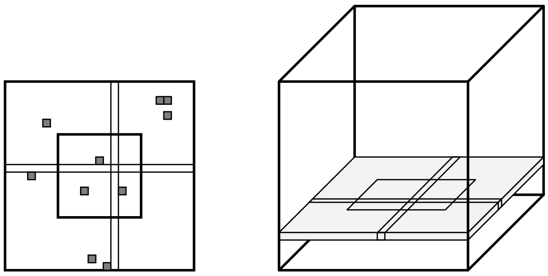


FIGURE 11. Clear positions.

The idea behind the above definitions is, as we said before, to guarantee free navigation from a clear cube within a sidelength-29 box to somewhere outside this box. A cube that lies clear will have enough free space around it to guarantee an easy route out. The forbidden cubes may, of course, not be used for travel. We do not speak of blocked cubes in Definition 17 because the little cubes will usually themselves be boxes of smaller cubes. But forbidden cubes will almost be blocked, meaning that their mass exceeds a certain threshold.

For *paths* through such boxes we allow axis parallel steps of unit distance only. That is, a single step of a path is a change of ± 1 in just one coordinate—in contrast to basic angel moves. This restriction is due to the hierarchical structure of our argument. We will be able to travel between two little cubes inside the big cube in Definition 17 only if these cubes share a face which may be used for a transition on the next lower level.

From a purist's point of view, the grids E and A of Definition 17 could, of course, just be called grid graphs, with "cubes" replaced by "vertices." Then a path would just be a paths in the graph theoretic sense and the following lemmas are in fact just statements about such grid graphs. However, we like to keep with our view of cubes and boxes, hoping that this does not cause any confusion.

The following lemma about planes only serves as a tool for the three-dimensional case. Our actual interest will be in paths through boxes.

18. LEMMA. *Let q be a cube lying clear in a 29×29 grid E. Then at least $763 = 29^2 - 78$ cubes of E are reachable from q in at most 40 steps each.*

PROOF. Any cube on the two lines through a is by assumption reachable directly through that respective line. For every other point $p \in E$ we consider the two potential paths that run parallel to the axes with exactly one turn. A cube p may not be reachable on either of these two paths for two reasons: both paths are blocked or p is a forbidden cube itself. Since by the special choice of our paths, a single pair of forbidden cubes covers at most one cube of E, the first situation can happen for at most $\binom{12}{2} = 66$ cubes, the second, by definition, for at most 12; which makes 78 inaccessible places altogether. One easily computes that any of the remaining $29^2 - 78 = 763$ cubes is reachable in at most 40 steps since the distance from any location in the central region to any side of E is at most 20. $\square$

19. Lemma. *Let q be a cube lying clear in a $29 \times 29 \times 29$ box grid C and let D be another $29 \times 29 \times 29$ box aligned with C along one face of C, also with no more than 333 points marked forbidden. Then there exists a cube r lying clear in D such that there is a path of length at most 165 from q to r, which after the first 96 steps uses no more cubes in C.*

Proof. Let E denote the plane within C in which p lies clear as required by Definition 17. The basic idea for the path construction is to pick a suitable plane F in D, which will contain the target point r, and then to find many disjoint paths from E to F not all of which can be blocked by forbidden cubes.

Observe that by the pigeon-hole principle, among the 29 axis-parallel planes in D that lie parallel to that face of D which borders on C, at least one contains no more than 12 forbidden cubes ($29 \cdot 13 = 377 > 333$). Choose F to be such a plane. For both dimensions of F, at most 12 of the 13 axis-parallel lines passing through the central 13×13 region of F are blocked by forbidden cubes, which leaves at least one clear line in each direction. We choose b as the intersection of two such lines, which makes it lie clear in D. We now distinguish two different cases: when the planes E and F are parallel and when they are not.

First case: E parallel to F. Partition the union of C and D into the $29^2 = 841$ disjoint lines that intersect E and F orthogonally. By Lemma 18, all but 78 of these lines intersect F in cubes that are reachable from q in at most 40 steps and likewise all but 78 lines intersect F in cubes that are reachable in 40 steps from r. This leaves $841 - 2 \cdot 78 = 685$ lines whose intersections with E and F are reachable in 40 steps from a respectively b. By assumption, there are no more than 666 forbidden cubes in C and D altogether, so several of those lines are completely free. Since the distance between the planes E and F is bounded by twice the sidelength of the boxes C and D, we get a path from q to r of no more than $2 \cdot (40 + 29) - 1 = 137$ steps.

The second case, where E and F are not parallel, can be treated similarly. Only the connecting lines must be chosen in a more complicated way. Partition the union of C and D into 29 parallel planes of size 29×58 such that each plane intersects E and F in exactly one line. Within each of these planes we match the 29 cubes of C with the 29 cubes of D by 29 disjoint paths as displayed in Figure 12. As in the first case, we thus get a positive amount of paths connecting locations in E reachable from q to locations in F reachable from r and free of forbidden cubes. The length bound is a little worse, however. Paths in Figure 12 can require up to $28 + 29 + 28 = 85$ steps, which together with the paths within the planes E and F yields an upper bound of 165 steps from q to r. It is also easily checked that in either configuration we spend no more than 96 steps inside C. $\square$

We want to apply the box-travel lemma to boxes of our hierarchy (H_j). Therefore we have to define which level-$(j-1)$ subboxes inside a level-j box should be considered forbidden. This shall, for now, depend on a simple mass constraint. (Later we will also need a slightly modified definition.)

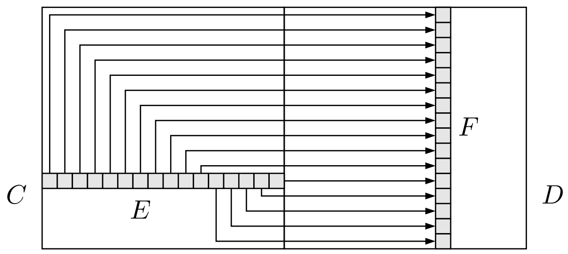


FIGURE 12. Traveling between non-parallel planes E and F.

20. DEFINITION. Call a box $A' \in H_{j-1}$, $j \geq 2$, *light* if

$$\mu(A') \leq \frac{17}{3} \cdot 165^{j-1} \tag{4}$$

and *heavy* otherwise.[2] We then say that the angel's position a is *nice on level* j if the subbox $Q_{j-1}(a)$ lies clear in $Q_j(a)$, with exactly the heavy level-$(j-1)$ boxes forbidden. The position is *nice on level 1* simply if

$$\mu\big(Q_1(a)\big) \leq 1157. \tag{5}$$

We say that a position is *nice up to level* j if it is nice on all levels from 1 through j.

The notion of niceness will be suitable to guarantee an escape route out of the current level-j box $Q_j(a)$. Recall that the constant 165 is exactly the step bound provided by Lemma 19. Level 1 receives a special treatment because it will be used in the induction basis, founding our hierarchy argument on actual angel moves.

The main induction—escaping from larger and larger boxes. With the notion of niceness at hand, it is actually rather straightforward to formulate an appropriate induction hypothesis for angel strategies that allow to travel between arbitrarily large boxes. Only a few constants remain to be chosen thoroughly. And of course, we have to make some assumption on the target box we want to run into. Actually, a simple mass constraint will do.

21. PROPOSITION. *Let B be one of the six level-j boxes neighboring the angel's current box $A \in H_j$, $j \geq 1$. If his current position is nice up to level j and the mass of B is bounded by*

$$\mu(B) \leq 7 \cdot 165^j \tag{6}$$

then the 13-angel can get in no more than

$$2 \cdot 165^{j-1} \tag{7}$$

elementary moves from his actual position in A to some location in B such that after he has arrived there, his position will be nice up to level j again.

[2]We prefer to write $j-1$ instead of simply j to emphasize that although lightness is a property of a single box, it shall always be used in reference to the containing box on level j.

Note that the coefficient 7 in (6) is slightly larger than the 17/3 in (4). So for the box B in Proposition 21, we impose a weaker mass constraint than would be required for being considered light as a subbox in the containing box on level $j+1$. We also remark that 165^j lies somewhere in between the sidelength of a level-j box and the number of points in a face of such a box. One could say that with increasing level, the mass bound (6) grows strictly faster than one-dimensional objects but strictly slower than two-dimensional objects. Likewise the path length (7); compared to the diameter of a level-j box, it gets arbitrarily large, hence, seen from a far distance, the angel slows down to almost zero speed. Compared to surface growth, however, and this is the crucial measure because potential devil obstacles must be two-dimensional, the speed can actually be seen to *increase* by $29^2/165 > 5$ per level.

Proof of Proposition 21. By induction on j. The induction basis is $j = 1$. We have exactly 2 moves to get from the current sidelength-13 box A to an arbitrary elementary cube in B. By niceness, A contains at most 1157 devil blocks and by (6), B contains no more than $7 \cdot 165 = 1155$ blocks. Thus, by the pigeon-hole principle, any 7 planes within the current box A or the target box B contain at least $7 \cdot 13^2 - 1157 = 26$ free locations. Hence, the 13-angel may jump from its current position a to some other place in A at most 7 units away from B. From there he can reach in just one further jump any point within the first 7 layers of B, which still contain some unblocked cubes. He jumps to one of them with his second move. The two devil answers cannot raise the mass of B over 1157, so afterwards the position will be nice on level-1 again, as required.

Induction step from $j-1$ to $j \geq 2$. Niceness of the current position a guarantees that there are at most 333 heavy subboxes A' in A, all the other boxes satisfying the lightness condition (4). In our target box B we also mark forbidden subboxes, based however, on a slightly stronger mass constraint. Mark a level-$(j-1)$ subbox B' in B forbidden if does *not* satisfy

$$\mu(B') \leq \frac{11}{3} \cdot 165^{j-1}. \tag{8}$$

So in B, non-forbidden subboxes are "ultra light." Since 334 such forbidden boxes in B would yield a total mass of

$$334 \cdot \frac{11}{3} \cdot 165^{j-1} > 7 \cdot 165^j,$$

our assumption (6) implies that B contains no more than 333 forbidden boxes.

Now there are two adjacent level-j boxes A and B with at most 666 level-$(j-1)$ subboxes forbidden altogether, based on two slightly different criteria. By niceness on level j of the current position a, the box $Q_{j-1}(a)$ lies clear within the box $A = Q_j(a)$. Further, the neighboring level-j box B contains fewer than 333 forbidden boxes. Hence Lemma 19 applies to A and B, giving a path $(U_0, U_1, \ldots, U_t)$ of level-$(j-1)$ boxes with $t \leq 165$ from the current box $Q_{j-1}(a) = U_0$ to some U_t that lies clear in B with respect to the ultra-light boxes there. Moreover, the lemma guarantees that from U_{97} on all boxes lie in B.

We use this path of boxes to obtain an actual strategy that gets the angel from a to some point in U_t. Niceness up to level j at his starting position a implies niceness up to level $j-1$, so we apply our induction hypothesis on level $(j-1)$ to the pair U_0, U_1, getting the angel to a position within U_1 that is also nice up to level $j-1$ and from there to a nice position inside U_2—and so on, all the way to some b that is nice up to level j in U_t. However, this will only work if the mass constraint (6) is satisfied for the target box U_τ in each single transition between two adjacent boxes $U_{\tau-1}$ and U_τ.

This is easily checked. The whole journey from a to b would grant the devil at most

$$165 \cdot 2 \cdot 165^{j-2} = 2 \cdot 165^{j-1} \tag{9}$$

moves. Even if he spends all of them on a single box U_τ in B, the mass of this box will remain bounded by

$$\mu(U_\tau) \leq \frac{11}{3} \cdot 165^{j-1} + 2 \cdot 165^{j-1} = \frac{17}{3} \cdot 165^{j-1}. \tag{10}$$

For a box U_τ in A we know that it cannot receive more than $95 \cdot 2 \cdot 165^{j-2}$ devil moves before we want to enter it, so that by the time we invoke Lemma 19 the following mass bound will hold:

$$\mu(U_\tau) \leq \frac{17}{3} \cdot 165^{j-1} + 95 \cdot 2 \cdot 165^{j-2} < 7 \cdot 165^{j-1}. \tag{11}$$

Both bounds, (10) and (11), satisfy the requirement (6) of Proposition 21 with j replaced by the appropriate level $j-1$ there. Hence, all those transitions between the U_τ will be possible. Also note that the number of moves counted in (9) is exactly what we had to show for (7).

Eventually, the angel reaches a point b in U_t in the required number of elementary moves such that by that time the resulting position is nice up to level $j-1$. It remains to show niceness on level j. To see this, recall that the relaxed mass bound for the originally ultra-light subboxes in B, which we computed in (10), matches exactly our definition (4) of light boxes. Hence, all subboxes B' of B that are heavy after the angel's trip from a to b, had already been forbidden in the beginning when the box-travel lemma was invoked, and thus the terminal box U_t lies clear in B with respect to those boxes. In other words, b is nice on level j, too. $\square$

Proposition 21 immediately implies the *existence* of an escape strategy. But since the following argument uses Lemma 3, we do *not* get an explicit strategy, yet.

Proof of Theorem 16 (non-constructive version). At the very beginning of the game all boxes on all levels of our hierarchy are empty and thus light within their respective containing boxes. Because of the symmetry of the hierarchy with respect to the origin, the angel starts at the very center of the box $Q_j(0)$ on every level $j \geq 1$. Therefore, the starting position is nice on every level $j \geq 1$.

By Proposition 21 the angel can thus travel to some adjacent box on any previously given level of the hierarchy, which allows him to escape the devil for any previously chosen amount of time. So by Lemma 3, the angel can escape forever. $\square$

An explicit infinite escape strategy. If someone really wants to play the angel game for some infinite time, the previous, abstract proof is no big help, telling us just that the angel *can* win—somehow. To obtain an explicit escape strategy, we have to work a little harder and revisit some details of the proof of Proposition 21.

Proof of Theorem 16 (constructive version). We start escape strategies on *all* levels of the hierarchy simultaneously—in such a way that on initial segments those strategies are compatible. Therefore we introduce a small technical convention about the paths provided by Lemma 19.

Unrolling the induction in the proof of Proposition 21, we can interpret that result as a concrete strategy for journeys between adjacent boxes of our hierarchy, which on each level invokes Lemma 19 as an algorithm (implicitly given by its proof) for path finding in grid graphs. In this algorithmic view, let us agree that whenever Lemma 19 is invoked to find a path between two boxes that contain no forbidden cubes at all, it returns a path that starts with a step in the direction of the target box.

The angel begins by traveling from the origin 0 to a nice position a_1 in the level-1 box B_1 that lies directly behind (in positive z-direction, say) the initial box $Q_1(0)$. Having arrived at position a_1, we can now interpret these first steps as the initial sequence of a travel from the box $Q_2(0)$ to a nice position a_2 in the level-1 box B_2 just behind the initial level-2 box $Q_2(0)$. As we already observed in the non-constructive proof above, such a strategy exists by Proposition 21 and by our convention it would have started with a travel to a position in B_1, just as we did. We now follow the new level-2 strategy until we reach the position a_2. At that point, we again interpret this journey as the initial sequence of a travel from the origin to a nice position a_3 in the level-3 box behind $Q_3(0)$. Iterating this argument indefinitely, we obtain an infinite escape strategy for the angel. The crucial argument here is that what we have done up to some point, will always fit into strategies on higher levels that we have not considered yet. □

Why our hierarchy does not work in 2D. One might want to try to transform the hierarchy approach for the three-dimensional case into an escape strategy for the two-dimensional game. Such an attempt would face two major obstacles. First, as we already remarked after the statement of Proposition 21, the step bound (7) grows strictly faster than the sidelengths of the boxes. This effect is due to the detours we are making with each application of Lemma 19. On higher and higher levels, the effective speed of the angel thus gets arbitrarily slow. In the plane, this would allow the devil to completely encircle the angel on a sufficiently large scale since the boundary of a rectangle is proportional to the radius. Hence, we would need an improved path finder that might probably employ some means of charging devil moves against angel moves such that devil plays that force the angel to make detours cannot be counted for wall building far away.

But even if one should succeed in maintaining the “effective speed” of the angel, there would remain a more fundamental problem about hierarchical strategies like the one we presented. While routing out of a level-j rectangle R (or whatever regular shape might be used), the angel must at some point decide which of the subrectangles on level $j-1$ should be the last on the way out. Then he will have to pass through the outward side S' of this subrectangle R' at some time in the future. While the angel approaches R', the devil uses a certain number of his moves, proportional to the sidelength of R', to destroy points of S' at some density. After the angel has entered R', he must then, as before, pick some subrectangle R'' of R' that should be the last before he leaves R' through S' and thereby confine himself to pass through its outward side $S'' \subset S'$, shown in Figure 13. Again, the devil uses a certain number of moves to increase the density on S'' *by the same amount* as on the previous level.

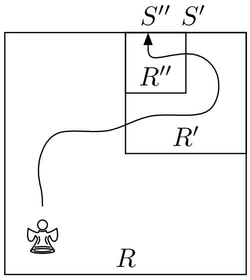


FIGURE 13. A boxed fool.

Repeated application of this scheme on sufficiently many levels eventually yields a completely blocked line through which the angel would have to travel. The reader will have noticed that what we just sketched is simply a hierarchical version of Conway's fool theorem. The implication for hierarchical approaches in the plane is clear: The different levels of an angel's hierarchy will have to interact in a considerably more sophisticated way than is sufficient for an escape in space.

CHAPTER 2

Weak Positional Games

1. Tic-Tac-Toe

Let $H = (V, E)$ be a hypergraph, that is, $V = V(H)$ is a finite set and $E = E(H)$ is a set of subsets of V. The elements of V are called the *vertices* of H and the sets in E are the *edges* of H. Two players, called *Maker* and *Breaker*, play the following game on H. Maker begins by picking some vertex of H, then Breaker chooses some different vertex. They alternate in this fashion until all vertices of H are taken, retaking of vertices being forbidden. Maker wins if he manages to claim all vertices of some edge $e \in E$, otherwise Breaker wins.

Note the obvious unfairness, or rather asymmetry in the game. Breaker does not win by getting a complete edge as Maker does. His moves are only meant to block vertices and make the incident edges useless for Maker. Also observe that by definition, there cannot be a draw.

Such a game is called a *weak positional game* on the hypergraph H. The term *positional game* goes back to Hales and Jewett [**19**] where a variant of such games were first studied. The attribute "weak" has been coined later to distinguish them from the so-called "strong" games which we shall address soon. Briefly, "weak" accounts for the fact that Breaker does not win when he claims an edge $e \in E$ himself.

The relevant question about a game on a fixed hypergraph is, of course, who can win. That is, does Maker or Breaker have a strategy that always wins. Formally, a *strategy* is a mapping σ from finite sequences $(x_1, x_2, \ldots, x_r)$ of distinct vertices of H to $V(H) \setminus \{x_1, x_2, \ldots, x_r\}$, where $r < |V(H)|$. The obvious semantic being that the x_i describe the course of play up to some point and then σ determines the next move. So in case of a Maker strategy σ is only defined for sequences of even length and only for sequences of odd length in case of Breaker strategies.

A *winning strategy* is a strategy that wins against all possible opponent plays. A fundamental theorem of combinatorial game theory tells us that either one of the two players must have such a winning strategy (games with this property are called *determined*) draw being impossible by the very definition of the game. This is easily shown by a simple game-tree backward-labeling argument, as described in many books on combinatorial games. The essential ingredient here is the finiteness of the game. See Section 2 of Chapter 1 for a brief discussion of some aspects of non-determined games.

Winning strategies for Maker will sometimes be called *making strategies* and such for Breaker *breaking strategies*. In our arguments, we usually like to consider a game out the perspective of Maker, which suggests the following convention.

1. Definition. A hypergraph H is a *winner* if Maker, playing first, has a winning strategy on H, otherwise, when Breaker has a winning strategy, we call it a *loser*.

In this work, our main motivation to study positional games is the computational complexity of the question whether a given hypergraph is a winner. Note that an efficient decision procedure for this question would immediately yield winning strategies on any winner by a standard reduction. At each move, we could simply determine the value of the outcomes of all our options together with all possible opponent plays. From this we would then be able to tell which moves are the best.

However, a polynomial-time algorithm for arbitrary hypergraphs should not be hoped for. Schaefer [**39**] showed that this problem is PSPACE-complete, which is "the right" class for a two-person game. The paper does not use the term hypergraph, though, but works with games on DNF formulas, which behave equivalently. Thanks to Jesper Makholm Byskov for pointing me at that result.

We will focus on hypergraphs with edges of bounded size.

2. Definition. The *rank* of a hypergraph is the size of a largest edge. A hypergraph is called *k-uniform* if all its edges are of size k.

Hypergraphs of rank 2 are not very interesting from the point of positional games. Any edge of size 1 yields an immediate Maker win, so we may assume that the hypergraph is 2-uniform, i.e., an ordinary simple graph. If such a hypergraph has any vertex of degree greater than one, i.e., if any two edges share a vertex, Maker wins by playing at such a vertex because in his next move he will complete either of the two edges since Breaker can only play in one of them. On the other hand, Schaefer's proof requires no edges larger than 11, so that the decision problem is already PSPACE-complete for hypergraphs of rank 11.

In this interval, between 2 and 11, the smallest interesting rank is 3. We set out to distinguish rank-3 winners from rank-3 losers efficiently, i.e., in polynomial time. Unfortunately, we do not succeed completely. There is a problem with too-much-overlapping edges. We shall solve the task only for hypergraphs with the following additional property.

3. Definition. A hypergraph is *almost-disjoint* if no two edges intersect in more than one vertex.

4. Theorem. *The question whether a given almost-disjoint hypergraph of rank-3 is a winner or a loser can be decided in polynomial time.*

Theorem 4 is not about efficient algorithms. Our motivation is not the desire to actually play such games better, like with chess, but to understand the underlying principles which let you win or lose on a hypergraph. The above result rests on a classification of rank-3 hypergraphs into winners and losers, which is somehow the more important result. That classification (Theorem 38) depends on several notions that first need to be developed, so that we must defer its statement to a later place, where the actual work is done.

It might be suspected that by restricting ourselves to almost-disjoint hypergraphs, we have defined away the essential part of the problem. This is not the case. Our investigation of almost-disjoint hypergraphs exhibits a lot of structure and the techniques we employ during the analysis reveal some of the deeper mechanics behind such games. Moreover, we shall give some evidence that the almost-disjointness condition can be removed through a preprocessing step, so that our result could be immediately applied to all rank-3 hypergraphs without further modifications in the proof.

Strong games. Positional games can be seen as the natural generalization of the well-known game *Tic-Tac-Toe*, which is played by two players on a board of $3 \times 3 = 9$ squares. Alternately the opponents claim squares, the first player by drawing crosses the second by drawing noughts; reclaiming of previously taken squares being forbidden. Either player wins if he manages to get three squares in a row, horizontally, vertically, or on one of the two diagonals. Figure 1 shows a game in progress.

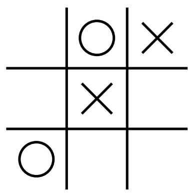


FIGURE 1. A game of Tic-Tac-Toe.

Note the obvious difference to weak positional games. In Tic-Tac-Toe the second player also tries to complete an edge of his own. In some sense, the game now seems fairer.

The natural generalization of this symmetric rule system to arbitrary hypergraphs $H = (V, E)$ leads to the definition of a *strong positional game*. Two players, not called Maker or Breaker now, alternately claim vertices in V until either one player has claimed all vertices of some edge $e \in E$, in which case he wins, or all vertices are claimed and neither player achieved this goal, in which case the game is a draw. The term "strong" will soon become clear when we relate these games to weak games.

The difference between weak and strong games already bears on the simple example of Tic-Tac-Toe. While every child knows that it is a draw in the strong version, Maker can win on the 3×3 board in the weak game because in certain situations Breaker lacks counter threats.

Due to the changed game definition we get a new type of strategy. A *drawing strategy* is a strategy that always leads to at least a draw, i.e., if you follow this strategy you can be sure not to lose and it may happen that you win. Similar to the case of weak games, a simple game-tree argument shows that either one of the two players has a winning strategy or both players have drawing strategies. A special feature of strong positional games is that this trichotomy (first player win, second player win, and draw) collapses to only two cases. The second player cannot win, as can be seen by the following well-known *strategy-stealing* argument. Assume for contradiction that the second player has a winning strategy. Then the first player can "steal" this strategy by playing his first move anywhere and then behaving as if he was the second player. The point is that the additional first move does not create any problems for the first player because of the *monotonicity* of the game. If the strategy prompts him to play a vertex he has already taken, he can just play this move anywhere else and still have all vertices taken that the strategy requires. Having more vertices claimed is never a disadvantage.

So, strong games, if played optimally, also have just two different possible outcomes: first player win or draw. The following trivial statement relates weak and strong games in terms of winning strategies, justifying the pair "weak"/"strong."

5. Remark. If the first player can win the strong game on a hypergraph H, Maker can win the weak game on H. If Breaker can win the weak game on a hypergraph H, the second player can force a draw in the strong game.

So, taking Maker's respectively the first player's perspective again, being able to win a strong game is really a stronger statement than being able to win the corresponding weak game. Beck's survey paper [**7**] contains a detailed discussion of the relation between weak and strong positional games.

Previous Results. A main branch of research about positional games aims at the development of strong criteria for the existence of winning strategies, often in terms of the number of edges and vertices, like the following early result by Erdös and Selfridge [**15**].

6. Theorem (Erdös-Selfridge). *Let $H = (V, E)$ be an n-uniform hypergraph. If $|E| < 2^{n-1}$ then Breaker wins the weak game on H and thus the second player can draw in the strong game.*

Beck [**4, 5**] has developed a variety of strong conditions of this kind. We refer to his extensive overview [**7**].

Sometimes hypergraphs are investigated that are implicitly defined by certain regular structures. For example, in [**20**] and [**6**] the two players pick edges from a complete graph and try to obtain a subgraph of a certain prescribed type. Another famous class of hypergraphs are generalized Tic-Tac-Toe boards, where the vertex set is the n^d grid cube $\{1, \ldots, n\}^d$ embedded in d-space with exactly all collinear n-sets as edges. These games have already been studied in Hales and Jewett's original paper [**19**]. Berlekamp, Conway, and Guy's classic [**8**] contains a whole chapter about some sorts of positional games, like five-in-a-row on a checker board and games with polyominoes. It also contains a detailed case analysis of the original 3 by 3 Tic-Tac-Toe.

Eventually, we should mention that also strong positional games are PSPACE-complete. Reisch [**37**] showed this for the special case of the board game Gomoku (five-in-a-row in the plane).

Our approach to positional games very much differs from most of the above in that it aims at *optimal* play for a limited class of hypergraphs. While density arguments like Theorem 6 usually give winning or losing criteria for much larger classes of games than the one we attempt to solve, they cannot give definite answers how to play on any arbitrary given instance. Usually the gap between the best winning criterion and the best losing criterion is rather large, leaving a lot of difficult instances unresolved.

The price we must pay for our desire for a complete analysis, are several lengthy case distinctions and sometimes a certain lack of beauty. Quite in contrast to the nice density theorems of [**19**] and [**15**]. Though we introduce tools to break hypergraphs into nice components, it cannot be avoided that eventually some dirty parts have to be sorted out by direct inspection. The ultimate result however, will be rather concise, a neat classification into winners and losers.

2. Winning Ways

Before we embark on the analysis of rank-3 games, let us briefly discuss a few very basic concepts and fix some related terminology. Consider a single move of Breaker at some vertex y. Clearly, all edges of H that contain this vertex will be of no use for Maker any more because he is not allowed to ever recolor y. So we may interpret Breaker's move as deleting the vertex y and all incident edges $f \ni y$ from H. On the other hand, a Maker move at some vertex x brings Maker one step closer to his goal in each edge that contains x. His move can be understood as shrinking all edges $e \ni x$ by the vertex x, i.e., deleting x from $V(H)$ and replacing each such e by $e' = e \setminus \{x\}$. In this interpretation, Maker wins iff he manages to produce an empty edge. Note how this point of view captures the inherent asymmetry of the game and it is very useful to analyze hypergraphs in which some vertices have already been played by any of the two players. We let

$$H^{[+x_1,\ldots,+x_r,-y_1,\ldots,-y_s]} \tag{12}$$

denote the hypergraph obtained from H by "shrinking away" the Maker vertices $x_1, \ldots, x_r$ and deleting all edges containing any of the Breaker vertices $y_1, \ldots, y_s$ in the above fashion. We shall also use obvious abbreviations of this expression like $H^{[+M]}$ with $M = \{x_1, \ldots, x_r\}$ a set of Maker moves.

Formally, there is no need for the numbers r and s in (12) to be related in any way. We can have, for example, a large number of Maker plays in H but no Breaker moves at all. This expression will still make sense. Hence, our notation can be used to describe the course of play on local fragments of a hypergraph, where the players not necessarily play in alternating fashion. In other words, we can treat *tenuki*—moves that do not directly answer the opponents preceding move locally but shift play to another part of the graph.[1] Second, the resulting hypergraph is clearly independent of the order of deletion and shrinking steps. This is convenient for analyzing snap shots of a game without bothering about the precise order of moves that lead to an actual position.

Playing along paths. We start our investigation of rank-3 hypergraphs by collecting some elementary, though important criteria that guarantee a Maker win. The crucial objects are paths.

[1]In the Asian board game *Go*, a move that stays away from a local fight is called *tenuki*.

7. Definition. A *walk* (from a vertex v_0 to another vertex v_r) in a hypergraph is a sequence $W = (v_0, e_1, v_1, \ldots, e_r, v_r)$, $r \geq 0$, of vertices v_i and edges e_i, such that $v_{i-1}, v_i \in e_i$ for $1 \leq i \leq r$. The index r is the *length* of the walk and we call v_0 and v_r the *start* and *end vertex* of the walk, respectively.

A walk is a *path* if all vertices v_i are distinct and $e_i \cap e_j = \emptyset$ for all pairs of indices i, j with $|i-j| > 1$. A *cycle* in a hypergraph is a walk of strictly positive length from a vertex to itself, satisfying all requirements of a path except that, of course, $e_1 \cap e_r$ must not be empty.

We shall often treat a path or cycle W itself as a hypergraph by letting $V(W) = \{v_0, \ldots, v_r\} \cup e_1 \cup \cdots \cup e_r$ and $E(W) = \{e_1, \ldots, e_r\}$. As usual, we say that two vertices of a hypergraph are *connected* if there is a walk between them, and a hypergraph is *connected* if any two of its vertices are.

In contrast to simple graphs, one might come up with alternative definitions for the concept of paths in hypergraphs. We just chose the one that will best serve our purposes. Though the following two notions are absolutely standard and should not bear any ambiguities, we like to provide a rigorous definition.

8. Definition. A *subhypergraph* of a hypergraph H is another hypergraph K with $V(K) \subseteq V(H)$ and $E(K) \subseteq E(H)$. The *induced subhypergraph* on a vertex set $W \subseteq V(H)$ of a hypergraph H is defined as the hypergraph $H[W] := \big(W, \{e \in E(H) \mid e \subseteq W\}\big)$.

If a and b are vertices of a path, we write aPb for the unique subpath of P from a to b. We often stack several such subpaths of different paths to obtain a single long path. For example, if a and b are vertices of paths P and Q, respectively, and P and Q intersect in some other vertex x, then we write $aPxQb$ for the path from a to b in $P \cup Q$ via x. Of course, we then have to check that the resulting walk is a path again but in most cases this will be obvious. When the path we want to use consists of only one edge, $f = \{a, b, x\}$, for example, we sometimes simply write afb. We also use constructs like $aPxQa$ to create a cycle from two paths that intersect in two vertices a and x.

The following lemma is rather trivial, but as we already emphasized, paths in hypergraphs require a slightly more careful treatment than paths in simple graphs. So we like to give a rigorous proof here to make sure not to overlook any details and to comply with our definitions.

9. Lemma. *If two vertices a, b in a hypergraph are connected then there exists a path from a to b.*

Proof. We claim that any shortest walk $(v_0, e_1, v_1, \ldots, e_r, v_r)$ from a to b is actually a path. Otherwise there would be two edges e_i, e_j with $i < j-1$, such that the intersection $e_i \cap e_j$ contains some vertex x. But then the sequence $(v_0, e_1, \ldots, e_i, x, e_j, \ldots, e_r, v_r)$ is a shorter walk from a to b; a contradiction. □

Figure 2 shows a path of length 7 with five 3-edges in the interior and a 2-edge at each end. Assume that Maker plays at x_1. Then Breaker must clearly answer at y_1. After that, Maker x_2 leaves only Breaker y_2 and then, Maker x_3 forces Breaker y_3. And so on. Maker can play all the way down to x_6, where he wins because Breaker will have to answer x_6 at y_6, leaving the singleton edge $\{x_7\}$ for Maker.

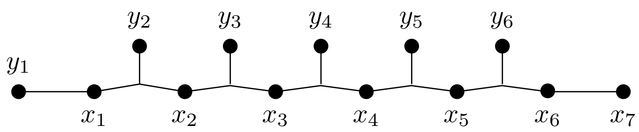


FIGURE 2. A winning path.

This scheme only works because any two adjacent edges of the path intersect in just one vertex. The hypergraph in Figure 3 is a loser. If Maker tries the same trick there, he gets stuck in the middle because after Maker x_4 there, Breaker y_4 will destroy his options for the right side. However, if the hypergraph at hand is almost disjoint then all paths are nice.

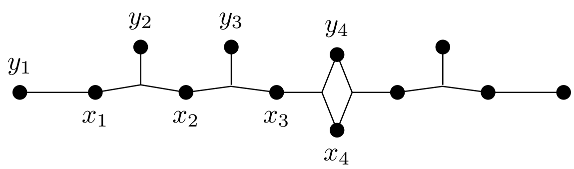


FIGURE 3. A non-almost-disjoint losing path.

10. LEMMA. *An almost-disjoint rank-3 hypergraph that has a path containing two 2-edges is a winner.*

PROOF. We may assume that the path contains exactly two 2-edges and that these are in the two terminal positions by simply removing further 2-edges and trailing 3-edges. So we have a path $(v_0, e_1, v_1, \ldots, e_r, v_r)$ where e_1 and e_r are 2-edges and the other e_i are 3-edges. Maker wins by playing along this paths as described above. $\square$

Combining Lemmas 9 and 10 we get the following useful win criterion.

11. COROLLARY. *Any connected almost-disjoint rank-3 hypergraph with at least two 2-edges is a winner.* $\square$

During the analysis of a game in progress, it will often be useful to have the following variant of Lemma 10 available, which tells us how Breaker has to reply to a Maker move in a component with a 2-edge.

12. LEMMA. *Let P be a path in an almost-disjoint rank-3 hypergraph and assume that P contains a 2-edge. If Maker plays somewhere in P then Breaker must answer somewhere in P, too; otherwise Maker wins.*

PROOF. If Maker plays inside the 2-edge the statement is trivial. Otherwise, Maker creates an additional new 2-edge that lies on a common path with the original 2-edge. By Lemma 10, Breaker must answer on this subpath of the original path. $\square$

Inner and outer vertices. Let us have a closer look at that carbon molecule in Figure 2 again. The vertices that Maker played there shall be of general importance for us.

13. DEFINITION. Let P be a path or a cycle. A vertex of P that appears in more than one edge is called an *inner vertex* of P; the other vertices are the *outer vertices* of P.

The way Maker won in Figure 2 was not unique. It is not hard to see—though we won't prove this now—that he could have started at any of the inner vertices and still have won, while the outer vertices would have all lead to a loss. The reason for this is in a way to be found in the following absolutely trivial, yet important fact.

14. REMARK. If x is an inner vertex on an almost-disjoint path P from a to b then the subpaths xPa and xPb only intersect in the vertex x. Similarly, if Maker plays at an inner vertex of an almost-disjoint cycle, this cycle is split into a path.

Note that outer vertices do *not* have this property. The following two lemmas, which will be useful in many situations, exploit the above observation for cycles.

15. LEMMA. *Let C be a cycle in an almost-disjoint rank-3 hypergraph. If Maker plays at an inner vertex of C then Breaker must answer somewhere in C, too; otherwise Maker wins.*

PROOF. Playing at an inner vertex, Maker turns the cycle into a path with a 2-edge at each end, which by Lemma 10 is a winner. See the left-hand side of Figure 4. □

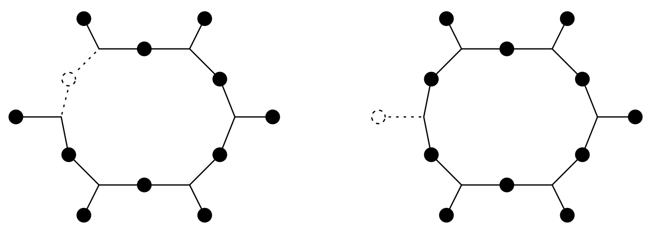


FIGURE 4. Playing an inner vertex of a 3-uniform cycle yields a path with two 2-edges (left), playing an outer vertex yields a cycle with a 2-edge (right).

Of course, it is crucial again to pick an inner vertex. Playing an outer vertex of a cycle yields just a cycle with a 2-edge, as shown on the right-hand side of Figure 4. If Maker then plays in such a cycle again, Breaker has only few options left.

16. LEMMA. *Let C be a cycle in an almost-disjoint rank-3 hypergraph and assume that C contains exactly one 2-edge. If Maker plays at an inner vertex of C then Breaker must answer in the 2-edge; otherwise Maker wins.*

PROOF. If Maker plays in the 2-edge, the statement is trivial. Otherwise, his move, which breaks up the cycle into a path, creates two 2-edges. This leaves a path with three 2-edges altogether. If Breaker does not play in the original 2-edge now, which is clearly the middle one, he leaves behind two 2-edges connected by a path. A win, by Lemma 10. □

3. Decomposing Hypergraphs

The last two lemmas from the previous section demonstrated the potential of cycles for Maker. With a single move at an inner vertex of a cycle he could create an immediate threat. A key tool for our analysis of hypergraph games will be a decomposition lemma that allows us to reduce any hypergraph into parts that are doubly connected in a certain way. In those parts we will then have good chances to find cycles that yield several Maker threats, allowing us to construct winning strategies for Maker.

We start with a simple observation about disconnected hypergraphs. For two hypergraphs H_1 and H_2, their union $H = H_1 \cup H_2$ is given simply by $V(H) = V(H_1) \cup V(H_2)$ and $E(H) = E(H_1) \cup E(H_2)$. In case of disjoint vertex sets $V(H_1)$ and $V(H_2)$ this yields a disconnected union $H = H_1 \dot\cup H_2$. It appears plausible that in such a case, moves played in one component should not interfere with those played somewhere else. Let us formalize this intuition.

17. LEMMA. *The disjoint union $H = A \dot\cup B$ of two hypergraphs A and B is a winner iff at least one of A and B is a winner.*

PROOF. If A or B is a winner then clearly H is. So assume that neither A nor B can be won. So there are breaking strategies α and β for A and B, respectively. Against any Maker strategy, Breaker can use these to obtain a breaking strategy for H. Whenever Maker plays in A he answers according to α, when Maker plays in B Breaker follows β, at each move always ignoring anything that happened in the other component. This way Breaker can assure that in none of the two components Maker can get a monochromatic edge. Thus, H is a loser. □

Lemma 17 tells us that if Maker can win on some hypergraph H he only needs one component of H, never playing in the rest of H. And of course, this rule can be applied recursively to any stage of the game: Maker never needs to leave a component he once played in.

Splitting at articulations. Lemma 17 is not very deep. But it paves the way for a stronger result that will become a vital tool for our analysis of games on rank-3 hypergraphs. Suppose that the components A and B of H are not completely disjoint but almost, i.e., they share just one vertex. Then we can still relate the winning and losing behavior of A and B to that of H.

18. DEFINITION. We call a vertex p an *articulation vertex* of a connected hypergraph H if H can be written as a union $H = A \cup B$ of two nontrivial hypergraphs A and B with $V(A) \cap V(B) = \{p\}$.

The left hypergraph in Figure 5 has exactly one articulation vertex, the square one. The central vertex in the hypergraph on the right is *not* an articulation.

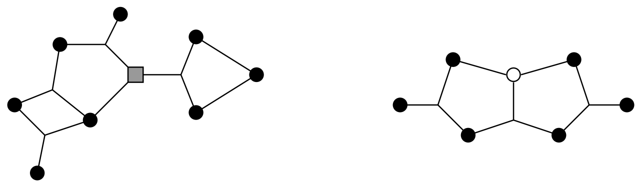


FIGURE 5. A hypergraph with an articulation vertex (left) and one without (right).

19. LEMMA (Articulation Lemma). *Let $H = A \cup B$ be the union of two hypergraphs A and B which have exactly one point p in common, i.e., $V(A) \cap V(B) = \{p\}$. Then H is a winner if and only if one of the following holds:*

- *A is a winner,*
- *B is a winner,*
- *$A^{[+p]}$ and $B^{[+p]}$ are both winners.*

PROOF. First note that every single one of the three cases implies a win for H. For the first two this is clear. If the last case holds, Maker can win by playing his first move at p. This leaves two disjoint graphs both of which he can win. Breaker cannot answer in both, so at his second move, one of $A^{[+p]}$ and $B^{[+p]}$ will still be available to Maker and give him a win.

For the converse implication consider the case that none of the three options in the statement of the lemma is true. By symmetry we may assume that $B^{[+p]}$ is a loser. So we have breaking strategies α and β for A and $B^{[+p]}$, respectively. Breaker combines these strategies as follows. Against any Maker move in A he also answers in A, according to his strategy α. When Maker plays in $B \setminus \{p\}$ he answers there, following to strategy β. This way Maker can never complete one edge of H since the edges of A are taken care of by α and the strategy β guarantees that even if Maker should get the vertex p, it won't help him on B because not only B but even $B^{[+p]}$ was a loser. $\square$

Note that we had to require the nontriviality and connectivity condition in the definition of an articulation vertex for technical reasons. (Otherwise every vertex would be an articulation.) Lemma 19 does obviously not depend on such restrictions.

Figure 5 indicates that in contrast to simple graphs, hypergraphs allow different notions of connectivity. If we removed the central vertex from the right hypergraph in that picture together with all incident edges, we would of course decompose the hypergraph into disjoint components. But that is not what we want because the Articulation Lemma does not apply to that hypergraph. The "right" notion of connectivity for us is the following.

20. DEFINITION. A hypergraph H with at least k vertices is *Maker-k-connected* if its reduction $H^{[+M]}$ is connected for every set $M \subseteq V(H)$ of Maker moves that has cardinality strictly less than k.

Practically, Maker-k-connectivity means that Maker would have to play at least k times until the hypergraph decomposes. Note also that Maker-1-connectivity is equivalent to ordinary connectivity because then $M = \emptyset$ is the only allowed set of Maker moves. We refrain from defining the analog concept of "Breaker-connectivity" since we shall not need it anyway.

21. LEMMA. *A hypergraph H with at least k vertices is Maker-k-connected iff it cannot be written as a union $H = A \cup B$ with $V(A), V(B) \neq V(H)$ and $|V(A) \cap V(B)| < k$.*

The crucial property here is, of course, that the hypergraphs A and B do not overlap on k vertices, the other restriction only makes sure that the decomposition is nontrivial in the sense that A and B are both really needed in the union.

PROOF OF LEMMA 21. Assume that we have such a representation $H = A \cup B$. Taking $M = V(A) \cap V(B)$ immediately gives us a Maker set such that $H^{[+M]}$ is disconnected. Conversely, assume that there exists a set $M \subset V(H)$ of cardinality $\ell < k$ such that the reduced hypergraph $H^{[+M]}$ is disconnected, i.e., $H^{[+M]} = A' \,\dot{\cup}\, B'$. This decomposition tells us that any H-edge lies completely in $V(A') \cup M$ or $V(B') \cup M$. Therefore, we can write H as the union $H = H^{[-V(B)]} \cup H^{[-V(A)]}$, where the two vertex sets intersect in the set M which has cardinality $\ell \ < \ k$. $\square$

22. COROLLARY. *A hypergraph with at least two vertices is Maker-2-connected iff it is connected and contains no articulation vertex.* $\square$

Path decompositions. Through repeated application of Lemmas 17 and 19 we will reduce statements about general hypergraphs to such about Maker-2-connected hypergraphs. Those are then amenable to the following path-adjoining lemma, which is very much redolent of classical ear-decomposition theorems. Here, however, it appears in a slightly technical guise, due to the special requirements of our analysis in the subsequent sections.

23. LEMMA. *Let H be a rank-3 Maker-2-connected hypergraph and let (B, M, T), with $\emptyset \neq B, M, T \subsetneq V(H)$, be a nontrivial partition of the vertices of H such that no vertex in B is adjacent to a vertex in T. In other words, the "middle layer" separates "bottom" from "top." Then there exists a path in $H[M \cup T]$ connecting two distinct vertices a and b in M and using no further vertices in M and no edges of $H[M]$.*

In one sentence, Lemma 23 tells us that if we step from the middle layer into the top layer then we find a path through T that brings us back to M.

Have a look at Figure 6. In the typical application of Lemma 23, the middle layer M will be a part of a hypergraph H that we are currently reconstructing and about which we already know a lot of structure, while the top layer T contains the unexplored parts of H that are somehow connected to M. The lemma then tells us that we can extend M into T path by path in a regular fashion. The lower layer B contains all the remaining vertices that are of no interest for the local situation.

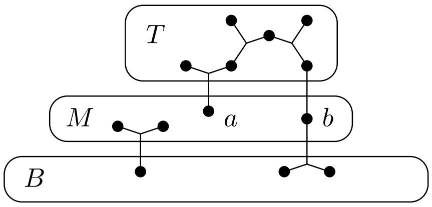


FIGURE 6. Finding paths with Lemma 23.

PROOF OF LEMMA 23. Pick a connected component C of the hypergraph $H[M \cup T] - E(H[M])$ (i.e., the subhypergraph induced by the vertex set $M \cup T$ without those edges that lie entirely in M) that contains at least one vertex in T. For example, in Figure 6, the path from a to b through T would be such a component C. The intersection $X = V(C) \cap M$ has cardinality at least two because H is Maker-2-connected. ($C^{[+X]}$ contains no vertex in M, so it is disconnected from B by assumption; therefore, the hypergraph $H^{[+X]}$ itself is disconnected and thus, $X \geq 2$.)

For each pair u, v of distinct vertices from X, pick a shortest path $P_{u,v}$ from u to v in C. Such paths exist by Lemma 9. Amongst all these paths (for all possible pairs u, v) pick one, $P_{a,b}$, say, of minimal length. We claim that this is a path as required by the statement of the lemma. Assume for contradiction that $P_{a,b}$ contains more vertices in M than only a and b, some additional vertex c, say. The three vertices a, b, c cannot lie in the same edge of C because then they would form an induced edge of $H[M]$ which we had excluded. Consequently, one of the paths $P_{a,c}$ and $P_{c,b}$ must be shorter than $P_{a,b}$—a contradiction to minimality. □

Creating a 2-edge. In Section 2 we emphasized that in a rank-3 hypergraph, 2-edges are good for Maker. Already two of them lead to a win if the hypergraph is almost disjoint, by Corollary 11. In this section we show how to reduce the problem whether a 3-uniform hypergraph is a winner, to the question whether a rank-3 hypergraph with a 2-edge is a winner. Those will then be easier to analyze.

24. LEMMA. *Let H be a 3-uniform hypergraph that is a winner. Then there exists a Maker move x such that for any Breaker answer y, the hypergraph $H^{[+x,-y]}$ has a connected component that contains a 2-edge and is a winner.*

PROOF. By induction on the size of H. Take the first move x from any making strategy for H. Assume for contradiction that for some Breaker answer y the hypergraph $H' = H^{[+x,-y]}$ has no connected component that is a winner with a 2-edge. By Lemma 17, H' must have a component W that is a winner and by assumption, W contains only 3-edges. But such a W is actually a proper subhypergraph of H, so by induction there is a Maker move $\hat{x} \in V(W)$ such that for every Breaker answer $\hat{y} \in V(W) \setminus \{\hat{x}\}$ the remainder $W^{[+\hat{x},-\hat{y}]}$ has a winning component that contains a 2-edge. Since W was a subgraph of H, we can use $\hat{x}$ as the first Maker move in H as well and this will then guarantee a winning component with a 2-edge after any Breaker answer. □

One could easily generalize the proof of Lemma 24 to show that Maker can actually win by always playing inside a component that contains at least one 2-edge, except for his first and last move, of course. But all we need here is a 2-edge after the first move as guaranteed by Lemma 24 because it gives us the following reduction from a 3-uniform hypergraph H to hypergraphs with at least one 2-edge.

For each pair x, y of first Maker and Breaker moves, check whether among those components of $H^{[+x,-y]}$ that contain a 2-edge there is at least one winner. If for some x this is the case for all possible answers y then H is a winner, otherwise it's a loser.

Once we have a 2-edge, we use the Articulation Lemma to cut our hypergraph recursively at articulation vertices, so that eventually we will be left with Maker-2-connected hypergraphs only. Having created a 2-edge is really important for this step. In the proof of the subsequent lemma, the presence of the 2-edge eliminates one alternative in the Articulation Lemma, giving us sufficient information to avoid a possible combinatorial explosion during the decomposition process.

25. Lemma. *Let H be an almost-disjoint connected rank-3 hypergraph with exactly one 2-edge f. Let $H = A \cup B$ be a decomposition of H with $V(A) \cap V(B) = \{p\}$ for some articulation vertex p, such that f lies in A. Let $B_1, \ldots, B_r$ be the connected components of the hypergraph $B^{[+p]}$. Then each of the connected hypergraphs $A, B_1, \ldots, B_r$ contains at least one 2-edge, and H is a winner iff at least one of them is a winner.*

Proof. Since H was connected, each of the B_i has at least one edge that contained the deleted vertex p in H. Hence, those edges are 2-edges. Clearly A is connected simply because H is and it contains a 2-edge by assumption.

For the stated equivalence, first observe that the preconditions alone imply that $A^{[+p]}$ is a winner: if $p \in f$ then because $A^{[+p]}$ contains a 1-edge and otherwise because $A^{[+p]}$ has at least two 2-edges. Now the Articulation Lemma tells us that H is a winner iff one of A and $B^{[+p]}$ is. (Since $A^{[+p]}$ is a winner, the third case of the Articulation Lemma reduces to "$B^{[+p]}$ is a winner," which makes the second case obsolete.) And by Lemma 17, $B^{[+p]}$ is a winner iff one of the B_i is. □

We use Lemma 25 as an algorithmic recipe for reducing the problem of deciding whether a given connected almost-disjoint rank-3 hypergraph H with exactly one 2-edge is a winner, to such hypergraphs that are even Maker-2-connected instead of just connected.

If an application of Lemma 25 yields any B_i with more than one 2-edge, this B_i is a winner by Corollary 11, and then H is one, too. Otherwise, we apply Lemma 25 recursively to each of $A, B_1, \ldots, B_r$ until we either find a component with two 2-edges or no articulation vertices are left and hence, all pieces are Maker-2-connected. (Remember that a single 2-edge is by definition Maker-2-connected.) Eventually we know that the original hypergraph H is a winner iff one of those Maker-2-connected fragments is.

4. Between the Docks

We are left with the task of finding out whether a given almost-disjoint rank-3 Maker-2-connected hypergraph H with exactly one 2-edge is a winner. Figuratively, we shall view the unique 2-edge, which we will henceforth denote by $\phi = \{\alpha, \beta\}$, as sitting at the center of H and everything else arranged around it. We then try to understand how this environment can look like, under what conditions it yields a win and why it perhaps does not.

Call all edges adjacent to ϕ *dock edges*, motivated by the fact that the rest of H is connected to ϕ through them. Anything else between the docks, that is, the subhypergraph of H with α, β, and all dock edges removed, will be called the *core*, denoted by K.

By almost-disjointness, each dock edge contains only one of α and β, so anticipating the way we shall draw pictures, we may speak of *lower docks*, those incident with α, and *upper docks*, incident with β. The vertices in the docks, except α and β, are called *dock vertices*. The two sets of upper and lower docks will sometimes be referred to as the upper and lower *shore*, respectively. We distinguish two types of docks, which have to be treated very differently. Call a dock *closed* if its dock vertices are connected in K, otherwise call it *open*.

Figure 7 gives an overview. It displays a hypergraph with four upper and four lower docks. Connections between docks being indicated as mere paths, though they can, in principle, be arbitrarily complicated, of course. As in most figures in this section, we omit the 2-edge ϕ between α and β from the drawing for graphical reasons.

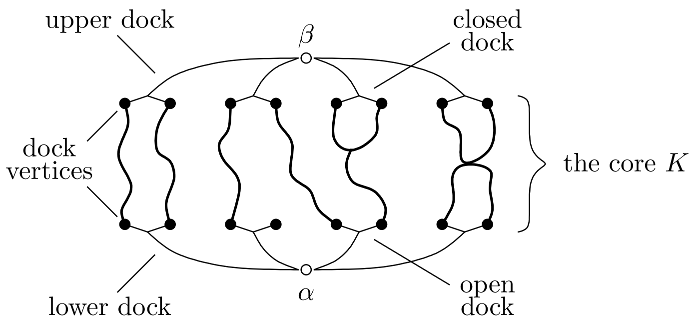


FIGURE 7. A schematic picture of docks and core.

To decide whether a hypergraph arranged as above is a winner or a loser, we take the following approach. Throughout this section we make the general assumption that the hypergraph H at hand is a loser and try to rule out configurations that would conflict with this assumption because they yield a Maker win. Eventually, we shall find that only a few connection types between the docks are possible. After that, in the next section, we shall prove that our classification is valid, i.e., none of the left-over configurations can be won by Maker.

We begin our analysis on a global scale. Our first observation accounts how many docks of what type can be connected to a single open or closed dock.

26. Observation. *In the core K of a loser H, no two different docks from the same shore are connected. A closed dock is connected to exactly one dock on the other shore. Each dock vertex of an open dock is either connected to one dock on the other shore or to no docks at all, but at least one of them is connected to another dock.*

Proof. The first statement is the basic observation, which then implies the others. Assume for contradiction that two different lower docks $e = \{\alpha, a, a'\}$ and $f = \{\alpha, c, c'\}$ are connected in K, i.e., there is a path from a or a' to c or c'. Pick a shortest such path P and change vertex labels if necessary, to have P going from a to c; this guarantees that none of a' and c' are touched by P. (Note that $a = c$ is possible.) See Figure 8. Maker can win by playing at α because by Lemma 15 this move requires an immediate answer in the cycle $\alpha e a P c f \alpha$ but Breaker must also destroy the now singleton edge $\{\beta\}$.

The rest is an easy implication of the above. Every dock must be connected to at least one other dock to make H Maker-2-connected and in each case a connection to some further dock would induce a connection between docks from the same shore. □

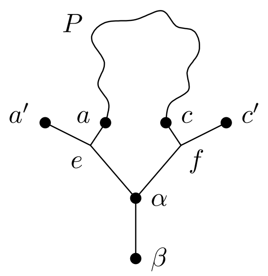


Figure 8. Two connected lower docks yield a win.

Figure 7 already contained schematic representations of all dock connections allowed by Observation 26.

In the following, we investigate the local structure of the different connection types between docks: open-open, closed-closed, and closed-open. In each case, we face a lower dock $g = \{\alpha, a, c\}$ and an upper dock $h = \{\beta, b, d\}$ that are somehow connected in the core K of H. As in Observation 26 above, we always make the general assumption that the whole hypergraph H is a loser.

Between two open docks. The situation between vertices from two open docks is very simple.

27. Observation. *Let the docks $g = \{\alpha, a, c\}$ and $h = \{\beta, b, d\}$ both be open, with a and b connected in the core K. Then the connected component of K that contains a and b is simply a path between these two vertices that contains no further dock vertices.*

Figure 9 visualizes Observation 27. We postpone the proof for a second to discuss a general issue. Almost all arguments throughout this section require us to pick inner vertices on paths that lie "between" certain given

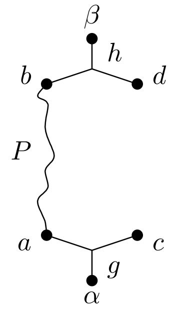


FIGURE 9. Two open docks connected by a path.

vertices. While such a notion would be clear for ordinary graphs, we should make it precise for our hypergraphs.

28. DEFINITION. For two distinct vertices u, v of a path or cycle P, we say that some other vertex x lies *between* u *and* v *on* P if x is an inner vertex of the subpath uPv or $x = u$ or $x = v$.

As an example, we have marked the vertices between u and v on the path on the left-hand side of Figure 10 with circles. We will use this concept in situations where there exists some other path Q from u to v, with Q disjoint from P except for the terminal vertices u and v. Then a vertex between u and v on P will be an inner vertex of the cycle $uPvQu$.

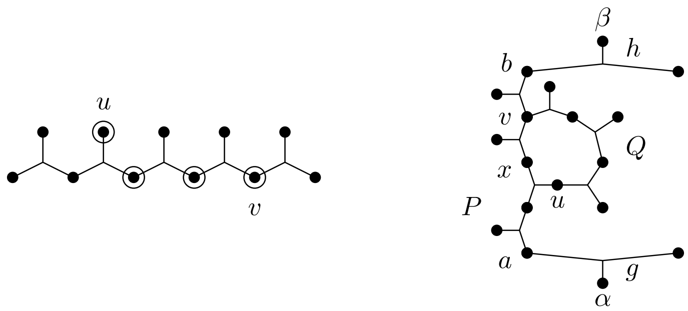


FIGURE 10. An example path with all vertices *between* two vertices u and v marked (left) and an extension of Figure 9 by another path (right).

The main ingredient for the proof of Observation 27 is Lemma 23, which we use here for the first time. It is the technical tool to provide us with the intuitively obvious fact that if we add any further edge to the hitherto constructed part of H between two docks, there will be a whole new path between two distinct vertices of this subgraph because H is Maker-2-connected. We will see this argument repeatedly in the following and we give it here in great detail as a general example.

PROOF OF OBSERVATION 27. Pick any path P from a to b as shown in Figure 9. Maker will use the cycle $C = \alpha gaPbh\beta\phi\alpha$ (i.e., the path P closed to a cycle by the two docks and the 2-edge ϕ) to set up threats against

Breaker. To show that no further edges are incident to vertices of P, we assume for contradiction that some edge $e \in E(K) \setminus E(P)$ is connected to P.

In the general case, when e contains a vertex $z \notin V(P)$, we apply Lemma 23 as follows. The middle layer M in that lemma is $V(P)$. The bottom layer B consists of all vertices of H that are connected to α and β in $H - P$, i.e., it contains α, β, and the vertices between all the other docks of H. The top layer T is the rest $V(H) \setminus (M \cup B)$, which is not empty because we have $z \in T$. Now Lemma 23 tells us that there is a path Q in K that connects two distinct vertices u, v of P and contains no further vertices of P.

Between the vertices u and v on P we find an inner vertex x of our cycle C. (We refer to the inner vertices of C rather than those of P because we need to include the end vertices a and b as well.) The right-hand side of Figure 10 shows a concrete example where one of u and v is an outer vertex of P and the other an inner. A suitable x is found between them. This x is clearly also an inner vertex of the cycle $D = xPuQvPx$.

In the special case $e \subseteq V(P)$ we do not need the Lemma 23 for path finding, of course. Simply pick two vertices u, v of e that are closest to each other on P. Again there is an inner vertex x of C between u and v, which is also an inner vertex of the cycle $D = xPuevPx$.

In any case, Maker wins by playing at x because Lemma 15 restricts Breaker's reply to the cycle D, while Lemma 16 requires a move in the 2-edge ϕ of C, which does not touch D. $\square$

Since Observation 26 leaves the possibility that one of the two dock vertices of an open dock is connected to no other dock vertex as long as the other one is, we must note this simple case, too.

29. Observation. *If a dock vertex of an open dock is connected to no other dock vertices then it is not incident to any edge of K.*

Proof. If some K-edge was connected to such a vertex, this vertex would be an articulation point of H, in contradiction to Maker-2-connectedness. $\square$

Between two closed docks. The situation of two closed docks connected in K is considerably more complicated to analyze than the previous case of two open docks. A waterproof discussion requires the investigation of many potential configurations. In the end, however, we shall see that all but one simple arrangement can be excluded because they would lead to immediate Maker wins.

Let us begin with the construction of the objects that we know must be there. Pick two paths A and B in K, the former from a to c and the latter from b to d. See Figure 11 for two concrete example configurations. We do not require, nor can we prove, disjointness of A and B but we know these paths cannot intersect too deeply. As it turns out, the vertex x in the left example from the figure already leads to a Maker win.

30. Observation. *The paths A and B cannot share a vertex that is at the same time an inner vertex of A or one of the two dock vertices a and c, and an inner vertex of B or one of the two dock vertices b and d. In particular, the docks g and h do not intersect.*

Proof. Assume there exists such a common vertex and pick such an x, if possible one of the dock vertices. We show that Maker wins by playing at x. If x is an inner vertex of A, we have the two paths $P_a := xAag\alpha\phi\beta$ and $P_c := xAcg\alpha\phi\beta$ (see left-hand side of Figure 11) in each of which Breaker must answer. So Breaker must play a vertex in $g \cup \phi$. If $x = a$ or $x = c$, Lemma 12 forces Breaker to answer in the same set. A symmetric argument for the upper shore shows that Breaker must also play in $h \cup \phi$.

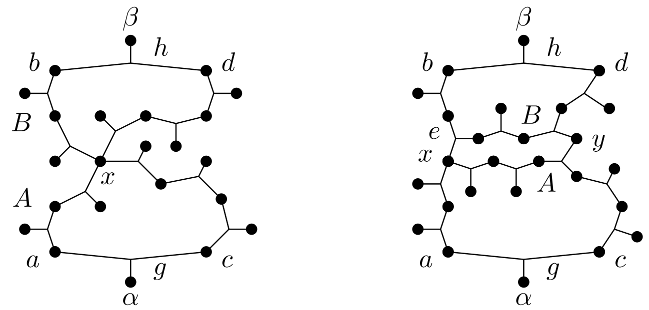


Figure 11. The two connecting paths A and B touching in a common inner vertex (left) and touching in two different vertices (right).

If g and h do not intersect then this already tells us that Breaker can only play at α or β. If g and h do intersect then their intersection is by almost-disjointness only one vertex, which by our choice must be x and is thus already taken by Maker. Therefore, Breaker is restricted to play at α or β in this case, too. In addition to this, Lemma 15 requires an answer in each of the two cycles $xAagcAx$ and $xBbhdBx$, whose intersection does not contain α and β. So Maker wins. □

Note that Observation 30 also excludes the possibility that the paths A and B share an edge because at least one vertex of such an edge would be an inner vertex in both paths (or dock vertex). This tells us that A and B cannot overlap too much. We now show that they cannot even intersect in two vertices.

31. Observation. *The upper and lower path, A and B, share at most one vertex.*

Proof. Assume for contradiction that the two paths intersect in more than one vertex; we show that this gives a Maker win. Pick a shortest path P in $A \cup B$ from g to h. By symmetry we may assume that P goes from $a \in g$ to $b \in h$ and then minimality implies that $c, d \notin V(P)$. Hence, $C = \alpha gaPbh\beta\phi\alpha$ is a cycle.

Starting at a, we walk along C into the core until we enter the first edge e that does not lie in A. (In an extremal case, e might actually be the dock edge h.) Denote the inner vertex of C that came just before e by x; clearly $x \in V(A) \cap V(B)$. Note that we do not claim that x be an inner vertex of A or B. Compare the right drawing of Figure 11.

Let $y \neq x$ be a further contact point of A and B such that the total length of the paths xAy and xBy is minimal. Observation 30 implies that these two subpaths share no edges and therefore, by minimality and almost-disjointness, the composition $D = xAyBx$ is a cycle (not self-touching) which clearly has x as an inner vertex. We have constructed two cycles, C and D, which share x as an inner vertex. Maker plays at x. By Lemma 16 Breaker must answer at α or β and by Lemma 15, he must play somewhere in D, but these vertex sets are disjoint. □

In case that A and B do not touch at all, we now extend our construction by a connecting path. Let F be a shortest such path from any vertex u of A to any vertex v of B. (See the right-hand side of Figure 12 for an example.) Note that in contrast to ordinary graphs, the minimality of F does not guarantee that F contains no further vertices of A or B. So we have to prove this property.

32. Observation. *The connecting path F touches A and B each in only one vertex.*

Proof. Assume for contradiction that F touches B in two vertices, u, v, say. If one of these vertices is a dock vertex, let x be this dock vertex. (By almost-disjointness there can be only one.) Otherwise there lies at least one inner vertex of B between u and v; let x be such an inner vertex then. See the left-hand side of Figure 12.

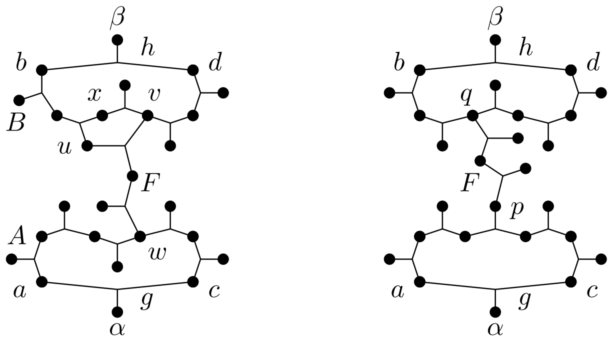


Figure 12. The connecting path F touching B in two places (left) and touching an inner vertex of B (right).

Maker plays at x. If x is a dock vertex then Lemma 12 requires an answer in the path $xh\beta\phi\alpha$. Otherwise we have the two paths $xBbh\beta\phi\alpha$ and $xBdh\beta\phi\alpha$ in both of which Breaker must play, which leaves the same replies b, d, β, α. By Lemma 15, Breaker must also answer in the cycle $xBuFvBx$ since it contains x as an inner vertex. Together, even in the best case for Breaker, when b and d both lie in that cycle, he is left with no more replies than b and d.

There are further threats on the lower side. We have the two paths $xBuFwAsg\alpha\phi\beta$ and $xBvFwAsg\alpha\phi\beta$, where w is a contact vertex of F and A, and s is either a or c. Lemma 12 forces Breaker to play in both paths but their intersection clearly contains none of b and d; hence Maker wins. □

We now know that F connects exactly one vertex p of A to one vertex q of B. (Where the case that A and B touch is included as the degenerate case where F has length 0 and consists of just one vertex $p = q$.) See Figure 13. We can say even a little bit more. The contact points p and q cannot be arbitrary vertices of A and B. Only outer vertices, as drawn in the figure, are allowed.

33. OBSERVATION. *The vertices p and q are outer vertices of A and B, respectively.*

PROOF. Assume for contradiction that one of them, q, say, is an inner vertex, of B. See the right-hand side of Figure 12. We assume by symmetry w.l.o.g. that c is no closer to p than a so that $qFpAag\alpha\phi\beta$ is a path (i.e., does not use a vertex twice).

Maker plays at a. Then Lemma 12 requires an answer in the path $ag\alpha\phi\beta$ and Lemma 15 one in the cycle $aAcga$. Therefore Breaker must play at α or c. Maker's next move is at q. It lies on the path $(qFpAa)^{[+a]}$ and is an inner vertex of the cycle $qBbhdBq$. Lemma 12 and Lemma 15 require an answer in the path and the cycle, respectively, and since these substructures intersect only in Maker's vertex q, Maker wins. □

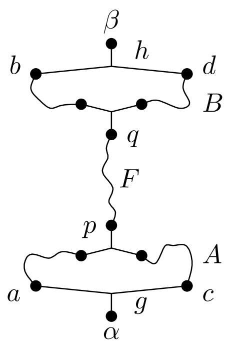


FIGURE 13. The final configuration between two closed docks consisting of the three paths A, B, and F.

We are almost done. It remains to show that Figure 13 is complete.

34. OBSERVATION. *There are no further edges in the core K touching any vertex of the three paths A, B, and F.*

PROOF. Let $M := A \cup B \cup F$. We assume for contradiction that there exists some further edge $e \in E(K) \setminus E(M)$ that contains a vertex of M. If e contains also some vertex outside of M, we can apply Lemma 23 to obtain a path P in K connecting two vertices $u, v \in V(M)$ and containing no other vertex of M. In the degenerate case, when $e \subseteq V(M)$, we pick two vertices $u, v \in e$ such that the unique path from u to v in M does not contain the third vertex w of e.

Denote by Q the unique path from u to v in M, precisely, Q is of the form uAv, uBv, uFv, $uApFv$, $uBqFv$, or $uApFqBv$, depending on the locations of u and v. Together with the path P, respectively the edge e, this path forms a cycle $C = uQvPu$ respectively $C = uQveu$ in K. Next we pick a shortest path R in M from the lower dock to the upper dock, w.l.o.g. $R = aApFqBb$. Minimality guarantees that this path does not contain the other two dock vertices c and d, so that the composition $D = \alpha g a R b h \beta \phi \alpha$ is a cycle which contains the 2-edge ϕ. The left-hand side of Figure 14 shows what we have constructed so far.

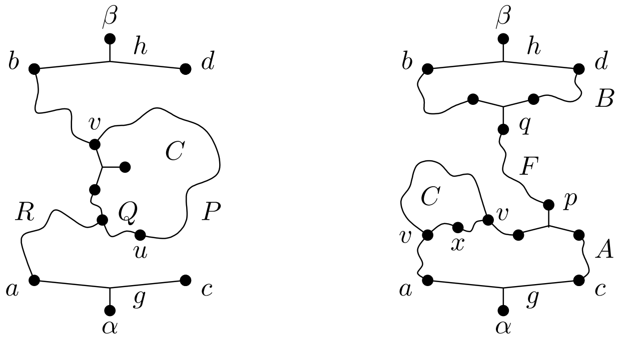


FIGURE 14. Cycle constructions for the proof of Observation 34.

We now have to distinguish the different types of Q. If the cycle C contains an edge of F then this edge contains a vertex x that is inner to both cycles. As in many situations before, Lemmas 15 and 16 then show that if Maker plays at x, Breaker has no reply to the threats of the two cycles C and D, so he loses.

The situation is similarly easy for Maker if $u, v \in V(A)$ and the subpath uAv contains the contact point p (the case $u, v \in V(B)$ being completely symmetric to this). Then the cycle C again shares an edge with R, namely the one edge of A that contains the vertex p. So Maker wins at a vertex in this edge.

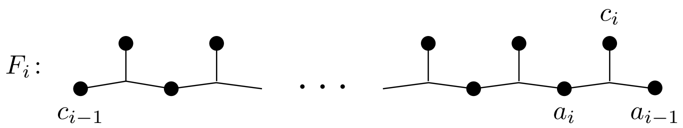
The only remaining configuration is one that has u and v on the same side of p on the path A, as depicted in the right drawing of Figure 14. Between u and v lies an inner vertex x of A ($x \in \{u, v\}$ being allowed) and this x is clearly also an inner vertex of C. We claim that Maker wins at x. Consider the two paths $P_1 = xAag\alpha\phi\beta$ and $P_2 = xApFqBbh\beta\phi\alpha$. Lemma 12 requires a reply in their intersection, the 2-edge ϕ plus possibly the dock vertex c. The cycle C, in which Breaker must also play, contains none of these vertices, so Maker wins. $\square$

This concludes the analysis of the core between two closed docks. It must look exactly as shown in Figure 13.

Between a closed and an open dock. To analyze the core between a closed and an open dock, we cannot proceed as in the previous cases. If we started with a few basic connections and then added new paths provided by Lemma 6, trying to sort out winning configurations, we would never reach an end. As we shall see, there exists an infinite family of topologically different core types. So we have to take a different approach, which unfortunately comes not as naturally as the incremental one. We first present a uniform class of hypergraphs—without further motivation—and afterwards prove that the core between a closed and an open dock must come from this class.

35. Definition. A 3-uniform hypergraph L is called *a ladder of height* $h \geq 0$ *on* a_0 *and* c_0 if it can be constructed by the following procedure:

- begin with the empty hypergraph $L_0 = (\{a_0, c_0\}, \emptyset)$ on two vertices a_0, c_0;
- `for` $i = 1, \dots, h$ `do`
  (if $h = 0$ then simply skip the loop)
  - take a new path F_i of length ≥ 2 with start vertex c_{i-1}, end vertex a_{i-1} (which are both vertices of L_{i-1}) and no further vertices common with L_{i-1};
  - denote the last inner vertex of F_i by a_i and the last outer vertex different from a_{i-1} by c_i; as shown in this figure:



    the vertices a_i and c_i will be the contact points for the next path F_{i+1};
  - let $L_i := L_{i-1} \cup F_i$;
- `either` end the construction of L by letting $L := L_h$
  `or` take an optional additional path R from c_h to some vertex r of the path $c_{h-1}F_h a_h$ except a_h (but $r = c_{h-1}$ allowed) that contains no further points of L_h and let $L := L_h \cup R$.

Figure 15 shows a ladder of height 4. The dotted bubbles indicate level sets, defined as follows. The ith level, $1 \leq i \leq h$, consists of the set $V(F_i) \setminus \{a_{i-1}, c_i\}$, i.e., the vertices of the path $c_{i-1}F_i a_i$. On level 0 lies only the vertex a_0; and the remaining vertices at the top of L, which are exactly those in $V(R) \setminus \{r\}$ or only the single vertex c_h, in case the optional path R is not present, form the highest level $h + 1$.

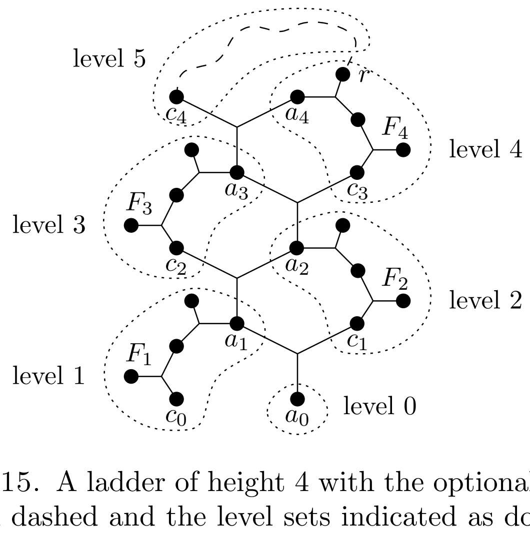


Figure 15. A ladder of height 4 with the optional top path R drawn dashed and the level sets indicated as dotted bubbles.

We let the highest level of a ladder be one above its formal height because we like to think of the vertex c_h and the optional path R as parts that do not belong to the regular structure. This convention shall turn out convenient.

The introduction of ladders is motivated by the next observation, which describes the closed-open case completely. We still face two docks $g = \{\alpha, a, c\}$ and $h = \{\beta, b, d\}$ in a hypergraph H, which we assume to be a loser. This time, g shall be closed and h shall be open, with a and c connected to b in the core K. Let J denote the connected component of K that contains the dock vertices a, b, c, extended by the vertices α and β and the dock edges g and h. We can now put all we have to say about J in one brief statement.

36. OBSERVATION. *The hypergraph J is a ladder on $a_0 = \alpha$ and $c_0 = \beta$. Its height is at least 1, and at least 2 if it does not contain the additional path R.*

Figure 16 shows such a ladder on α and β with the two contained docks arranged in the way we usually draw them.

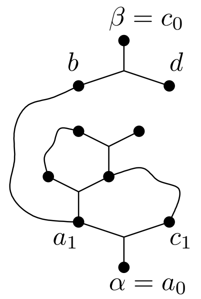


FIGURE 16. A ladder of height 3 connecting a closed lower and an open upper dock.

In order to allow the rather long and technical proof of Observation 36 to focus on the basic ideas, we prepare the main technical tools separately in advance. Like in the open-open and closed-closed case, we will argue that if J contains any further edges not in L then the whole hypergraph H must be a winner. Therefore we again need a suitable set of paths that end in a 2-edge and can thus be used as threats against Breaker. For the present open-closed case, we shall make repeated use of certain paths that connect some vertex x somewhere up in the ladder to one of the base vertices a_0 and c_0, which we define recursively as follows.

For a level-1 vertex x let

$$P_a(x) = xF_1a_0 \qquad \text{and} \qquad P_c(x) = xF_1c_0$$

denote the shortest path from x to the respective base vertex. For x on a level j with $2 \le j \le h$, let

$$P_a(x) = \begin{cases} xF_jc_{j-1}P_a(a_{j-1}) & \text{if } j \text{ even,} \\ xF_ja_{j-1}P_a(a_{j-1}) & \text{if } j \text{ odd} \end{cases}$$

and

$$P_c(x) = \begin{cases} xF_ja_{j-1}P_c(a_{j-1}) & \text{if } j \text{ even,} \\ xF_jc_{j-1}P_c(a_{j-1}) & \text{if } j \text{ odd.} \end{cases}$$

These somewhat cumbersome definitions describe rather simple geometrical objects: two kinds of paths that climb down the ladder on its left and its right rail. The paths P_a all head for a_0 while the P_c aim for c_0. The parity conditions simply take care of the alternating orientations of the paths F_i: $P_a(x)$ goes through the a_i with even i and through the c_i with odd i; for $P_c(x)$ vice-versa. Figure 17 depicts the two complementary paths $P_a(x)$ and $P_c(x)$ for a level-4 vertex x (compare to Figure 15). The path $P_c(x)$ starts from x along F_4 to its left end, from where it descends down the ladder along the left rim. Likewise, the path $P_a(x)$ climbs down the right-most edges of the ladder.

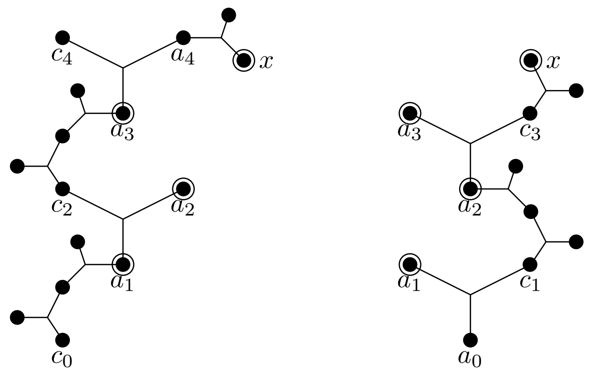


FIGURE 17. The paths $P_c(x)$ (left) and $P_a(x)$ (right) for a vertex x on level 4 of the ladder from Figure 15. Common vertices marked.

The following property makes the paths P_a and P_c useful for Maker.

37. LEMMA. *Let x be a level-j vertex of some ladder of height h. If x is an inner vertex of F_j or its starting vertex c_{j-1} then the two paths $P_a(x)$ and $P_c(x)$ intersect in no vertices other than x and all a_i with $1 \leq i < j$.*

PROOF. From their starting point x on F_j the two paths in consideration head in opposite directions. (Note that in the case $x = c_{j-1}$ this is guaranteed because the level of this vertex was defined to be j, not $j - 1$.) Once the two paths enter F_{j-1}, they stay on opposite sides of the ladder as far as possible. Hence, they can only intersect in the middle vertices a_i that lie below. □

The paths P_a and P_c shall now be used to derive Maker wins for any configuration that deviates from a ladder shape.

PROOF OF OBSERVATION 36. Pick any inclusion-maximal ladder L on $a_0 = \alpha$ and $c_0 = \beta$ in J, which will have height at least 1 because any path from β to α can serve as the path F_1. We do not demand that L have greatest possible height but only that we cannot extend it with J-edges to a larger ladder. It might be helpful to convince oneself that this means exactly that either L contains the optional path R—which in a way seals off the top part of L—or that there is no additional path from c_h to any other vertex of F_h; although formally, this fact shall not be needed in this proof.

So assume for contradiction that $J \supsetneq L$. As before we either employ Lemma 23 to get a J-path P between two distinct vertices of L or, in the degenerate case, we find a single edge $e \in E(J) \setminus E(L)$ with $e \subseteq V(L)$. Let

j be the lowest level touched by P respectively e. We distinguish different possible contact configurations.

If the second contact point of the path P lies also on level j or, in the degenerate situation, if at least one further vertex of the additional edge e does, then Maker wins as follows. Denote the two contact points of P and L by u and v. In the degenerate case, pick $u, v \in e$ such that the third vertex of e does not lie between u and v on the path F_j (respectively R, if $j = h + 1$). Then there exists an inner vertex x of F_j respectively R between u and v. See Figure 18. This x is an inner vertex of the cycle $C = xF_juPvF_jx$ respectively $C = xRuPvRx$, which in either case contains no vertices on levels strictly less than j. (Observe that calling c_j a level-$(j+1)$ vertex was again necessary to guarantee that the cycle C cannot use the edge $\{a_{j-1}, a_j, c_j\}$.) For the case $j = 1$ we note that C does surely not contain c_0 because otherwise it would include the upper dock, making it a closed dock.

Now Maker plays at x. By Lemma 15, Breaker's reply must be in C but Lemma 12 prompts for an answer in each of the paths $P_a(x)a_0\phi c_0$ and $P_c(x)c_0\phi a_0$. By Lemma 37 the intersection of these two paths and the cycle C contains no vertex other than x, so Maker wins.

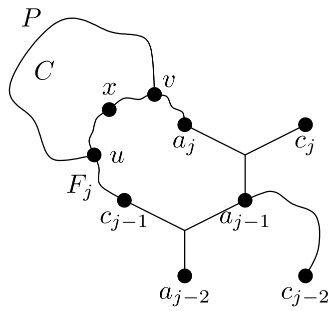


FIGURE 18. Both contact points u and v on level j.

Our analysis of the situation where there is only one contact point, u, say, on the lowest contact level $j \geq 1$ splits into two cases. First the general case: $j < h$. The union of all paths F_i with $i > j$ together with the path R (provided it is present), i.e., the induced subhypergraph of L on all vertices on levels above j and the vertex a_j, forms a connected subhypergraph M of L, shown in Figure 19. Pick a shortest path Q in M from a_j to v, the second contact point of the new path P (respectively e) and L, which must lie in M because u is the only contact on level j. If there is a third contact point w, relabel v and w if necessary, such that v lies no farther from u than w, so that by almost disjointness w does not lie on Q. We obtain a cycle $C = a_jQvPuF_ja_j$ with a_j as an inner vertex. By construction, C contains no vertices strictly below level j. Maker plays at a_j. Just like above, Breaker is forced to answer in C but also in each of the two paths $P_a(a_j)a_0\phi c_0$ and $P_c(a_j)c_0\phi a_0$, whose intersection contains no vertex of C, except a_j, of course. Hence Maker wins.

It remains to consider the case $j = h$. ($j = h + 1$ is impossible because that would leave no higher levels for the second contact point.) First observe

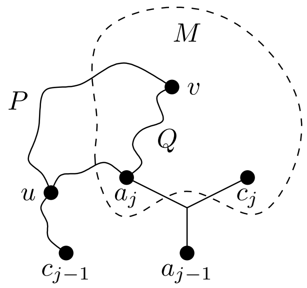


FIGURE 19. Second contact point v on a higher level.

that L surely contains the optional path R since otherwise the new path P (or the edge e) would have to connect to the only level-$(h+1)$ vertex c_h, forming such a path R itself, thereby contradicting the maximality of L. We know that P (resp. e) connects $u \in V(F_h) \setminus \{a_{h-1}, c_h\}$ to some $v \in V(R) \setminus V(F_h)$. See Figure 20. A possible further contact point w would also have to lie in this set, in which case we assume w.l.o.g. that v come before w on rRc_h, so that w does not lie on the path rRv.

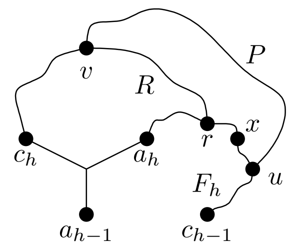


FIGURE 20. Contact points on levels h and $h+1$.

If $u = r$ then Maker wins easily at u because this is then an inner vertex of the cycle $uRvPu$, which intersects at least one of the paths $P_a(u)$ and $P_c(u)$ only at x (depending on the parity of h). Note that u need not be an inner vertex of F_h for this to work. So we are left with the case $u \neq r$. Between u and r on F_h we find an inner vertex x of F_h ($x = u$ and $x = r$ being allowed). This x is also an inner vertex of the cycle $xF_huPvRrF_hx$, which contains no vertices strictly below level h. Like we argued before, Maker wins by playing at x because Breaker cannot play in this cycle and the two paths $P_a(x)a_0\phi c_0$ and $P_c(x)c_0\phi a_0$ at the same time.

This eventually shows that our assumption $J \supsetneq L$ must be false. The additional statements about the height of L follow immediately from the fact that the lower dock is closed. □

5. Playing for Breaker

The classification into different connection types in the core started from the assumption that the whole hypergraph at hand was a loser. We do not know yet, whether any hypergraph with one 2-edge whose core uses only those connections singled out in the previous section, could perhaps be a winner. We settle this issue by proving the open implication of the following theorem.

38. Theorem. *An almost-disjoint Maker-2-connected hypergraph with only 3-edges except exactly one 2-edge is a loser if and only if its core connections are of the following three types:*

- *between two open docks there is only a path as described in Observation 27 and shown in Figure 9 on page 44,*
- *between two closed docks the connection satisfies all properties stated in Observations 30 through 34 as shown in Figure 13 on page 48,*
- *between a closed and an open dock the connection is a ladder as stated in Observation 36 and indicated in Figure 16 on page 51.*

Elementary losers. Essentially, the task in this section will be to prove that certain hypergraphs, usually subhypergraphs of the given hypergraph at hand, are losers. Besides some side remarks along our discussion of winning paths and cycles in Section 2, we have by now not really proven any hypergraphs losers. So let us start by collecting some necessary basic facts, again about paths and cycles.

39. Lemma. *Any almost-disjoint 3-uniform path P is a loser. Moreover, even $P^{[+u]}$ is a loser for any vertex $u \in V(P)$.*

Proof. It suffices to prove the second, stronger statement; by induction. Let v be any Maker move in $P^{[+u]}$. Breaker can always separate u and v in the following way. If u and v do not lie in a common edge of P, Breaker plays a vertex y between them. (For example, in Figure 10 on page 44, y would be one of the two marked vertices between u and v.) Otherwise he plays the third vertex y in the edge that contains u and v.

The hypergraph $P^{[-y]}$ is then the disjoint union of two paths, where u lies in one component and v in the other. Each of those components are losers by induction and consequently, the whole graph $P^{[+u,+v,-y]}$ is a loser by Lemma 17. A length-zero path with just one vertex is trivially a loser because it contains no edges that Maker could fill. □

40. Lemma. *An almost-disjoint cycle of 3-edges is a loser. Even more, it remains a loser if we replace one 3-edge by a 2-edge.*

Proof. It obviously suffices to prove the second statement. (The right-hand side of Figure 4 on page 36 showed how a cycle with one 2-edge can be interpreted as a 3-uniform cycle with a Maker play at an outer vertex.) Irrespective of where in the cycle Maker plays his first move x, Breaker always takes one vertex of the 2-edge, destroying that edge. The resulting hypergraph can be interpreted as a path of 3-edges in which Maker has played one vertex, x. Hence it is a loser by Lemma 39. □

Typical applications of Lemma 39 will be configurations in which some path is only connected through a single vertex to the rest of the hypergraph. In such a situation, the Articulation Lemma tells us that we can either remove that path completely or, if it already contains a Maker move, replace it by another Maker move at the contact point. The precise conditions are captured by the following corollaries to Lemma 39.

41. Corollary. *Let $H = P \cup B$ be the union of an almost-disjoint 3-uniform path P and an arbitrary hypergraph B that have exactly one point in common. Then H is a winner if and only if B is.* $\square$

42. Corollary. *Let P be an almost-disjoint 3-uniform path and B be an arbitrary hypergraph, such that $V(P) \cap V(B) = \{p\}$ for some articulation vertex p. Let furhter x be any vertex of P. Then the union $H = P^{[+x]} \cup B$ is a winner if and only if $B^{[+p]}$ is.* $\square$

Since ladders play an important role in our classification, we shall need losing conditions for them, too.

43. Lemma. *A ladder on a 2-edge is a loser.*

44. Lemma. *Let x be a vertex on the 1st level of a ladder L on a_0 and c_0. Then the hypergraph $L^{[+x,-a_0]}$ is a loser.*

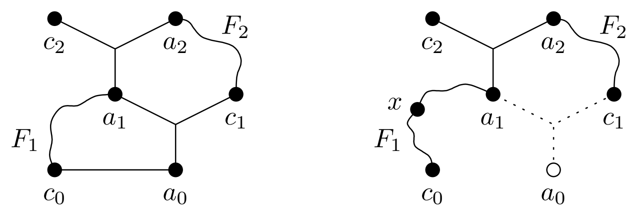


Figure 21. The ladder configurations of Lemma 43 (left) and Lemma 44 (right).

Figure 21 shows the respective configurations of these lemmas. The two statements are closely related. We prove them together by an interleaved induction.

Proof of Lemmas 43 and 44. We parameterize the lemmas by the height: $\mathcal{A}(h)$ denote the statement of Lemma 43 restricted to ladders of height h and $\mathcal{B}(h)$ denote the statement of Lemma 44 restricted to ladders of height h. We perform a mixed induction on h by reducing $\mathcal{B}(h)$ to $\mathcal{A}(h-2)$, and $\mathcal{A}(h)$ to $\mathcal{A}(k)$ and $\mathcal{B}(\ell)$ with $k < h$ and $\ell \leq h$. Note that this avoids circular arguments although $\mathcal{A}(h)$ may use $\mathcal{B}(h)$, because $\mathcal{B}(h)$ does not rely on $\mathcal{A}(h)$.

Induction bases. Since a height-0 ladder on a 2-edge is just that 2-edge, $\mathcal{A}(0)$ is obviously true, and $\mathcal{B}(0)$ is true simply because the respective hypergraph does not contain any edges on which Maker could win. Let us also treat $\mathcal{B}(1)$ at this point to take care of some irregularities which result from the path R. If the optional path R is not present, the hypergraph is just the path F_1 with one vertex played, a loser by Lemma 39. If R is present, we can simply remove it because Breaker's move a_0 has destroyed the second contact point c_1 of R and F_1. So we get the same loser as before. (Though Figure 21 shows the regular path F_2 instead of the path R, one can still see there that the rightmost path can be deleted because of Breaker's move at a_0.)

The induction step for $\mathcal{B}(h)$, $h \geq 2$, works similarly. We use Corollary 42 to replace the path F_1 by a single Maker move at a_1. Then we delete the dangling path F_2 (see right of Figure 22) by Corollary 41. What's left is a ladder of height $h-2$ on the new 2-edge $\{a_2, c_2\}$.

Induction step for $\mathcal{A}(h)$, $h \geq 1$. If Maker plays his first move x on level 0, i.e., $x = a_0$, then Breaker answers at c_0. We can then delete the path $c_0 F_1 a_1$ (or a slightly shorter path up to r if $h = 1$ and R is present). This leaves a ladder on the 2-edge $\{a_1, c_1\}$, a loser by induction.

If Maker's first move x is on level h or $h+1$, Breaker answers at a_{h-1}. This disconnects levels h and $h+1$ from all lower levels. See the left-hand side of Figure 22. If x lies on level h, the top part is a loser by $\mathcal{B}(1)$ and if x lies on R then we can remove most of F_h so that the rest of the top part is a loser by Lemma 39. The lower part is (after removal of the dangling path F_{h-1}) a ladder of smaller height, hence also a loser by induction.

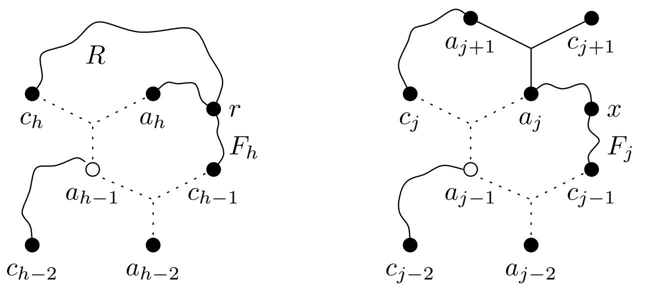


FIGURE 22. Maker plays x on level h or $h+1$ (left) and Maker plays on an intermediate level $j < h$ (right).

We turn to the general case: Maker x on a level j with $1 \leq j < h$. In this situation Breaker plays a_{j-1}. See the right-hand side of Figure 22. As in the previous situation, the ladder breaks up into a lower and an upper part, the former again (after removal of the dangling path F_{j-1}) being a shorter ladder on the 2-edge $\{a_0, c_0\}$, a loser by induction. The upper part can be interpreted as a ladder on a_{j-1} and c_{j-1} with a_{j-1} already played by Breaker and the vertex x (now on level 1) already played by Maker. A loser by induction. □

Almost all arguments during our classification in Section 4 were in a sense written out of Maker's perspective. Usually, we proved that some configuration cannot occur in a loser by presenting a winning strategy for Maker. The case distinctions were set up in such a way that in each step we could derive a Maker win with very few explicit moves—often just one—by listing several threats in the form of paths and cycles, that could not all be countered by Breaker at the same time.

The present situation is very different. We want to show that Maker cannot win on certain hypergraphs. So we pick good Breaker moves and must, in principle, provide counters against *all* possible Maker attacks. The obvious problem here is: Breaker has no threats; by the very definition of the game.

In the proof of the two preceding ladder lemmas, we could exploit the strong symmetry of ladders, which allowed an induction. The question now is: How to get control over all possible Maker strategies on the whole hypergraph H? The key is again the central role of the 2-edge ϕ. If we manage to get a Maker or Breaker move into that edge, the hypergraph will lose its Maker-2-connectivity. Precisely, if β is taken then α becomes an articulation vertex, which makes the hypergraph amenable to an application of the Articulation Lemma to break it into smaller parts. The resulting components will then be simple enough to be analyzed by the above lemmas about paths, cycles, and ladders.

The basic components. Let us collect all such components that arise when Breaker plays one vertex of the 2-edge ϕ, at β, say. Precisely, we list all types of hypergraphs M such that $H^{[-\beta]}$ can be written as a union $M \cup D$ with $V(M) \cap V(D) = \alpha$ and such that α is not an articulation vertex of M, i.e., we only consider minimal components.

First observe that such a component M contains no more than 3 docks because any lower dock g is connected to at most two upper docks and in $H^{[-\beta]}$ any upper dock vertex is connected to at most one lower dock. Closed upper docks have unique lower partners anyway and all open docks are destroyed at β so that they no longer link their partners on the lower shore.

Out of the three connection types from the previous section, we assemble again three essentially different types of such components M.

(i) Two connected closed docks, where the upper dock has been destroyed. See the upper left of Figure 23.
(ii) A closed lower dock connected to an open upper dock. This is simply a ladder, shown on the upper right of Figure 23.
(iii) An open lower dock between two closed upper docks, which both have their base point $a_0 = \beta$ deleted. This is the union of two ladders with the base point a_0 deleted in each, glued together on the first edge of their F_1-paths. See the lower part of Figure 23.

The remaining possibilities of an open lower dock between two open upper docks or one open and one closed upper dock, or an open lower dock linked to just one upper dock, can be interpreted as subhypergraphs of configurations covered by case 3 since a path to an open dock can be seen as the first level of a ladder. So we omitted them from the above list since it will suffice just to observe that all relevant properties of components of type (iii) will carry over to them.

The base case. Our analysis of possible Maker moves begins with the easiest situation, where Maker takes α and Breaker gratefully answers at β so that afterwards everything is nicely decomposed. Although this is a very special case, it forms the basic result to which we shall later reduce all the other possible Maker plays.

45. Observation. *If in the first move each player takes one vertex from the 2-edge ϕ, the game is lost for Maker.*

Proof. To go conform with the above classification, we assume by symmetry that Maker has played at α and Breaker has answered at β. We now simply go through our list and verify for each type whether $M^{[+\alpha]}$ is a loser.

Case (i). Two closed docks. Maker's move has produced a 2-edge in the lower cycle. Two applications of Corollary 41 remove the upper cycle

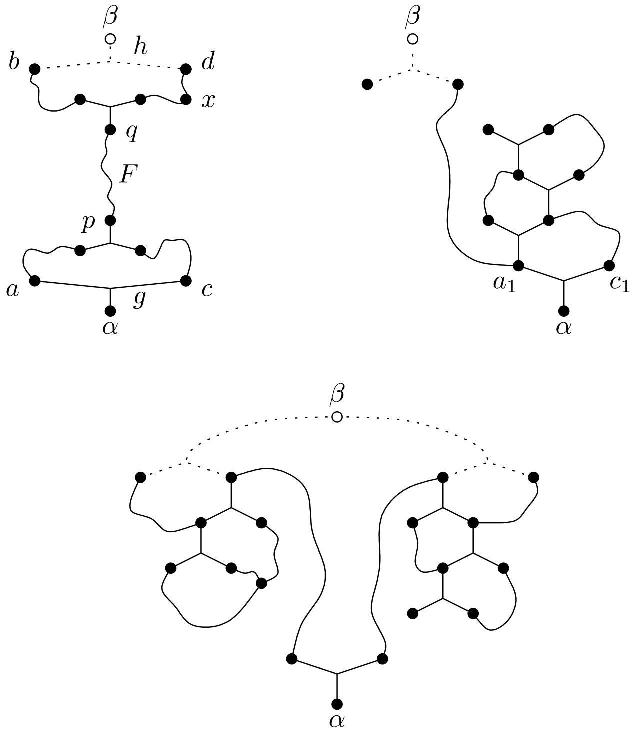


FIGURE 23. Components of $H^{[-\beta]}$.

entirely, together with the path in the middle, leaving only the lower cycle which is lost by Lemma 40.

Case (ii). A closed lower dock connected to an open upper dock. We interpret the ladder as sitting on the two dock vertices of the lower dock, which are now linked by a 2-edge. This is a loser by Lemma 43.

Case (iii). An open lower dock between two closed upper docks. The two ladders overlap on the lower dock. We shorten one ladder by this edge so that afterwards they only touch on one vertex. Then one ladder contains the additional Maker vertex α while the other does not. Applying the Articulation Lemma to this common point, we see that the whole component must be a loser by Lemma 44.

The remaining cases are covered by case 3, as remarked above. □

Although Observation 45 deals with only two very special first Maker moves, it is the essential step towards the proof of Theorem 38. In the following we check all possible first Maker moves outside of ϕ. The analysis is again split into the old three classes: whether Maker plays between two open docks, between two closed docks, or between an open and a closed dock; the classification above, into components M of $H^{[-\beta]}$, will be used as a tool only.

The general scheme is the same for all cases. Breaker answers Maker's move x by a move in the 2-edge ϕ, at β, say. Then α has become an articulation vertex, so we can write

$$H^{[-\beta]} = M \cup D \quad \text{with} \quad M \cap D = \{\alpha\}$$

and such that M contains Maker's vertex x, which we technically consider as not deleted for a second to get a sound definition of M. The component M is then of one of the three types in Figure 23.

Now comes the decisive trick. We show two things: $M^{[+x]}$ is a loser but $M^{[+x,+\alpha]}$ is a winner. Then by the Articulation Lemma, this implies that the whole hypergraph $H^{[+x,-\beta]}$ is a winner if and only if $D^{[+\alpha]}$ is a winner! But $D^{[+\alpha]}$ is by construction a subhypergraph of $H^{[-\beta,+\alpha]}$. Note that we don't have to put an additional $+x$ in the exponent because the vertex x lies not in D. Now Observation 45 tells us that this hypergraph is lost, so we are done.

What we did in the previous paragraph could be termed less formally in the following way. When we know that $M^{[+x]}$ is a loser but $M^{[+x,+\alpha]}$ is a winner, the Articulation Lemma tells us that α is a reasonable move for Maker. Since he cannot win on $M^{[+x]}$ he makes the best of this part by playing the threat α which turns it into a winner. Now, since we may legitimately assume that Maker will play at α, the problem is reduced to the question whether the rest $D^{[+\alpha]}$ is a winner. Which, as we know, is not.

46. Observation. *If Maker plays his first move between two open docks (including the respective dock vertices) he loses.*

Proof. Breaker answers Maker's move x by playing at β, destroying the upper docks. We write $H^{[-\beta]} = M \cup D$ as described above, where M contains two open docks, so it's type is one of those subtypes of case (iii) in our classification.

Clearly $M^{[+x,+\alpha]}$ is a winner, and since M is a subhypergraph of a type-(iii) component, Lemma 44 tells us that it is a loser. As described above, we conclude that the whole hypergraph $H^{[+x,-\beta]}$ must be a loser. □

47. Observation. *If Maker plays his first move between two closed docks (including the respective dock vertices) he loses.*

Proof. Breaker again takes a vertex from the 2-edge. He has to be a little careful with his choice, however. Have a look at Figure 13 from page 48 again. If Maker's first move x is a vertex of the lower path A then Breaker replies at α, breaking the lower cycle. Likewise, Breaker answers a move in the upper path B at β. In the remaining case $x \in V(F)$ he picks one of α and β arbitrarily. (In the special case when F has length 0 and Maker plays the unique contact vertex in $V(A) \cap V(B)$, we also let Breaker pick one of α and β at will.)

Assume by symmetry that Breaker plays β, i.e., x was played on the upper cycle or the connecting path F. (Have a look again at Figure 23, where the vertex x was already marked.) As the upper cycle has been broken, we can apply Corollaries 41 and 42 to replace the complete upper part $B \cup F$ by a single Maker move at the contact point $p \in V(B)$. Then Lemma 40 tells us that the resulting cycle $A^{[+p]}$ is lost. In terms of our general recipe, we have thus shown that $M^{[+x]}$ is a loser. On the other hand, $M^{[+x,+\alpha]}$ is clearly a winner. Again the general argument described above now settles the issue. □

The remaining closed-open case again bears a difficulty. The general argument we used in the previous cases will only work for the special situation that Maker's move is on the first level of the ladder. (Recall that the core between an open and a closed dock is a ladder.) Plays at higher levels require an inductive argument and are deferred to the moment when we compile all our observations into a proof of Theorem 38.

48. Observation. *If Maker plays his first move on the first level of the ladder between a closed and an open dock, he loses.*

Proof. Assume by symmetry that the lower dock is the open one. Breaker plays at β, destroying all upper docks. Then we know that the resulting component M that contains x is of type (iii) (or a subhypergraph with just one ladder) with Maker's move x on the first level of one ladder. Lemma 44 tells us that $M^{[+x]}$ is a loser and $M^{[+x,+\alpha]}$ is as always trivially a winner. □

The alert reader might have noticed that case (ii) of our classification did not show up in the last three observations. This does not mean that it has been overlooked. It simply was not needed for the proofs. Remember that Observations 46 to 48 are statements about the three connection types from Theorem 38, they only *used* the three M-types from this section as a tool.

Eventually, almost all details of Theorem 38 have been studied. It is time to put our observations together.

Proof of Theorem 38. That the core of a loser can only have the listed connection is obviously true, simply because they are just those types that survived our lengthy discussion from Section 4.

The converse almost follows from Observations 45 to 48. They provide successful Breaker counters against all first Maker moves except for a play on a higher level of a ladder between a closed and an open dock.

This remaining case is the only situation where Breaker must not play in the 2-edge ϕ. Instead, he chops a few steps off the ladder. We prove that Breaker wins if Maker plays on a level $j \geq 2$ of some ladder between a closed and an open dock by induction on the sum S of the heights of all ladders in the core.

At the induction base $S = 0$ there are no ladders, so the statement is trivially true. For the induction step, we let Breaker answer Maker's move x at a_{j-1}, just like in the proof of Lemmas 43 and 44. (See the right-hand side of Figure 22 from page 57 again.) This decomposes the ladder into an upper and a lower part such that the upper is lost by Lemma 44 and the lower remains, after removal of the dangling path F_{j-1}, a ladder of smaller height. Since Maker's move x does not lie in the lower part, we have reduced the original hypergraph to one that still satisfies all requirement of our Theorem but has smaller ladder-height sum S. This finishes the proof. □

The algorithm. It is time to return to our initial complexity question. In the following proof of Theorem 4 we compile the results of the preceding sections into a polynomial-time algorithm for the decision problem of winning and losing. This is a straightforward procedure, simply revisiting all reduction steps and showing that the core types from Theorem 38 are checkable efficiently. We emphasize again that a detailed runtime analysis of the below method is not our goal. Neither do we strive for an actual implementation of the described procedures nor for an improvement of asymptotic runtime bounds. Theorem 38 is a purely qualitative result, identifying the games at hand as a *tractable* subclass of general hypergraph games.

Proof of Theorem 4. Let H be the given almost-disjoint hypergraph of rank-3. By Lemma 17 we can assume that H is connected. If H contains more than one 2-edge, it is a winner by Corollary 11. If it contains no 2-edges, we create all first-move hypergraphs $H^{[+x,-y]}$ with $x, y \in V(H)$ as described in Section 3 in connection with Lemma 24. This produces a quadratic number of hypergraphs, amongst which we have to check those that contain a 2-edge for winning or losing.

All hypergraphs with one 2-edge can be severed at articulation vertices, as described in Lemma 25, until we are left with Maker-2-connected hypergraphs only, each of which contains exactly one 2-edge. (Remember that whenever this process produces two 2-edges, we are done by Corollary 11.)

The core of each of those Maker-2-connected hypergraphs is then decomposed into links between the docks, as we did in Section 4. For each such link we check whether it complies with the specifications of Theorem 38 to see if Maker can win. This is not a difficult task. Each admissible connection type is expressed in terms of paths that are built upon each other. We can use a simple greedy path-finding method to successively reconstruct any required or allowed connection. Whenever we spot a violation of the admissible topology we know that we face a winner. □

6. Almost-Disjointness

We promised some comments on the influence of the almost-disjointness restriction on our games on rank-3 hypergraphs. Have a look at the two overlapping 3-edges in Figure 24, who violate this condition. Assume this configuration occurs within a hypergraph H in such a way that no further edges touch upon the vertices a and b, so that our edge pair is linked to the rest of H only through p and q. We claim that in such a configuration the two 3-edges are of no use for Maker.

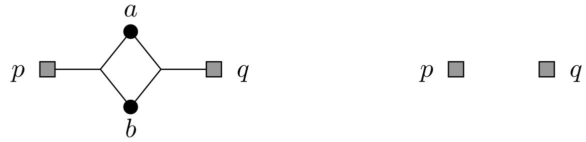


Figure 24. Two worthless 3-edges.

49. Lemma. *A hypergraph H containing the left configuration of Figure 24 with no further edges connected to a and b is a winner iff $H^{[-a,-b]}$, the same hypergraph with this configuration replaced by the one to the right, is a winner.*

PROOF. The hypergraph $H^{[-a,-b]}$ is a subhypergraph of H, so if Maker wins on the former he clearly also wins on the latter. We show that a making strategy σ for H on the left yields also a Maker win on the reduced hypergraph on the right. Therefore we follow this strategy on both hypergraphs, copying our Maker moves given by σ from the left to the right and the Breaker answers, which are played on the right, back to the original hypergraph H. This works fine as long as our strategy σ does not prompt us to play at a or b. In that case, if we must play a, say, we actually do so on the left and then—this is the trick—answer it immediately by a fake Breaker move at b. In the reduced hypergraph on the right side, these two half-moves are simply left out. After a and b are taken on the left, we can continue with σ until the whole board is full.

Who has won? Since we followed the winning strategy σ on the left, we must have won there, i.e., some edge $e \in V(H)$ is completely ours. But since we have given Breaker a vertex in each of the two 3-edges on a and b, this winning edge is neither of them. Consequently, we have also occupied all vertices of e on the right. □

A similar situation—or rather the opposite—is shown in Figure 25. Again the two edges are part of some bigger hypergraph H in such a way that no further edge contains a or b and everything else is linked through p and q, who now are the inner vertices of this little cycle.

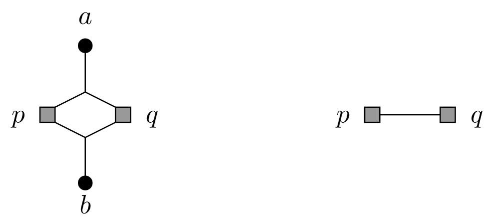


FIGURE 25. Two 3-edges behave like a single 2-edge.

50. LEMMA. *A hypergraph H containing the left configuration of Figure 25 with no further edges connected to a and b is a winner iff $H^{[+a,+b]}$, the same hypergraph with this configuration replaced by the one to the right, is a winner.*

PROOF. Assume a making strategy σ for the left hypergraph H. As above we follow σ on the right until a move in $\{a, b\}$ is required. In this case, play this vertex, a, say, and as above, reply by a fake Breaker move at b. This deletes one of the two 3-edges and turns the other one into a 2-edge on p and q. From then on we just pursue σ again on both sides to the end of the game. As in the proof of Lemma 49, we conclude from the fact that σ has lead to a win on the left that we have also won on the right because all edges on the left are also present on the right. The newly created 2-edge is just the one that was present on the right in the first place.

The other implication works very similar, with exchanged sides. Assume we have a making σ on the right. Against Breaker on the left we also follow σ—until Breaker takes one of the vertices a and b. (We won't play there first because our strategy does not know those vertices.) In this case, we take the other vertex and then resume our strategy σ again. Just as above the two hypergraphs are now completely identical, so we win on the left because we are sure to win on the right. □

Let us call a pair of two 3-edges that overlap on two vertices a *diamond*. The previous discussions have shown again that the inner vertices are, as so often, the valuable ones, while the outer vertices are of minor interest.

Assume we try to find out whether some given rank-3 hypergraph that is not almost-disjoint is a winner. If we find a configuration like the one on the left of Figure 26, Maker can win if the path P connecting the two diamonds is almost-disjoint because the terminal diamonds behave like 2-edges. If P is not so nicely behaved and there sits a diamond somewhere on P, as shown on the right-hand side of the figure, we may assume that this diamond has some further edge f attached to one of its inner vertices because otherwise, we could just remove that diamond without changing the value of the game. From where f is connected, the new diamond looks like a 2-edge again; so if we trace a path from f back to one of the two terminal diamonds (using Maker-2-connectivity) we win as soon as we meet another diamond at an inner vertex.

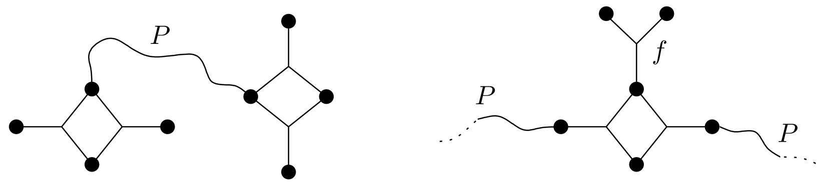


FIGURE 26. Two diamonds connected at their "good" vertices.

Though we have only just started the discussed of a simple example, it appears as if the presence of only two or three diamonds in a Maker-2-connected rank-3 hypergraph create an influence of "pseudo 2-edges" that should, in general, lead to a win like in the left of Figure 26. What this "general case" should precisely be, is of course unclear and a proper analysis appears to bring a lot of case distinctions about. Yet, this brief discussion might indicate that the problem might be solvable in a way that rids a given hypergraph from its diamonds so that we may afterwards apply Theorem 4 directly, as a black box, without unrolling the tedious proof of Theorem 38 again.

7. Comparing Games

We close this chapter by introducing a new view on positional hypergraph games that incorporates several concepts we have met so far.

Let us have a closer look at our favorite tool, the Articulation Lemma. Intuitively, it tells us that the two halves of a hypergraph that are only connected through a single articulation vertex, can interact in only three different ways. So in a sense, seen through an articulation, there exist only three different types of hypergraphs: those halves A that win on their own, those that do not help the B on the other side at all, and those "semi-winners" who are not winners themselves but for which $A^{[+p]}$ is a winner. Cutting such a hypergraph in two at the articulation, we get an isolated "half" with a marked contact point.

51. Definition. A *pointed hypergraph* is a pair (H, p) of a hypergraph $H = (V, E)$ and a *point* $p \in V$. The *one-point union* $(A, p) \sqcup (B, q)$ of two pointed hypergraphs (A, p) and (B, q) is the pointed hypergraph

$$\big((A \mathbin{\dot\cup} B)/\{p = q\}, \{p, q\}\big),$$

meaning that we take the disjoint union of A and B and then identify the two points p and q, choosing this merged vertex as the point of the union.

The term "one-point union" is borrowed from topology, confer [**10**, Chp. 1, Sec. 13]. Sometimes, when the precise choice of the point is not relevant, we shall treat a pointed hypergraph just as a hypergraph, simply ignoring the point, speaking of winners and losers, for example.

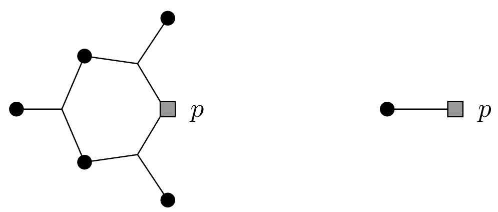


Figure 27. Two equivalent pointed hypergraphs.

Of course, we want to *play* on such one-point unions. Compare the two pointed hypergraphs in Figure 27. We claim that with respect to composition at the point p, these pointed hypergraphs have the same value in any game. Whatever partner (X, q) you plug in at p from the right, either you win on both one-point unions or on neither of them. We defer the proof of this statement until we have prepared suitable notions, which shall allow for a much more general result.

The partial order $\mathcal{H}_1$. Define a partial quasi-order on the class of all pointed hypergraphs by letting

$$A \leq B$$

for two pointed hypergraphs A, B iff

$$A \sqcup X \text{ is a winner} \Rightarrow B \sqcup X \text{ is a winner} \tag{13}$$

for all pointed hypergraphs X.

This relation is obviously reflexive and transitive but clearly not anti-symmetric. Call A and B equivalent if $A \leq B$ and $B \leq A$, denoted by $A \equiv B$. We define $\mathcal{H}_1$ to be the partially ordered set that results from identifying equivalent pointed hypergraphs.

This notion of equivalence captures all information about a pointed hypergraph with respect to its impact on winning and losing when plugged into some other pointed hypergraph. In the union $A \sqcup X$ we may replace A by any $B \equiv A$ without changing Makers prospects of winning—independent of the partner X. Note that by the very definition of $\leq$, two pointed hypergraphs A and B are not equivalent iff there exists some "separating" pointed hypergraph Z such that $A \sqcup Z$ is a winner but $B \sqcup Z$ is a loser or vice versa. So with respect to this Z, the pointed hypergraphs A and B show a different behavior.

What can we say about $\mathcal{H}_1$? First note that it contains a maximal and a minimal element. Any winner A with any vertex $p \in V(A)$ as its point, is greater or equal than any other pointed hypergraph. Hence, there is a maximal element 1 in $\mathcal{H}_1$ that contains all pointed winners. To see that it contains only winners, consider some winner A together with an arbitrary loser B and let U be a pointed hypergraph without any edges. Then $A \sqcup U$ is a winner while $B \sqcup U$ is still a loser. Hence, $A \not\leq B$. This means that no loser lies above any winner and consequently the class 1 contains only winners (each with an arbitrary vertex as point). This observation allows us to abbreviate the expression "A is a winner" as $A \in 1$.

A similar argument shows that $\mathcal{H}_1$ has a minimal element, 0, which contains all *absolute losers*—pointed hypergraphs that do not contribute anything. All empty graphs, like U from above, fall into this class. Trivially, because whenever $U \sqcup X$ becomes a winner for such a U and some X then X alone must already be a winner. Hence, for any pointed C the one-point union $C \sqcup X$ is also a winner and thus $U \leq C$. Note that unlike the case of the maximal element, 0 is *not* the class of all losers but much smaller. So $U \in 0$ is really a stronger statement than saying that U is a loser!

What lies between 0 and 1 in $\mathcal{H}_1$? The answer is simple, we already know. The following theorem is the Articulation Lemma in disguise.

52. Theorem. *The poset $\mathcal{H}_1$ is a linear order of exactly three elements.*

Proof. We show that $A \equiv B$ for any two arbitrary pointed hypergraphs, neither of which is a winner nor an absolute loser, i.e., $A, B \notin \{0, 1\}$. Then we know that there can be at most one further class besides 0 and 1.

Since $B \notin 0$, there exists a $Y \notin 1$ with $B \sqcup Y \in 1$. Then the Articulation Lemma tells us that $B^{[+q]}$ must be a winner, where q be the point of B. On the other hand, we know that for any X with $A \sqcup X \in 1$ the reduction $X^{[+p]}$ must be a winner (p being the point of X), also by the Articulation Lemma, because $A \notin 1$. Together this means that for any such X the composition $B \sqcup X$ is also a winner. Hence, $A \leq B$. Exchanging the roles of A and B we also obtain the converse relation and therefore, $A \equiv B$.

To see that a third class in $\mathcal{H}_1$ exists at all, simply note that the 2-edge in Figure 27 is neither a winner nor an absolute loser. □

Our original claim about the two pointed hypergraphs from Figure 27 is now almost proven. We just argued that the single edge lies in the unique intermediate class of $\mathcal{H}_1$. By Lemma 40 the cycle on the left is no winner either and it also no absolute loser because it gives a win if composed with itself. Hence, by Theorem 52 the two pointed hypergraphs must lie in the same equivalence class. The whole order $\mathcal{H}_1$ is shown in Figure 28, with a typical representative for each class.

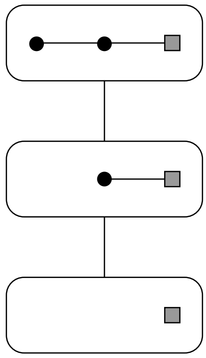


Figure 28. The poset $\mathcal{H}_1$.

Merging along many points. One can generalize the union at just one point to amalgamations along larger sets. Actually, the index of $\mathcal{H}_1$ already calls for the following definitions.

53. Definition. A *k-pointed hypergraph* is a tuple $(H, p_1, \ldots, p_k)$ consisting of a hypergraph $H = (V, E)$ and a list of distinct vertices $p_1, \ldots, p_k \in V$ called *points*. The *k-point union* $(A, p_1, \ldots, p_k) \sqcup (B, q_1, \ldots, q_k)$ of two k-pointed hypergraphs is the k-pointed hypergraph

$$\big((A \dot\cup B)/\{p_i = q_i : 1 \le i \le k\}, \{p_1, q_1\}, \ldots, \{p_k, q_k\}\big),$$

meaning that we take the disjoint union of A and B and then merge each individual point pair $\{p_1, q_1\}$ through $\{p_k, q_k\}$ into a single new point.

Our partial quasi-order generalizes naturally by letting $A \le B$ for two k-pointed hypergraphs iff (13) holds for all k-pointed hypergraphs X. Then $\mathcal{H}_k$ is defined as the partially ordered set of equivalence classes of k-pointed hypergraphs with the order induced by $\le$.

As an example for 2-pointed hypergraphs we remark that we have already worked with the partial order $\mathcal{H}_2$: in the previous section on almost-disjointness. The reader will have already noticed the similarity of Figure 27 with Figures 24 and 25 from pages 62 and 63. This is, of course, no coincidence. Phrased in our new terminology, the respective Lemmas 49 and 50 are actually equivalence proofs for 2-pointed hypergraphs.

As with $\mathcal{H}_1$ we observe that each $\mathcal{H}_k$ has a maximal element 1, which contains exactly all winners, and a minimal element 0, the class of absolute losers. The respective arguments are exactly the same as for the case $k = 1$ above. We note that the degenerate case $k = 0$ has already appeared, in form of Lemma 17. With no points, $A \sqcup B$ is just $A \dot\cup B$ and therefore the dichotomy of Lemma 17 applies: $\mathcal{H}_0$ consists of only two classes, 0 and 1. (Here losers are always absolute losers.)

Can we say anything more about $\mathcal{H}_k$ for $k \ge 2$? Unfortunately, our knowledge amounts to pretty little. We have the following basic lower bounds.

54. Proposition. *For each $k \ge 0$, the partial order $\mathcal{H}_k$ contains a chain of length $k + 2$.*

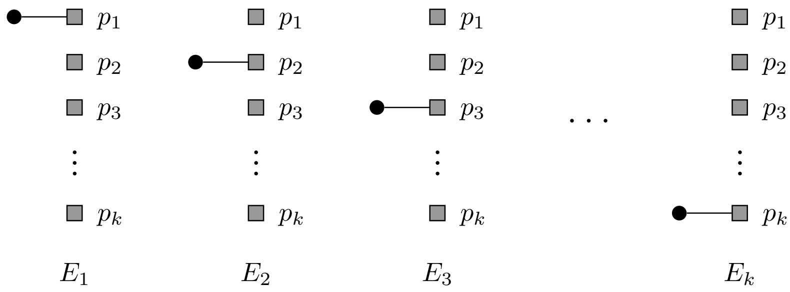


FIGURE 29. Some basic k-pointed hypergraphs.

PROOF. From the basic k-pointed hypergraphs E_i in Figure 29 we construct a chain of length $k+2$ in $\mathcal{H}_k$ as follows. Let U_r denote the k-point union $E_1 \sqcup \cdots \sqcup E_r$ of the first r such hypergraphs, $0 \leq r \leq k$. So the k-pointed hypergraph U_r contains exactly r independent 2-edges on the points p_1 through p_r, and $k-r$ isolated points. Further let U_{k+1} be an arbitrary k-pointed winner. We have

$$U_0 < U_1 < \cdots < U_k < U_{k+1} \quad \text{in } \mathcal{H}_k$$

because for each $r \leq k$ the k-point union $U_r \sqcup E_r$ is a winner while $U_{r-1} \sqcup E_r$ is obviously lost; and U_{k+1} is larger than all the other U_r. □

55. PROPOSITION. *For each $k \geq 1$, the partial order on $\mathcal{H}_k$ contains an antichain of length $\binom{k}{\lfloor k/2 \rfloor}$.*

PROOF. For each index set $I \subseteq \{1, \ldots, k\}$ of cardinality $\lfloor k/2 \rfloor$ we let U_I denote the composition of all E_i with $i \in I$. For any pair $J \neq J'$, the k-pointed hypergraphs U_J and $U_{J'}$ are incomparable because for $r \in J \setminus J'$ the composition $U_J \sqcup E_r$ is a winner but $U_{J'} \sqcup E_r$ is not, i.e., $U_J \not\leq U_{J'}$; and likewise, any $E_{r'}$ with $r' \in J' \setminus J$ shows that $U_J \not\geq U_{J'}$. □

These basic calculations might give us some first feeling for the complexity of the $\mathcal{H}_k$. However, they do not address the important point. The crucial question is:

Are all $\mathcal{H}_k$ finite?

If some $\mathcal{H}_k$ is finite then so are all $\mathcal{H}_j$ with $j \leq k$, obviously, because any $\mathcal{H}_j$ is embeddable in $\mathcal{H}_k$ by adding $k-j$ isolated dummy points to any j-pointed hypergraph. We know that $\mathcal{H}_1$ is finite. Is there a level in the hierarchy $(\mathcal{H}_k)$ where the complexity explodes from finite to infinite? If so, this should probably happen quite early, maybe on level two or three. However, any such statement appears to be difficult to prove.

The finiteness of $\mathcal{H}_k$ would have strong implications on the complexity of weak positional games on hypergraphs that are only Maker-k-connected. Such a hypergraph H can be written as a nontrivial union of two subhypergraphs A and B who overlap on no more than k vertices. If we interpret A and B as k-pointed hypergraphs with these vertices as points, we can write $H = A \sqcup B$. If $\mathcal{H}_k$ should be finite we could, in principle, identify the equivalence classes of A and B *independently*—by solving the k-point unions $A \sqcup X$ and $B \sqcup X$ for a complete set of representatives X of $\mathcal{H}_k$. The outcomes of those subproblems would then tell us the value of H. This way we decompose the big problem whether H is a winner into a constant number of such questions for smaller hypergraphs. (Note that the size of the representatives is bounded.) For a decision problem that is PSPACE-complete in general, this would be quite a remarkable result: we could divide and conquer with very little overhead.

Actually, we have used this principle already extensively throughout this chapter—for the case $k = 1$. In Section 3 we repeatedly cut at articulations until we obtained Maker-2-connectivity. Each decomposition step used implicitly, through the Articulation Lemma, the fact that $\mathcal{H}_1$ contains only three classes, one of which could always be excluded because of the existence of a 2-edge in one half.

I have constructed an approximation of $\mathcal{H}_2$ that carries a lot of symmetries and which might already be the complete picture but I see by now no way of proving such a statement. Intuitively, finiteness of $\mathcal{H}_k$ means that through only k points, the two halves cannot exchange an arbitrary amount of information. It should be that during a play across a small interface, the points soon get congested—until the graph eventually decomposes into completely disjoint parts. I am strongly convinced of the following.

56. CONJECTURE. *The poset $\mathcal{H}_k$ is finite for every $k \geq 0$.*

CHAPTER 3

Digraph Roots

1. Matrices and Digraphs, Powers and Roots

Consider a directed graph D (or digraph, for short) on n vertices together with its adjacency matrix, i.e., the $n \times n$ matrix $A = (a_{ij})$ with $a_{ij} = 1$ iff there is an arc $j \to i$ in D and $a_{ij} = 0$ otherwise. We can use iterated multiplications of the adjacency matrix with itself to find paths in the digraph. Precisely, the (i, j)-entry of the kth power A^k of A is positive iff there is a walk of length *exactly* k from vertex i to vertex j in D. By *walk* of *length* k we mean a sequence $(v_0, v_1, \ldots, v_k)$ of $k + 1$ vertices with an arc from v_{i-1} to v_i for $1 \leq i \leq k$, where vertices may appear several times; in contrast to a *path*, which is a walk with all vertices distinct. We are only interested in the existence of such walks, not their number—which is counted by the respective entry of A^k—so we interpret A as a Boolean 0/1-matrix with the product $C = A \cdot B$ defined in the usual way:

$$c_{ij} = \bigvee\nolimits_{h=1}^{n} a_{ih} \wedge b_{hj}.$$

Identifying a digraph with its adjacency matrix, we define the kth power, $k \in \mathbb{N}$, of a digraph D to be the digraph D^k on the same vertex set and with an arc from a to b if and only if there is a directed walk of length *exactly* k from a to b in D (possibly visiting some vertices several times). Figure 1 shows an example.

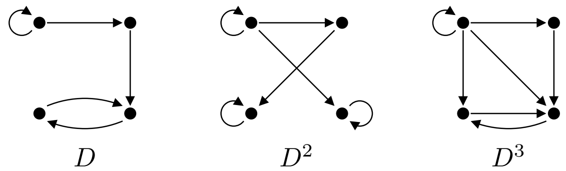


FIGURE 1. Powers of a digraph.

Note that the interpretation of digraphs as Boolean matrices implies that our digraphs may have loops but no multiple arcs. It is easy to see that the adjacency matrix of D^k is in fact the kth Boolean power of the adjacency matrix of D (see, for example, [**38**]).

Boolean matrix algebra serves as a fundamental tool in algorithmic graph theory. The correspondence between graphs and matrices lies at the heart of many fundamental algorithms for transitive-closure or shortest-path computations [**35, 1, 12**] (where usually powers of the matrix $A + I$, with I the identity matrix, are considered to account for all paths *up to* a certain length k).

We are interested in the inverse operation to exponentiation: root finding. The complexity of the following problem was open until now.

THE BOOLEAN-MATRIX-ROOT PROBLEM. Given a Boolean $n \times n$ matrix A and an integer $k \geq 2$, does there exist a kth *root* B of A, that is, an $n \times n$ matrix B with $B^k = A$.

Or equivalenty, stated in terms of digraphs:

THE DIGRAPH-ROOT PROBLEM. Given a digraph D and an integer $k \geq 2$, does there exist a kth *root* R of D, that is, a digraph R on the same vertex set, with $R^k = D$.

Twenty years ago, in the open-problems section of his book [**26**], Kim inquired for the special case $k = 2$, whether the Boolean-matrix-root problem might perhaps be NP-complete. We answer this question in the affirmative.

1. THEOREM. *Deciding whether a square Boolean matrix or, equivalently, a digraph has a* kth *root is NP-complete for each single parameter* $k \geq 2$.

With the right computational problem for the reduction, the proof of this result turns out surprisingly simple. This is quite remarkable since it thus relates digraph roots very closely to a well-known NP-complete problem. It allows to identify quite accurately "the reason" for the hardness of the problem. In an attempt to isolate and inhibit these computationally difficult aspects, we shall discover a close connection between digraph roots and graph isomorphism, which eventually leads to a further complexity result (Theorem 3). But let us postpone these issues till after the proof and discussion of Theorem 1.

Related work—related questions. Over the field of complex numbers or the reals, matrix roots are a well-studied and still up-to-date topic of linear algebra [**29, 24, 36**]. But results from that field of research do generally not apply to Boolean matrices. While it is known, for example, that every regular matrix over the complex numbers has a kth root for any $k \geq 2$ [**36**], this is not true for Boolean matrices, as the invertible matrix $\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ shows. Further, complex or real matrices are amenable to numerical methods like Newton iteration [**22**], whereas such techniques clearly do not apply to Boolean matrices. When it comes to roots, Boolean matrices don't seem to have much in common with matrices over $\mathbb{C}$ since the former behave much more rigidly than the latter.

The situation is, however, different if we ask for powers of a matrix instead of roots. There are theoretical results on Boolean matrix powers [**13**] and in practice we can, of course, compute the kth power of a Boolean matrix A by treating it as a matrix over the reals. We calculate A^k over $\mathbb{R}$ and afterwards replace each positive entry with 1. This simple reformulation allows us, for example, to apply fast matrix multiplication methods such as Strassen's to path problems in graphs [**35, 1**]. But this simulation through matrices over the reals clearly only works because there cannot happen cancellation between positive and negative entries. For root finding, such simulation over $\mathbb{R}$ or $\mathbb{C}$ would lead into major problems.

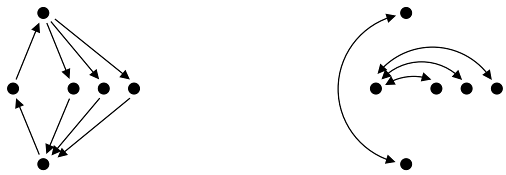


FIGURE 2. A directed square root (left) of a symmetric digraph (right) which does not have a symmetric square root.

Alternative notions of graph powers. A problem similar to the one at hand has been discussed by Motwani and Sudan. In [**34**] they showed that computing square roots of undirected graphs is NP-hard. But their notion of graph powers differs from ours in two important points.

They consider undirected graphs only, which in our setting would correspond to symmetric digraphs, i.e., all edges are bidirectional. This not only restricts the set of possible inputs but also—and this is the decisive difference—the solutions. For example, the symmetric digraph on the right of Figure 2 has the digraph to its left as a square root, but it is not the square of any symmetric digraph. To see this, observe that any square root of an undirected graph with maximum degree strictly greater than 2 must also have a vertex of degree at least 3. Such a vertex would in turn induce a triangle in the square. The digraph in the figure has maximum degree 3 but it does not contain a triangle.

Further, Motwani and Sudan define squaring to maintain existing edges, which in our setting would correspond to attaching loops to all vertices. This monotonicity ensures that much information of the underlying graph can be read off from its square and the hardness proof of [**34**] makes essential use of this property. In contrast to this, squaring a digraph under the rules derived from Boolean matrix multiplication can almost completely destroy the neighborhood information and may even decompose the digraph. Actually, most of our arguments depend crucially on such vanishing edges. So apparently, the squares in [**34**] and our notion of powers are fundamentally different concepts.

Nomenclature. In the light of the preceding discussion, Boolean matrices form the right framework to ask questions about roots in the sense we defined them. They do not leave the ambiguities that the expression "graph root" obviously has and locate the problem correctly in the context of semigroups. However, for the actual work, the proof of Theorem 1, we will resort to the language of graph theory since our arguments will extensively use respective notions like paths, cycles, and vertex neighborhoods. Moreover, after the NP-completeness proof we shall emphasize the link to graph theory even more by relating our roots to graph-isomorphism.

So let us agree on the precise meanings of some common graph theoretic notions whose exact distinction will be crucial in certain situations. A *walk* is simply a sequence $(a_0, a_1, \ldots, a_r)$ of vertices with an arc $a_i \to a_{i+1}$ for $0 \leq i < r$, whereas a *path* is a walk of pairwise distinct vertices. The parameter r is the length of the walk respectively path. A *cycle* is a closed walk, that means, $a_0 = a_r$ and vertices may be traversed several times. By *isolated cycle* we mean a strongly connected component of a digraph where each vertex has indegree and outdegree 1, i.e., a single non-self-touching cycle without further arcs.

For a digraph R on vertex set V we let

$$R(v) := \{w \in V \mid v \xrightarrow{R} w\}$$

denote the set of *outneighbors* of v in R. Defining $\bar{R}$ to be the digraph obtained from R by inverting all arcs, we write $\bar{R}(v)$ for the *inneighbors* of v. Note that our generalization

$$R(U) := \bigcup_{u \in U} R(u)$$

to subsets $U \subseteq V$ diverts from standard notation (as for example in [**3**, sec. 1.2]) as $R(U)$ need not be disjoint from U. In other words, we let a digraph act on vertex sets just like its adjacency matrix acts on the characteristic vectors of such sets.

These definitions help simplify our notation. For example, we write $x \in \bar{R}^j(Y)$ to state that there is a walk of length j from x to some vertex in $Y \subseteq V$ and expressions like $R^3\bar{R}^8R$ make perfect sense, encoding some kind of zig-zag walk through the digraph R.

2. NP-Completeness

This section comprises the proof of Theorem 1; but before turning to the details, presenting a suitable NP-complete problem which we can reduce to digraph roots, let us collect some motivating observations about digraph square roots.

Consider some set X of vertices of a digraph D and let Z denote all outneighbors of vertices in X. Assume for simplicity that X and Z are disjoint, so in particular, there are no loops or cycles on these vertices. In a square root of the digraph D, any of the arcs from X to Z must be realized as paths of length two. Hence, the root must provide a set Y of intermediate vertices through which all these paths can pass. If now—for whatever reason—there is only a small number of such intermediate vertices available, $|Y| \leq r$, say, with r a little smaller than $|X|$ and $|Z|$, these paths must intersect in order to ship all their information from X to Z. This situation is almost exactly captured by the following decision problem, which is already listed in Garey and Johnson's classic [**18**] (p. 222).

The Set-Basis Problem. Let $\mathfrak{C}$ be a collection of subsets of some finite set S. A *set basis* for $\mathfrak{C}$ is another collection $\mathfrak{B}$ of subsets of S such that each $C \in \mathfrak{C}$ can be written as a union of sets from $\mathfrak{B}$. Given a finite set S, a collection $\mathfrak{C}$ of subsets of S, and an integer $r \leq |S|$, the *set-basis problem* asks whether there exists a set basis $\mathfrak{B}$ for $\mathfrak{C}$ consisting of at most r sets. This problem is known to be NP-complete [**40**].

We claim that the local configuration of the above square-root problem is nothing but a set-basis instance. The sets X and Z correspond to the given collection $\mathfrak{C}$ and the ground set S, respectively, while the intermediate vertex set Y takes the place of the sought-after collection $\mathfrak{B}$.

(a)

(b)

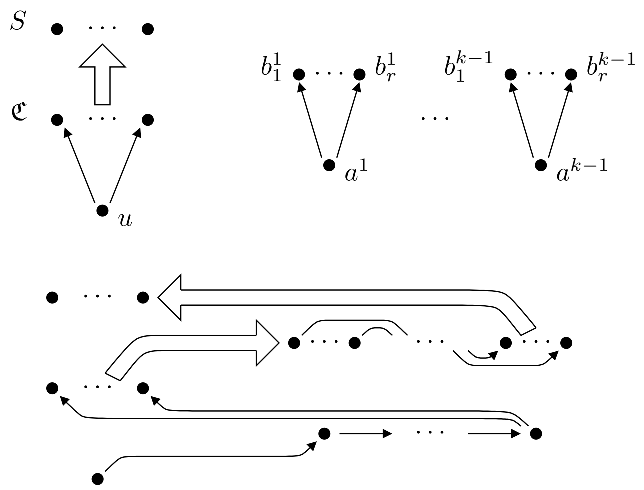


FIGURE 3. Reducing set basis to kth root (a) and encoding a set basis as a root (b). (Wide arrows represent collections of arcs that depend on the actual instance.)

Our precise proof of this claim, which also treats the general case of arbitrary kth roots, comes in the three customary parts: a reduction from a set-basis instance to a kth-digraph-root instance and the two complementary transformations between valid solutions.

The reduction. From a set-basis-problem instance $(\mathfrak{C}, S, r)$ we construct a digraph D such that D has a kth root iff $\mathfrak{C}$ has a set basis $\mathfrak{B}$ of size at most r. We may assume w.l.o.g. that neither the collection $\mathfrak{C}$ nor any $C \in \mathfrak{C}$ be empty, that all $C \in \mathfrak{C}$ be pairwise distinct, and further that $\bigcup \mathfrak{C} = S$, i.e., each $s \in S$ lie in some set $C \in \mathfrak{C}$.

As suggested by the above discussion, our construction essentially draws the containment graph of the set system $\mathfrak{C}$ on S and provides the right number of intermediate vertices. Surprisingly few framework arcs will have to be added in order to ensure that any root uses them as intended.

We start with the containment relations. The digraph D possesses the sets $C \in \mathfrak{C}$ and the elements $s \in S$ as vertices and additionally an "anchor vertex" u. Define the containment arcs

$$C \xrightarrow{D} s \qquad \text{for all pairs } (C, s) \in \mathfrak{C} \times S \text{ with } s \in C \tag{14}$$

and additionally the grounding arcs

$$u \xrightarrow{D} C \qquad \text{for each } C \in \mathfrak{C}.$$

Compare the left component of Figure 3(a).

The intermediate vertices come in $k-1$ isomorphic components which are simply stars. The μth component consists of the $r+1$ vertices $a^\mu, b_1^\mu, b_2^\mu, \ldots, b_r^\mu$ connected via

$$a^\mu \xrightarrow{D} b_i^\mu \qquad \text{for } i \in \{1, \ldots, r\},$$

as shown in the right half of Figure 3(a).

Constructing a root from a set basis. To show that our construction works, we describe how to obtain a kth root R of the digraph D from a set basis of size r for $\mathfrak{C}$. Therefore we first need a lot of framework arcs that are independent of the actual basis $\mathfrak{B}$: the horizontal paths

$$u \xrightarrow{R} a^1 \xrightarrow{R} a^2 \xrightarrow{R} \cdots \xrightarrow{R} a^{k-1}$$

and

$$b_i^1 \xrightarrow{R} b_i^2 \xrightarrow{R} \cdots \xrightarrow{R} b_i^{k-1} \qquad \text{for each } i \in \{1, \dots, r\},$$

and also the back connections

$$a^{k-1} \xrightarrow{R} C \qquad \text{for each } C \in \mathfrak{C};$$

drawn as thin arcs in Figure 3(b).

The remaining arcs depend on the given set basis $\mathfrak{B} = \{B_1, \dots, B_r\}$, which comes with a representation

$$C = \bigcup_{i \in I_C} B_i, \qquad I_C \subseteq \{1, \dots, r\} \tag{15}$$

of each set $C \in \mathfrak{C}$.

Note that a basis with less than r sets can be extended to one of size r by adding singleton sets $\{s\} \subseteq S$ and it is also clear that we can pick the collection $\mathfrak{B}$ and the index sets I_C in such a way that each index $i \in \{1, \dots, r\}$ appears in at least one I_C.

The set basis $\mathfrak{B}$ is now wired via

$$b_i^{k-1} \xrightarrow{R} s \qquad \text{for each pair } (i, s) \text{ with } s \in B_i,$$

while the corresponding representations are realized as

$$C \xrightarrow{R} b_i^1 \qquad \text{for each index } i \in I_C.$$

These connections appear bundled as wide arrows in Figure 3(b).

These definitions guarantee that there exists an R-walk of length k from a certain C to some $s \in S$ iff there exists any basis set B_i with $s \in B_i$ and $i \in I_C$. By the definition of a set basis, the latter condition is equivalent to $s \in C$, which, by construction of the digraph D, means just that there is a D-arc from C to s. Thus we have shown that R^k equals D on $\mathfrak{C} \times S$. The identity of these two digraphs on the remaining vertices is immediate.

Getting a set basis from a root. We turn to the other, slightly more intricate implication. Let D be the digraph constructed from a given set-basis instance $(\mathfrak{C}, S, r)$ and let R be any kth root of D. From this root we must obtain a set basis $\mathfrak{B}$ for $\mathfrak{C}$ with at most r sets. The basic idea is, of course, to show that the root R must look essentially as the one we constructed in the preceding paragraph.

First of all, observe that cycles in R would induce cycles in any positive power of R. Thus, R contains no cycles. Now consider an arbitrary vertex $C \in \mathfrak{C}$. Since $u \to C$ in D, there must be an R-walk of length k from u to C. We claim that all interior vertices of any such walk P are from the set $\{a^1, \dots, a^{k-1}\}$. To see this, pick any interior vertex x on P. Clearly x must have positive outdegree in D because C has. So x can only be some a^μ or from the set $\mathfrak{C}$; the remaining alternative $x = u$ would yield a cycle. Assume for contradiction that $x \in \mathfrak{C}$. Then there is a path Q of length k in

R from u to x. Because x was assumed to be an inner vertex on the path P, a certain inner vertex y on Q is at distance $-k$ from C. This means $y = u$, which implies that the vertex u lies on an R-cycle—a contradiction.

So all interior vertices of R-walks from u to some $C \in \mathfrak{C}$ are from the set $\{a^1, \ldots, a^{k-1}\}$. Obviously, any such path must use each of these a^μ exactly once since otherwise there would be cycles. Furthermore, all such paths pass the a^μ in the same order, again because two different orders would yield cycles. We may assume by symmetry that the a^μ are traversed from a^1 through a^{k-1}. Thus we see that $R(a^{k-1}) = \mathfrak{C}$ and conclude

$$R^{k-1}(\mathfrak{C}) = R^{k-1}R(a^{k-1}) = D(a^{k-1}) = \{b_1^{k-1}, \ldots, b_r^{k-1}\}.$$

So all R-walks from $\mathfrak{C}$ to S pass through these b_i^{k-1}. We focus on the ultimate edges on any such walk and define

$$B_i := R\big(b_i^{k-1}\big) \qquad \text{for } 1 \le i \le r.$$

We claim that $\mathfrak{B} := \{B_1, \ldots, B_r\}$ is a set basis for $\mathfrak{C}$. This is easily verified. Reading the defining relation (14) as

$$C \xrightarrow{R^k} s \qquad \Longleftrightarrow \qquad s \in C,$$

one sees that the index sets

$$I_C := \big\{ i \mid b_i^{k-1} \in R^{k-1}(C) \big\}$$

yield basis representations of the sets $C \in \mathfrak{C}$ as in Equation (15).

This concludes the proof of Theorem 1.

We emphasize that the given set-basis instance is completely maintained by our reduction. Its containment relations are encoded one-to-one by arcs of the digraph. Thus, on the large scale, an instance of the digraph-root problem can be seen as a collection of many interacting set-basis problems. One might well argue that finding digraph roots is actually a generalized set-basis problem.

As a corroboration for this point of view we mention that the set-basis problem already appeared before in connection with Boolean matrix algebra. Markowsky [**31**] used it in a very economic proof for the NP-completeness of Schein-rank computation.[1]

3. Roots and Isomorphism

Let us carry the concluding remarks of the preceding section a little further and have a closer look at Figure 3 from page 75 again. The construction there required only paths of length 2, which then induced a few long paths in the root. One could say that the computational complexity of root finding results from the described interaction of many very short paths. In some sense, our proof has exploited a *local* phenomenon. If we suppress the local interaction by some restriction on the digraph, maybe we can find some further properties of digraph roots that live on a *global* scale. Here is our approach.

[1]Analogous to the matrix rank over fields, the *Schein rank* of a Boolean matrix A is the minimal integer ρ such that A can be represented as a Boolean sum $A = \bigvee_{i=1}^{\rho} c_i r_i$, where the c_i are column and the r_i row vectors with zero-one entries [**26**, Sec. 1.4].

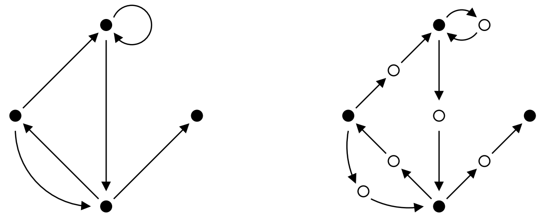


Figure 4. The complete subdivision (right) of a digraph (left).

2. Definition. The *complete subdivision* of a digraph D is the digraph obtained from D by replacing each arc $a \to b$ of D by a new vertex x_{ab} and the two arcs $a \to x_{ab} \to b$. (See Figure 4.) We call a digraph a *subdivision digraph* if it is (isomorphic to) the complete subdivision of some digraph.

Subdivisions are a fundamental notion in graph theory but opposed to their common usage in relation with topological minors, we employ them here to equip our digraphs with a certain stiffness. The effect is the desired inhibition of the local interaction we exploited in the NP-completeness proof. However, the problem of root finding for such subdivision digraphs does not become trivial. Instead, the following surprising relation to graph isomorphism shows up.

3. Theorem. *Deciding whether a subdivision digraph with positive minimal indegree and outdegree has a* kth *root, is graph-isomorphism complete for each single parameter* $k \geq 2$.

The *graph-isomorphism problem* asks whether two given (di)graphs are isomorphic or not, i.e., whether there exists an arc-preserving bijection between their vertex sets.[2] No polynomial-time algorithm for this problem is known, neither is it known to be NP-complete. On the contrary, it is a prime candidate for a problem strictly between P and NP-completeness (cf. [**27**] and [**30**]). Computational problems of the same complexity as the graph-isomorphism problem are called *graph-isomorphism complete*, or simply *isomorphism complete* because isomorphism problems for several algebraic or combinatorial structures fall into this class. For example, isomorphism of semigroups and finite automata [**9**], finitely represented algebras, or convex polytopes [**25**]. Other problems ask for properties of the automorphism group of a graph, for example, computing the order of this group or its orbits [**33**].[3] Finally, several restrictions of the graph-isomorphism problem are known to remain isomorphism complete, as for example isomorphism of regular graphs [**9**].

As the above list indicates, actually all problems known to be isomorphism complete are more or less obviously isomorphism problems of various combinatorial structures. Hence, the relation between digraph roots and

[2] One usually considers undirected graphs but it is well-known and easily seen that with respect to their computational complexity the undirected and directed version of the problem are equivalent.

[3] The latter two problems are known to be isomorphism complete only in the weaker sense of Turing reduction, as opposed to the concept of many-one reduction.

graph isomorphism we are going to establish in our proof of Theorem 3 may come quite as surprise.

From isomorphisms to roots. Theorem 3 rests on a structural result (Theorem 6) which states that any kth root of a subdivision digraph D establishes isomorphisms between the components of D. This is just the kind of global structure we wanted to find when we defined subdivision digraphs.

The starting point is the following connection between digraph roots and digraph isomorphism, which holds for arbitrary digraphs. Subdivisions will then be needed to obtain a converse of this result.

4. PROPOSITION. *Let $D = D_1 \,\dot\cup\, D_2 \,\dot\cup \cdots \dot\cup\, D_k$ be the disjoint union of k isomorphic digraphs $D_1, \dots, D_k$. Then D has a* kth *root.*

PROOF. We construct a digraph R on the vertices of D with $R^k = D$. Pick isomorphisms $\varphi_i \colon D_1 \to D_i$, $1 \le i \le k$ (φ_1 being simply the identity). For each vertex a of D_1 we let R contain the path

$$\varphi_1(a) \xrightarrow{R} \varphi_2(a) \xrightarrow{R} \cdots \xrightarrow{R} \varphi_k(a) \tag{16}$$

and additionally the arcs

$$\varphi_k(a) \xrightarrow{R} \varphi_1(b) \quad \text{for all } b \in D_1(a). \tag{17}$$

Figure 5 shows a local picture of this construction.

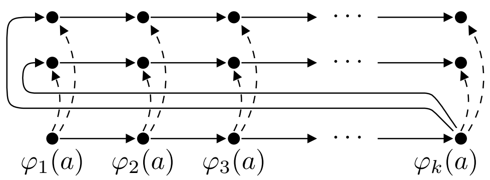


FIGURE 5. Constructing a kth root (continuous lines) for a disjoint union of k isomorphic digraphs (dashed lines).

We claim that $R^k = D$. To see this, pick any $v \in D_i$, $1 \le i \le k$, and compute

$$\begin{aligned}
R^k(v) &= R^i \varphi_k \varphi_i^{-1}(v) && \text{by (16)}\\
&= R^{i-1} D_1 \varphi_i^{-1}(v) && \text{by (17)}\\
&= \varphi_i D_1 \varphi_i^{-1}(v) && \text{by (16)}\\
&= D_i(v) = D(v),
\end{aligned}$$

treating digraphs and isomorphisms equally as mappings between subsets of the vertex set. $\square$

From roots to isomorphisms. Note how the root arcs in the above construction encode the isomorphism between the components of the digraph D. Our goal is to show that for a subdivision digraph, *any* root establishes isomorphisms between the weakly connected components of this digraph in exactly the same way. Before we can embark on this venture, however, we have to take care of some degenerate cases that do not fit into this picture.

Usually in a subdivision digraph one can easily distinguish the original vertices, sometimes called *branching vertices*, from the newly inserted *subdivision vertices*. In fact, a subdivision digraph is obviously bipartite and as soon as every weakly connected component contains at least one vertex whose indegree or outdegree differs from 1, the two classes can be uniquely identified.

A problem arises with subdivision digraphs that contain isolated cycles. In such components, all vertices look like subdivision vertices and this absence of clearly identifiable branching vertices leads to untypical behavior with respect to root finding. Fortunately, isolated cycles are simple objects and we can completely describe their powers.

5. Lemma. *The* kth *power of an isolated cycle of length* r *is the disjoint union of* $\gcd(r,k)$ *isolated cycles of length* $r/\gcd(r,k)$.

Proof. For every vertex x on an isolated cycle C, the sets $C^k(x)$ and $\bar{C}^k(x)$ are singletons. So each vertex of C^k has in– and outdegree 1, that means, C^k is the disjoint union of isolated cycles and by symmetry, all these cycles are of the same length. To determine this common length, start at an arbitrary vertex a and walk around C until you first reach a again in a multiple ℓ of k steps. Clearly, ℓ is the least common multiple of r and k; so the length of a cycle in C^k is

$$\frac{\ell}{k} = \frac{\operatorname{lcm}(r,k)}{k} = \frac{r}{\gcd(r,k)}. \qquad \square$$

As a consequence of Lemma 5, isolated cycles cannot have the isomorphism property we are looking for. But this is no problem. We shall show later that any vertex on an isolated cycle of a subdivision digraph D must also lie on an isolated cycle in any root of D. Thus, with respect to roots, cycle vertices do not interact with vertices from the other components of a subdivision digraph and we may in the following restrict our attention to subdivision digraphs without isolated cycles.

Ignoring isolated cycles we can show that subdivision digraphs bear the desired isomorphism structure—under the unfortunately indispensable additional condition that each vertex has at least one inneighbor and one outneighbor. We shall prove the following theorem.

6. Theorem. *A subdivision digraph without isolated cycles and with positive minimal indegree and outdegree has a* kth *root if and only if it is the disjoint union of* k *isomorphic digraphs.*

The basic idea for the proof of Theorem 6 is to show that in any kth root of a subdivision digraph, subdivision vertices and branching vertices appear in blocks of length k. More precisely, we will show that any subdivision vertex of D lies on an R-path of length k that consists only of subdivision vertices (of D) and likewise for branching vertices.

A direct proof of this statement, however, appears quite difficult since "subdivision vertex" is a semantic concept depending on the global structure of the digraph. Therefore we work with the simple local properties of subdivision vertices that can easily be dealt with.

7. Definition. We call a vertex of a digraph *thin* if its indegree and outdegree are 1; otherwise we call it *proper*.

The second step in our analysis will be to identify root arcs that are unique for their incident vertices, thus establishing unique correspondences that will be needed to identify the sought-after isomorphisms.

8. Definition. We call an arc ab of a digraph R *strong* if no further arcs leave a or enter b, i.e., $R(a) = \{b\}$ and $\bar{R}(b) = \{a\}$. More generally, a walk is called *strong* if all of its arcs are strong.

Most of the forthcoming proofs will be indirect, leading to contradictions to the following trivial observation about subdivision digraphs, which expresses the simple fact that digraphs, as we define them, cannot have parallel edges.

9. Observation. *No two vertices in a subdivision digraph have a common inneighbor and a common outneighbor.* $\square$

A general remark to avoid confusion. As before, we shall deal with two different digraphs on the same vertex set. When we talk about subdivision and branching vertices or thin and proper vertices, these notions shall always refer to (the arcs of) the subdivision digraph D. On the other hand, the term "strong" will always refer to arcs of the root R.

For technical reasons we provide the lemmas about unique arcs first and construct the long paths afterwards, since the latter rely on the former. Here is our first criterion for strongness of root arcs:

10. Lemma. *In a root R of a subdivision digraph D, any R-arc between two D-thin vertices is strong.*

Proof. Consider any pair a, b of D-thin vertices with $a \to b$ in R. As a thin vertex, a must also have at least one outneighbor in R, so assume for contradiction that $\deg^+_R(a) > 1$, i.e., there exists some $c \neq b$ with $a \to c$ in R. By symmetry, the case $\deg^-_R(b) \neq 1$ reduces to this situation by reversing all arcs.

The unique vertex u in $\bar{R}^{k-1}(a)$ has at least two D-outneighbors, b and c. Hence, this u is proper and therefore c is thin. So b and c are both thin and the sets $R^{k-1}(b)$ and $R^{k-1}(c)$ must therefore be nonempty. From $R^{k-1}(b) \cup R^{k-1}(c) \subseteq R^k(a)$ we thus conclude that $R^{k-1}(b) = R^{k-1}(c) = \{v\}$, where v is the unique D-outneighbor of a. Altogether, we have found two vertices, b and c, with common in– and common outneighbors—a contradiction to Observation 9. $\square$

One could actually relax the preconditions in Lemma 10 but its present form is sufficient for our purposes and it will fit quite naturally into its later applications.

There is an analog of Lemma 10 for proper vertices but it requires an explicit minimal-degree condition that was trivially met by thin vertices. Actually there can be non-strong arcs between pairs of proper vertices. So it is in the following lemma where the additional degree condition of Theorem 3 enters.

11. LEMMA. *In a root R of a subdivision digraph D, any R-arc between two D-proper vertices that have each at least one in– and one outneighbor is strong.*

PROOF. Consider any pair a, b of D-proper vertices with $a \to b$ in R. Assume for contradiction that there exists some $c \neq b$ with $a \to c$ in R. Again, the case $\deg^-_R(b) > 1$ reduces to this situation. Since a has a D-inneighbor, the set $\bar{R}^{k-1}(a)$ is nonempty. But any vertex from this set is an inneighbor of two vertices, one of which is proper. An impossible configuration in a subdivision digraph. □

The preceding two lemmas provide us with a simple procedure to identify R-walks of D-thin or D-proper vertices. Starting from a thin vertex a_0 of D, we check whether there is some D-thin outneighbor a_1 of a_0 in R. If such an a_1 exists it must be unique by Lemma 10. Next check for a D-thin outneighbor a_2 of a_1 and iterate this process until some ultimate a_t has no further D-thin outneighbors in R. Likewise we may search for inneighbors, altogether constructing a unique maximal R-walk of D-thin vertices containing a_0—provided we don't run into cycles. Analogously, we can find unique maximal walks of proper vertices.

We have now all necessary prerequisites to prove that thin vertices and proper vertices come in blocks.

12. LEMMA. *Let R be a* kth *root of a subdivision digraph D and let $a_0 \to a_1 \to \cdots \to a_\ell$ be an R-walk of length $\ell \leq k$ between two D-thin vertices a_0 and a_ℓ. Then all intermediate a_i, $0 < i < \ell$, are also thin.*

PROOF. We pick an arbitrary index j between 0 and ℓ and show that a_j is a thin vertex. Therefore first observe that the sets $R^k(a_j)$ and $\bar{R}^k(a_j)$ are nonempty because a_0 and a_ℓ are thin. We assume for contradiction that a_j is a proper vertex, so one of those two sets must contain at least two elements. By symmetry assume that $|R^k(a_j)| > 1$; so let x, y be two different elements from this set.

Denote the unique vertex in $R^k(a_0)$ by v. Since $R^{k-j}(a_j), R^{k-\ell}(a_\ell) \subseteq R^k(a_0)$, we get precisely

$$R^{k-j}(a_j) = \{v\} = R^{k-\ell}(a_\ell).$$

The first identity tells us that from a_j the two vertices $x, y \in R^k(a_j)$ are only reachable via v, i.e., $x, y \in R^j(v)$, and together with the second identity this implies

$$x, y \in R^{k-\ell+j}(a_\ell). \tag{18}$$

See Figure 6.

Since a_ℓ is thin, the set $R^k(a_\ell)$ contains exactly one vertex, w, say. Thus, by (18), we have $R^{\ell-j}(x) \cup R^{\ell-j}(y) \subseteq \{w\}$. As neighbors of the proper vertex

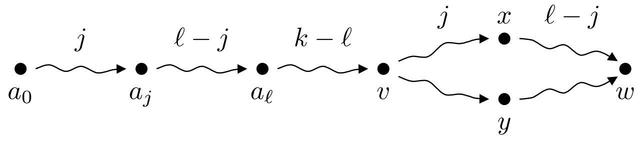


FIGURE 6. Path construction from the proof of Lemma 12.

a_ℓ the vertices x and y must be thin, so the sets $R^{\ell-j}(x)$ and $R^{\ell-j}(y)$ are nonempty and we actually get $R^{\ell-j}(x) = R^{\ell-j}(y) = \{w\}$, which implies $R^k(x) = R^k(y)$. Altogether, x and y have the common D-inneighbor a_ℓ and also a common outneighbor, in contradiction to Observation 9. $\square$

13. LEMMA. *Let R be a kth root of a subdivision digraph D and let $a_0 \to a_1 \to \cdots \to a_\ell$ be an R-walk of length $\ell \le k$ between two D-proper vertices a_0 and a_ℓ. Then all intermediate a_i, $0 < i < \ell$, are also proper.*

PROOF. Assume for contradiction that some a_j is a thin vertex. Then $R^k(a_j)$ is nonempty, so we may pick some $u \in R^{k-j}(a_j)$ together with some R-walk P of length $k-j$ from a_j to u. As a D-outneighbor of the proper vertex a_0 the vertex u is thin. Thus, by Lemma 12, all vertices on the walk P are in fact thin and Lemma 10 then implies that this walk is strong. Therefore the set $R^{\ell-i}(a_i)$ contains exactly one vertex, which can only be a_ℓ. But this vertex was assumed to be proper. $\square$

The proofs of Lemmas 10 through 13 show very graphically how the local properties of subdivision digraphs are exploited on the way to Theorems 3 and 6. They all employ a kind of squeezing technique along R-paths, leading to the unique identification of certain vertices or a contradiction involving too many neighbors of a subdivision vertex.

Combining the homogeneous paths provided by Lemmas 12 and 13 with the uniqueness statements of Lemmas 10 and 11, we are now able to construct isomorphisms from roots.

PROOF OF THEOREM 6. We already know from Proposition 4 that the disjoint union of k isomorphic digraphs has a kth root. So it remains to decompose D into k isomorphic subgraphs $D_1, \ldots, D_k$ and to provide isomorphisms between them. We do this by partitioning the whole vertex set into blocks of size k, such that each block contains exactly one vertex from each D_i.

For each proper vertex a of D, determine the maximal R-walk P_a through a that consists entirely of D-proper vertices, as described in connection with Lemmas 10 and 11. Such a walk cannot extend indefinitely, precisely, it consists of at most k vertices because all vertices at distance k from a proper vertex are thin. On the other hand, P_a must have at least k vertices because otherwise its thin neighbors would, by Lemma 12, force all its vertices to be thin, too.

For a thin vertex b we proceed similarly. Determine the maximal R-walk Q_b through b that consists entirely of D-thin vertices. Again, such a walk is bounded by some proper vertices to its left and right because otherwise we would get a cycle of thin vertices, which we excluded in the statement of the theorem. As in the case of proper vertices, the length of Q_b is at least $k-1$ (i.e., it contains at least k vertices) because by Lemma 13 the proper neighbors at the two ends must be at least $k+1$ steps apart. To determine its exact length, we turn back to the original concept of subdivision and branching vertices. Observe that by what we already know about proper vertices, Q_b is adjacent to a sequence of k branching vertices at each end. Hence, the first k and also the last k vertices of Q_b must be subdivision vertices of D. The next k vertices, on either end of Q_b, are then by definition branching vertices again, followed by another sequence of k subdivision vertices, etc. Clearly, this pattern only works out even if Q_b contains exactly $(2t+1)k$ vertices, for some nonnegative integer t.

We then subdivide all paths Q_b into paths of size k so that afterwards each vertex v of D lies on a unique strong path P_v of k thin respectively proper vertices and any two such paths P_b, P_c are either vertex disjoint or identical.

The obvious idea to identify isomorphic subgraphs now, is to put each vertex v of D into the subgraph D_i that corresponds to the position of v on the path Q_v, i.e., the ith vertex goes into D_i. The sought-after isomorphisms $\phi_{ij}\colon D_i \to D_j$ are also induced by the partition. Simply let ϕ_{ij} map a vertex $v \in D_i$ to the unique vertex of D_j that lies on the path Q_v. Clearly this mapping is well-defined. In order to check that it is also an isomorphism, we essentially only have to revisit the proof of Proposition 4, which constructed a root from isomorphisms. The crucial observation is again the strongness of our paths. Any walk of length k in R passes exactly once from one path P_a to a some path P_b with $a \to b$ in D, the remaining $k-1$ steps using only strong arcs. From this correspondence we see immediately that two vertices from the same path P_a have D-neighbors in the same set of adjacent paths. $\square$

For computational purposes we note the following simple reformulation of Theorem 6.

14. Corollary. *Let D be a subdivision digraph without isolated cycles and with positive minimal indegree and outdegree. Let further $D_1, \ldots, D_m$ be the different isomorphism classes of weakly connected components appearing in D and let d_i count the components in D isomorphic to D_i, $1 \le i \le m$. Then D has a kth root if and only if $k|d_i$ for all $i \in \{1, \ldots, m\}$.* $\square$

Counting cycles. We already discovered in Lemma 5 that powers of cycles are again cycles. To justify our hitherto ignorance towards cycles, we now also establish the converse: cycles have cycles as roots.

15. Lemma. *All vertices that lie on isolated cycles of a subdivision digraph D also lie on isolated cycles in any root of D.*

Proof. Let R be some kth root of D. We show that for any vertex c on a D-cycle, the sets $R^i(c)$ and $\bar{R}^i(c)$, $1 \le i < k$, are all singletons. This means that two D-adjacent vertices are connected through a strong walk in R, which proves the lemma.

So assume for contradiction that there exist two different vertices x, y in $R^j(c)$, $1 \le j < k$. (For $\bar{R}$ the statement is completely symmetric to this case.) There exists some $u \in \bar{R}^k(x) \cap \bar{R}^k(y)$ because $\bar{R}^k(c)$ is nonempty. With two outneighbors in the subdivision digraph D, this u must be a branching vertex, hence, x and y are subdivision vertices. Therefore the sets $R^k(x)$ and $R^k(y)$ are nonempty and since $R^k(c)$ consists of exactly one vertex, we even have $R^{k-i}(x) = R^{k-i}(y)$, which now implies $R^k(x) = R^k(y) \neq \emptyset$. Hence, the two vertices x and y yield a contradiction to Observation 9. □

Lemma 5 told us that a single isolated root cycle yields only cycles of the same length in D. When we want to decide whether a collection of cycles in a given subdivision digraph D has a root, we may thus treat cycles of different lengths separately.

So assume that that D is the disjoint union of isolated cycles, all of a common length ℓ, and that R is a kth root of D. Let C be a cycle in R of some length r. We write

$$\ell = \prod p_i^{\ell_i}, \quad k = \prod p_i^{k_i}, \quad r = \prod p_i^{r_i}, \tag{19}$$

where $p_1, p_2, \ldots$ are the prime numbers. Lemma 5 tells us $r = \ell \cdot \gcd(r,k)$; expressed in terms of prime factorizations this reads $r_i = \ell_i + \min\{r_i, k_i\}$, which yields the implications

$$\ell_i > 0 \quad \Rightarrow \quad r_i = \ell_i + k_i, \tag{20}$$

$$\ell_i = 0 \quad \Rightarrow \quad 0 \leq r_i \leq k_i. \tag{21}$$

So the length r of the root cycle C is determined up to the order r_i at p_i for those indices i that satisfy $\ell_i = 0$ and $k_i > 0$.

We now argue that for root checking we may restrict our attention to root cycles with $r_i = 0$ in (21). Assume that some root cycle C of length r has $r_j > 0$ for some index j with $\ell_j = 0$. Replace C by $p_j^{r_j}$ many cycles of length

$$r' := \frac{r}{p_j^{r_j}} = \prod_{i \neq j} p_i^{r_i}$$

each. One easily checks $r'/\gcd(r',k) = r/\gcd(r,k)$ to see that the new cycles together have the same kth power as the old cycle C. Hence, the new digraph is also a root of D. By repeating this transformation until all root cycles satisfy $r_i = 0$ in (21) for all primes, we may assume that all cycles in R have the same (minimal) length

$$r = \prod_{l_i > 0} p_i^{\ell_i + k_i}.$$

How many D-cycles of length ℓ does one R-cycle of length r give? By Lemma 5 this number is exactly

$$\gcd(r,k) = \prod p_i^{\min\{r_i,k_i\}} = \prod_{\ell_i > 0} p_i^{\min\{\ell_i + k_i, k_i\}} = \prod_{\ell_i > 0} p_i^{k_i}.$$

This shows that a disjoint union of m cycles of length ℓ has a kth root if and only if

$$\prod_{\ell_i > 0} p_i^{k_i} \text{ divides } m, \tag{22}$$

where ℓ_i and k_i are the orders of ℓ resp. k at p_i as defined in (19).

16. PROPOSITION. *Given a subdivision digraph D that consists of isolated cycles only and a parameter $k \geq 2$, we can check in polynomial time whether D has a kth root.*

PROOF. We sum up the results of the preceding discussion in a simple algorithm. For each integer ℓ that appears as the length of a cycle in D, compute the prime factorization $\ell = \prod p_i^{\ell_i}$ and then the order k_i of k at each prime p_i with positive ℓ_i, i.e., the maximal k_i so that $p_i^{k_i} | k$. The digraph D has a kth root iff (22) is satisfied for each length ℓ (the integer m there counting the number of length-ℓ cycles).

The ℓ_i can be obtained in polynomial time since ℓ is bounded by the size of D and the relevant k_i are determined efficiently by simple division, even if k is exponential in the input size. □

Reducing isomorphism to subdivision roots. It remains to merge the results of the preceding sections into a proof of our isomorphism-completeness theorem. We now give the details of both polynomial-time reductions between digraph-isomorphism and subdivision-digraph roots.

PROOF OF THEOREM 3. Let us first show that digraph roots are no easier to compute than digraph isomorphism, by giving a many-one reduction from the latter problem to the former.

For a given pair D_1, D_2 of digraphs, we construct a subdivision digraph D as follows.

(i) Make $k-2$ isomorphic copies $D_3, \ldots, D_k$ of D_2

(ii) Extend each D_i, $1 \leq i \leq k$, to a digraph D'_i by adding two new "super vertices" s_i, t_i, introducing the double connections $s_i \to a \to s_i$ for each $a \in D_i$, equipping t_i with a self-loop $t_i \to t_i$, and attaching it via $s_i \to t_i$.

(iii) Form the complete subdivision D''_i of each extended D'_i.

(iv) Let $D := D''_1 \,\dot\cup\, D''_2 \,\dot\cup \cdots \dot\cup\, D''_k$ be the disjoint union of the D''_i.

Clearly D is a subdivision digraph and the vertices s_i guarantee that it has positive minimal in– and outdegree and consists of exactly k components, none of which is an isolated cycle. Hence, Theorem 6 tells us that D has a kth root iff all D''_i are isomorphic or, equivalently, all D'_i are isomorphic. Since the t_i are distinguishable from all other vertices in the respective D'_i (because they are the only self-looped vertices with outdegree 1) this is the case iff all D_i are isomorphic or, by step (i), simply iff $D_1 \simeq D_2$.

We turn to the other reduction from subdivision-digraph roots to digraph isomorphism, which, by means of Proposition 16 and Theorem 6, is now very easy to formulate; but only as a Turing reduction, as opposed to the stronger notion of many-one reduction. That is, we describe a polynomial-time algorithm for the subdivision-digraph-root problem that may use a digraph-isomorphism oracle arbitrarily often.

Given a subdivision digraph D with positive minimal in– and outdegree, together with an integer k, we first use Proposition 16 to test in polynomial time whether the union of all isolated cycles of D has a kth root. Then we group the non-cycle components of D into isomorphism classes and apply Corollary 14. The independent treatment of isolated cycles and non-cycle components was justified by Lemmas 5 and 15. □

Outlook. While the original problem, the open complexity status of Boolean matrix root computation, is now settled, our search for further structure has lead to new questions. First of all, it would be desirable to get rid of the degree condition in Theorem 3. Let us indicate what can happen in a subdivision digraph that contains vertices without in– or outneighbors. Figure 7 shows such a digraph D together with a square root R. The two final root arcs can touch each other because the topmost vertex has no outneighbor and Lemma 11 about strong root arcs does not apply. Consequently, the minimal-degree condition is in fact indispensable for Theorem 6. But could it still be possible to remove it from the complexity result of Theorem 3? Observe that instead of being the disjoint union of two isomorphic subgraphs, the digraph D in Figure 7 can be decomposed into two parts, A and B (the former consisting of the two paths on the left, the latter containing the remaining five vertices), such that there exists a surjective *homomorphism* (i.e., an arc-preserving map) from A onto B. This homomorphism corresponds exactly to those arcs of R that go from A to B.

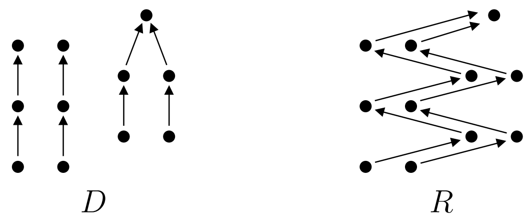


FIGURE 7. Dropping the degree condition in Theorem 6.

Though the general situation seems more difficult to analyze, this simple example indicates that when the degree condition is dropped, we have to deal with several interacting homomorphism problems. Thus, it is not at all clear whether the relaxed digraph root problem remains isomorphism complete since the general homomorphism problem for graphs is NP-complete [**21**]. (3-Colorability can be written as a homomorphism problem, for example).

More generally, we might ask for stronger versions of Theorem 3 showing isomorphism completeness of root finding for larger classes of digraphs. Although the structural result of Theorem 6 requires the special appearance of subdivision digraphs, their strict regularity should not ultimately be needed to deactivate the computationally hard aspects of the root problem established through Theorem 1. Yet, the concept of subdivisions and the techniques we employed throughout the proofs of Lemmas 10 to 13 might serve as a guideline for such generalizations.

Bibliography

[1] A. V. Aho, M. R. Garey, and J. D. Ullman. The transitive reduction of a directed graph. *SIAM Journal on Computing*, 1(2):131–137, 1972.

[2] P. Arnoux. Strumian sequences. In V. Berthé, S. Ferenczi, C. Mauduit, and A. Siegel, editors, *Substitutions in Dynamics, Arithmetics and Combinatorics*, volume 1794 of *LNM*, chapter 6, pages 143–198. Springer, 2002.

[3] Jørgen Bang-Jensen and Gregory Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer, 2001.

[4] József Beck. Van der Waerden and Ramsey type games. *Combinatorica*, 2:103–116, 1981.

[5] József Beck. On positional games. *Journal of Combinatorial Theory, Series A*, 30:117–131, 1991.

[6] József Beck. Deterministic graph games and a probabilistic intuition. *Combinatorics, Probability and Computing*, 3:13–26, 1994.

[7] József Beck. Tic-Tac-Toe. In Béla Bollobás, editor, *Contemporary Combinatorics*, number 10 in Bolyai Society Mathematical Studies, pages 93–137. Springer, 2002.

[8] Elwyn R. Berlekamp, Hohn H. Conway, and Richard K. Guy. *Winning Ways for your mathematical plays*, volume 2: Games in Particular. Academic Press, 1982.

[9] Kellogg S. Booth. Isomorphism testing for graphs, semigroups, and finite automata are polynomially equivalent problems. *SIAM Journal on Computing*, 7(3):273–279, 1978.

[10] Glen E. Bredon. *Topology and Geometry*. Springer, 1993.

[11] John H. Conway. The angel problem. In Richard Nowakowski, editor, *Games of No Chance*, volume 29 of *MSRI Publications*, pages 3–12. 1996.

[12] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*, chapter 26. MIT Press, 1990.

[13] Bart de Schutter and Bart de Moor. On the sequence of consecutive powers of a matrix in a Boolean algebra. *SIAM Journal on Matrix Analysis and Applications*, 21(1):328–354, 1999.

[14] Eric D. Demaine. Playing games with algorithms: Algorithmic combinatorial game theory. In Jiří Sgall, Aleš Pultr, and Petr Kolman, editors, *Proceedings of the 26th Symposium on Mathematical Foundations in Computer Science*, volume 2136 of *LNCS*, pages 18–32, Mariánské Lázně, Czech Republic, 2001.

[15] Paus Erdös and J. L. Selfridge. On a combinatorial game. *Journal of Combinatorial Theory, Ser. A*, 14:298–301, 1973.

[16] David Gale and Frank M. Steward. Infinite games with perfect information. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games II*, volume 28 of *Annals of Mathematics Studies*, pages 245–266. Princeton University Press, 1953.

[17] Martin Gardner. Mathematical games. *Scientific American*, 230(2):106–108, 1974.

[18] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, 1979.

[19] A.W. Hales and R.I. Jewett. Regularity and positional games. *Transactions of the American Mathematical Society*, 106:222–229, 1963.

[20] Frank Harary. Achievement and avoidance games for graphs. *Annals of Discrete Mathematics*, 13:111–120, 1982.

[21] Pavol Hell and Jaroslav Nešetřil. On the complexity of H-coloring. *Journal of Combinatorial Theory, Series B*, 48:92–110, 1990.

[22] Nicholas J. Higham. Newton's method for the matrix square root. *Mathematics of Computation*, 46(174):537–549, 1986.

[23] Thomas Jech. *Set Theory*. Academic Press, 1978.

[24] C. R. Johnson, K. Okubo, and R. Reams. Uniqueness of matrix square roots and an application. *Linear Algebra and Applications*, 323:52–60, 2001.

[25] Volker Kaibel and Alexander Schwartz. On the complexity of polytope isomorphism problems. *Graphs and Combinatorics*, 19(2):215–230, 2003.

[26] Ki Hang Kim. *Boolean matrix theory and applications*. Marcel Decker, Inc., 1982.

[27] Johannes Köbler, Uwe Schöning, and Jacobo Torán. *The Graph Isomorphism Problem*. Birkhäuser, 1993.

[28] Martin Kutz. The complexity of Boolean matrix root computation. In T. Warnow and B. Zhu, editors, *Computing and Combinatorics, 9th Ann. Intern. Conf., COCOON 2003, Proceedings*, volume 2697 of *LNCS*, pages 212–221, Big Sky, MT, USA, 2003.

[29] Ya Yan Lu. A Padé approximation method for square roots of symmetric positive definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 19(3):833–845, 1998.

[30] Anna Lubiw. Some NP-complete problems similar to graph isomorphism. *SIAM Journal on Computing*, 1981.

[31] George Markowsky. Ordering D-classes and computing Shein rank is hard. *Semigroup Forum*, 44:373–375, 1992.

[32] Donald A. Martin. Borel determinacy. *Annals of Mathematics*, 102(2):363–371, 1975.

[33] Rudolph Mathon. A note on the graph isomorphism counting problem. *Information Processing Letters*, 8(3):131–132, 1979.

[34] Rajeev Motwani and Madhu Sudan. Computing roots of graphs is hard. *Discrete Applied Mathematics*, 54(1):81–88, 1994.

[35] Ian Munro. Efficient determination of the transitive closure of a directed graph. *Information Processing Letters*, 1:56–58, 1971.

[36] Panayiotis J. Psarrakos. On the mth roots of a complex matrix. *The Electronic Journal of Linear Algebra*, 9:32–41, 2002.

[37] Stephan Reisch. Gobang ist PSPACE-vollständig. *Acta Informatica*, 13:59–66, 1980.

[38] Kenneth A. Ross and Charles R. B. Wright. *Discrete Mathematics*, chapter 7.5. Prentice-Hall, second edition, 1988.

[39] Thomas J. Schaefer. On the complexity of some two-person perfect-information games. *Journal of Computer and System Science*, 16:185–225, 1978.

[40] Larry L. Stockmeyer. The minimal set basis problem is NP-complete. IBM Research Report No. RC-5431, IBM Thomas J. Watson Research Center, 1975.

Zusammenfassung

In dieser Dissertation betrachten wir neue Strategien für ein unendliches kombinatorisches Spiel, untersuchen die Komplexität einer Klasse von Spielen auf Hypergraphen und beantworten eine offene Frage zu einem Berechnungsproblem auf gerichteten Graphen. Obwohl gewisse Verbindungen zwischen den behandelten Gebieten bzw. den prinzipiellen Fragestellungen bestehen, handelt es sich um drei unabhängige Themen.

Das Engel-Problem. Auf einem unendlichen Schachbrett versucht der *Engel*, eine "Schachfigur" mit beschränkter Schrittweite, seinem Gegner, dem *Teufel*, unendlich lange davonzulaufen. Der Teufel blockiert Zug um Zug Felder des Brettes mit der Absicht, den Engel einzukreisen. Es ist ein offenes Problem, ob ein Engel mit hinreichend großen Schritten gewinnen kann. Wir verbessern eine bekannte Teufel-Strategie und zeigen außerdem, dass der Engel auf einem dreidimensionalen Brett entkommt.

Positionelle Spiele auf Hypergraphen. Zwei Spieler wählen abwechselnd Ecken eines Hypergraphen bis alle Ecken vergeben sind. Der Anziehende gewinnt, wenn es ihm gelingt, eine Kante des Hypergraphen zu vervollständigen, andernfalls siegt sein Gegner. Diese asymmetrische Verallgemeinerung des bekannten Spiels Tic-Tac-Toe wird als *schwaches positionelles Spiel* bezeichnet und ist PSPACE-vollständig. Der entsprechende Härtebeweis verwendet Kanten mit bis zu 11 Ecken. Wir versuchen Spiele auf Hypergraphen mit nicht mehr als drei Ecken pro Kante vollständig zu lösen. Dies gelingt uns fast, mit der zusätzlichen Einschränkung, dass sich je zwei Kanten in höchstens einer Ecke treffen dürfen. Mittels einer vollständigen Klassifizierung in Gewinner und Verlierer erhalten wir einen Polynomialzeit-Algorithmus, der solche Spiele optimal spielt.

Wurzeln gerichteter Graphen. Interpretiert man eine quadratische Boolesche 0/1-Matrix als Adjazenzmatrix eines gerichteten Graphen, so induziert die Matrixmultiplikation auf natürliche Weise eine Potenzoperation auf gerichteten Graphen. Wir zeigen, dass in diesem Sinne das Berechnen von Wurzeln gerichteter Graphen NP-vollständig ist. In einem zweiten Teil stellen wir eine Beziehung zwischen solchen Wurzeln und Graphisomorphie her und zeigen, dass für eine spezielle Klasse von gerichteten Graphen das Wurzelproblem von derselben Komplexität ist wie das Isomorphie-Problem für Graphen.