THE MINIMUM DEPTH OF OPEN-BOUNDARY YANG–BAXTER INTEGRABLE CIRCUITS: AN EXACT FORMULA AND TWO ERRATA FOR ARXIV:2607.02093

KOREA SUPERINTELLIGENCE LABS

Abstract. An open-boundary integrable circuit in the construction of García Fernández, Paletta and Retore (arXiv:2607.02093) is a fixed word in two-site gates and two boundary gates, determined by the set $\vec{n} \subseteq \{1, \ldots, N\}$ of sites carrying the inhomogeneity $-\kappa$. They ask for the minimum depth of the circuit over all $\vec{n}$ with $|\vec{n}| = \kappa_-$ and conjecture $\frac{1}{2}(N+3) - \kappa_-$ for odd N and $\frac{1}{2}(N+4) - \kappa_-$ for even N. Both conjectures are false: at $N = 8$, $\kappa_- = 2$ the placement $\vec{n} = (6, 3)$ has depth 3 against the conjectured 4, and at $N = 11$, $\kappa_- = 2$ the placement $\vec{n} = (8, 4)$ has depth 4 against 5; $N = 8$ lies inside the range $N = 2, \ldots, 10$ the source says it studied, and both lie well inside the $N \le 51$ (odd), $N \le 50$ (even) to which it reports having checked the conjectures. The minimum is $\lceil (N+1)/(\min(\kappa_-, N-\kappa_-)+1) \rceil$ for every $N \ge 2$ and $0 \le \kappa_- \le N$: the lower bound is a chain argument on the word, the upper bound an explicit placement whose $+\kappa$ sites are spread over $\kappa_- + 1$ staircases rather than the two of the conjectured placements. The minimum was also computed exhaustively over all 2^N placements for $2 \le N \le 19$, and the explicit placement was checked to attain the formula for every $N \le 120$. For the circuits with inhomogeneities $+\kappa$ and ρ only, the inequality $d \ge N+1$ that the source verified exhaustively for $N = 5, 6, 7, 9$ and by sampling for $N = 11, 13, 15$ has a one-line proof for every N, and the minimum is verified here to be exactly $N+1$, exhaustively over all 2^{N-1} placements, for every $3 \le N \le 20$; a product in the source's Eq. (73) is printed in the wrong order. Every finite statement here — the depth semantics, the tables, the two counterexamples, the placements to $N = 120$, the $(+\kappa, \rho)$ sweeps and the product-order check — is verified in Lean 4 without Mathlib; the closed form for all N is proved on paper.

1. Introduction

García Fernández, Paletta and Retore [1] build Yang–Baxter integrable quantum circuits with open boundaries from the double-row transfer matrix of an open spin chain of N sites whose inhomogeneities $\theta_1, \ldots, \theta_N$ take two values $+\kappa$ and $-\kappa$. Their Lemma 1 and Lemma 2 (Eqs. (35) and (47) of [1]; every section, equation and figure number of [1] quoted in this paper is read off the compiled arXiv PDF of version 1) rewrite the transfer matrix $t(\kappa)$ as a product of two-site gates $U_{j,j+1}$ and two boundary gates, and their Theorems 1 and 2 (Eqs. (42), (53)) give that product explicitly in terms of the set $\vec{n}$ of sites carrying $-\kappa$. The construction extends to non-difference-form R-matrices the open-boundary integrable Trotterization of Vanicat [3]: Sec. 2.2.2 of [1] says "This generalises the construction of [30] for R-matrices of non-difference form", its reference [30] being [3]. That paper, with Vanicat, Zadnik and Prosen [4], introduced the R-matrix-independent Trotterization framework; the string "depth" occurs nowhere in either e-print, so the question of the least depth begins with [1]. The isospectrality of open-boundary circuits in which every gate acts once per period is cited in [1] from Bensa and Žnidarič [5] — its Sec. 3.1.3 reads "The following theorem generalizes the result of [44], where it was shown that, for open boundary conditions, all circuits in which each gate acts exactly once per period are isospectral", with [44] = [5] — and generalised by its Theorem 3. Circuits with the same $\kappa_- = |\vec{n}|$ are isospectral (their Theorem 3), so [1] takes as the canonical representative of each class the circuit of least *depth*, defined there (Sec. 3.1.3) as the number "of sequential gate layers required to implement the circuit". This raises a purely combinatorial question: given N and κ_-, what is

$$d_{\min}(N, \kappa_-) = \min\Big\{ \operatorname{depth} M(\vec{n}) : \ \vec{n} \subseteq \{1, \ldots, N\}, \ |\vec{n}| = \kappa_- \Big\} \, ?$$

Section 3.2.1 of [1] answers it by two conjectures, quoted here verbatim:

Conjecture 1 For odd N and $0 < \kappa_- \le \frac{N-1}{2}$, the minimum depth is given by $d = \frac{1}{2}(N+3) - \kappa_-$. (60)

Conjecture 2 For even N and $0 < \kappa_- \leq \frac{N}{2}$, the minimum depth is given by $d = \frac{1}{2}(N+4) - \kappa_-$. (62)

together with explicit placements, Eq. (61) $\vec{n} = (N-(d-1), N-(d+1), \ldots, d+2, d)$ for odd N and Eq. (63) $\vec{n} = (N-(d-2), N-d, \ldots, d+2, d)$ for even N, said to attain the conjectured depth. The evidence reported is: "We explicitly constructed quantum circuits for chain lengths $N = 2, \ldots, 10$ using Theorems 1 and 2, and subsequently verified these results for system sizes up to $N = 51$"; for Conjecture 1, "By investigating a number of cases ($N = 3, \ldots, 9$ and all possible positions for the inhomogeneities)" and "We checked this conjecture for odd N up to $N = 51$"; for Conjecture 2, "By studying $N = 2, \ldots, 10$" and "We checked this conjecture for even N up to $N = 50$".

What is settled here. Both conjectures are false, and the true minimum has a closed form.

(i) At $N = 8$, $\kappa_- = 2$ the placement $\vec{n} = (6,3)$ has depth 3 where Conjecture 2 predicts 4; at $N = 11$, $\kappa_- = 2$ the placement $\vec{n} = (8,4)$ has depth 4 where Conjecture 1 predicts 5 (Theorems 3.5 and 3.6, with the circuits and their schedules in Figures 1 and 2). These are the smallest counterexamples; the exhaustive minima agree with the conjectures exactly on odd $N \leq 9$ and even $N \leq 6$, and disagree at every N from 8 (even) and 11 (odd) up to the computed limit $N = 19$ (Table 1). The placements (61) and (63) do have the depths the source assigns them; what fails is minimality.

(ii) For every $N \geq 2$ and every $0 \leq \kappa_- \leq N$,

$$d_{\min}(N, \kappa_-) = \left\lceil \frac{N+1}{\min(\kappa_-, N-\kappa_-)+1} \right\rceil. \tag{1}$$

The proof (Theorem 4.1) has three parts: flipping every inhomogeneity's sign reverses the word and so preserves the depth, which is the $\kappa_- \leftrightarrow N - \kappa_-$ duality asserted in [1] after its Fig. 8; every run of consecutive $+\kappa$ sites contributes a chain of gates whose length bounds the depth from below; and an explicit placement, whose $+\kappa$ sites are spread over $\kappa_- + 1$ runs of nearly equal length, attains the bound. The conjectured placements (61), (63) force every interior run to a single site, which is the "staircase+brickwork+staircase" geometry the source describes in its Sec. 3.2.3; the optimum is $\kappa_- + 1$ staircases. The source did look outside that family: its Appendix D displays, at $N = 9$, an alternative set of minimum-depth placements and discards it because the regular gate picture cannot be kept beyond $N = 11$ — a statement about the picture, not about the minimum (Remark 3.8). Independently of the proof, (1) was verified exhaustively over all 2^N placements for $2 \leq N \leq 19$ (Theorems 3.1–3.3) and the explicit placement was checked to attain it for every $N \leq 120$ and every $\kappa_- \leq N$ (Theorem 4.2).

(iii) For the circuits with inhomogeneities $+\kappa$ and ρ only (Sec. 3.3.1 of [1]), the source remarks: "in all cases we checked with odd N and only $+\kappa$ and ρ (but no $-\kappa$), the depth was always $d \geq N + 1$. We checked this for all cases for $N = 5, 6, 7, 9$ and some randomly chosen cases for $N = 11, 13, 15$." The minimum over all 2^{N-1} placements of the ρ's is exactly $N + 1$ for every $3 \leq N \leq 20$ (Theorem 5.1); in particular the three sampled sizes are settled. The value itself is not deep — one chain gives $d \geq N + 1$ for every N and $\vec{n}^\rho = \emptyset$ attains it (Remark 5.2) — so what Theorem 5.1 contributes is the exhaustive, kernel-checked form of the source's own sampled statement, on the range the source named.

(iv) Eq. (73) of [1] prints the left product of the $(+\kappa, \rho)$ circuit as an increasing product where the source's own Lemma 1 gives a decreasing one; the two readings are not equivalent, and only the decreasing one reproduces the depth $2N$ that the source itself states for the circuit of its Fig. 14 (Theorem 5.3).

The errata (i) and (iv) are stated as errata, with the evidence; the conjectures were supported by the source's own exhaustive search up to $N = 9$ (odd) and, as far as the text says, by constructions up to $N = 10$ (even), and the failure begins at $N = 8$ and $N = 11$. We do not know how the source's Mathematica notebook [2] was used at the sizes $N \leq 51$ it mentions; the sentence "We checked this conjecture for odd N up to $N = 51$" follows the display of the placement (61), and the placement does have the stated depth. The deposited notebook `OpenQCforDiffGeom.nb` of [2] was read: it defines the Lemma 1 and Lemma 2 words (its `Ma`, `Mb`, selected by `MM[nn]:=If[MemberQ[listkappam,nn],Mb[nn],Ma[nn]]`, i.e. by whether site N carries $-\kappa$) and rewrites them into words of gates by the regularity rules (its `ruleU`: $\check{R}(\kappa,-\kappa)\mapsto U$, $\check{R}(\kappa,\rho)\mapsto V$, $\check{R}(\rho,-\kappa)\mapsto W$). That is exactly the model used here. It carries no depth function and no search over placements: its fifteen input cells evaluate M at the individual configurations of the source's figures, and the depth is then read off the drawing. So there is no depth for the notebook to report; it was not executed here.

Everything finite in this paper — the depth semantics, the exhaustive tables, the two counterexamples with their schedules, the explicit placements to $N = 120$, the $(+\kappa,\rho)$ sweeps and the product-order check — is verified in Lean 4 (Section 7); Theorem 4.1, the formula for all N, is proved on paper only and is labelled as such. Section 2 fixes the objects, Section 3 gives the tables and the counterexamples, Section 4 the closed form, Section 5 the $(+\kappa,\rho)$ circuits and the misprint, and Section 6 the controls.

2. Preliminaries

2.1. **Words of gates and their depth.** A *gate* is an integer interval $[a,b]$ of sites, its *support*; the bulk gate $U_{j,j+1}$ has support $[j,j+1]$, and the two boundary gates $K = K_1^R(\kappa)$ and $L = \tilde{K}_N^L(\kappa)$ of [1] have supports $[1,1]$ and $[N,N]$. Two gates *conflict* when their supports meet. A *word* is a finite sequence of gates $w = (g_1, g_2, \ldots, g_m)$ written in *operator order*, i.e. exactly as the product $M = g_1 g_2 \cdots g_m$ is printed; the last factor g_m acts first in time and g_1 last. We say g_j is *earlier* than g_i when $j > i$.

Definition 2.1. A *layering* of w is a map $f : \{1,\ldots,m\} \to \{1,2,\ldots\}$ such that $f(i) > f(j)$ whenever $j > i$ and g_i, g_j conflict. Its *number of layers* is $\max_i f(i)$. The *ASAP levels* of w are defined from the earliest gate backwards by

$$\ell(i) = 1 + \max\bigl\{\ell(j) :\ j > i,\ g_j \text{ conflicts with } g_i\bigr\}, \qquad \max\emptyset = 0,$$

and the *depth* of w is $\operatorname{depth} w = \max_i \ell(i)$ (with $\operatorname{depth}() = 0$).

A layering is precisely a way to run the circuit in sequential layers: gates in one layer have disjoint supports (two gates of the same layer cannot conflict, since one of them is earlier), and a gate waits for every earlier gate it conflicts with. So "the number of sequential gate layers required to implement the circuit" of [1], Sec. 3.1.3, is the least number of layers of a layering, and the first thing to check is that depth computes it.

Theorem 2.2 (depth is the minimum number of layers)**.** *For every word w and every layering f of w, $\operatorname{depth} w \le \max_i f(i)$; and ℓ is a layering of w with $\max_i \ell(i) = \operatorname{depth} w$.*

Proof. Induction on the word, from the earliest gate: $\ell(i) \le f(i)$ for every i, since every $j > i$ conflicting with g_i has $\ell(j) \le f(j) < f(i)$ by induction and the layering condition; and ℓ itself satisfies the layering condition by construction. This is `depth_le_of_layering` and `depth_attained` in Appendix A, proved without any evaluation by compiled code. □

The following reformulation, which is the form the lower bounds of Section 4 use, is not part of the formal development; its proof is two lines. A *chain* in w is a sequence of gates $g_{i_1}, g_{i_2}, \ldots, g_{i_k}$ with $i_1 > i_2 > \cdots > i_k$ (strictly increasing in time) in which consecutive members conflict.

Lemma 2.3. $\operatorname{depth} w$ *is the largest length of a chain in w. Consequently the depth of a word equals the depth of its reversal.*

Proof. By induction from the earliest gate, $\ell(i)$ is the largest length of a chain ending at g_i: a chain ending at g_i has as its penultimate member some earlier conflicting g_j, so its length is at most $\ell(j)+1 \le \ell(i)$; and a longest chain ending at the g_j that attains the maximum in the definition of $\ell(i)$ extends by g_i. Reversing the word maps chains to chains (reverse the sequence), so the longest chain has the same length. □

Definition 2.1 and Lemma 2.3 are the standard heap-of-pieces picture of Viennot [7]: a word of gates is a heap of pieces over the sites, two pieces are concurrent exactly when their supports are disjoint, and the depth is the height of the heap. Nothing below uses more than the two statements just proved.

2.2. **The circuits.** Let $\theta_1, \dots, \theta_N \in \{+\kappa, -\kappa, \rho\}$ with $\theta_N \neq \rho$. Write $\vec{n} = (n_{\kappa_-}, \dots, n_2, n_1)$, $1 \le n_1 < \cdots < n_{\kappa_-} \le N$, for the positions of the $-\kappa$'s (Eq. (33) of [1]), and use the ordered products of its Eqs. (31)–(32),

$$\overleftarrow{\prod}_{1\le i\le b} A_i = A_b \cdots A_2 A_1, \qquad \overrightarrow{\prod}_{1\le i\le b} A_i = A_1 A_2 \cdots A_b.$$

Lemma 1 of [1] (the case $n_{\kappa_-} < N$, i.e. $\theta_N = \kappa$; Eq. (35)) states

$$t(\kappa) = g(\kappa)\Big(\overleftarrow{\prod}_{1\le i\le N-1} \check{R}_{i,i+1}(\kappa,\theta_i)\Big) K_1^R(\kappa) \Big(\overrightarrow{\prod}_{1\le j\le N-1} \check{R}_{j,j+1}(\theta_j,-\kappa)\Big) \tilde{K}_N^L(\kappa),$$

and Lemma 2 (the case $n_{\kappa_-} = N$; Eq. (47)) the same product with $\tilde{K}_N^L(\kappa)$ moved to the front (and $g(-\kappa)$ in place of $g(\kappa)$). For a regular R-matrix, $\check{R}(u,u)$ is a scalar, so the factor at site i of the left product is a genuine two-site gate exactly when $\theta_i \neq \kappa$ and the factor at site j of the right product exactly when $\theta_j \neq -\kappa$; this is how Theorems 1 and 2 of [1] (Eqs. (42), (53)) read the two products as products of gates U_{n_r,n_r+1} over $n_r \in \vec{n}$ and $U_{j,j+1}$ over $j \notin \vec{n}$. Only the supports matter for the depth. Accordingly, for a configuration with $A = \{i \le N-1 : \theta_i = -\kappa\}$ and $B = \{1, \dots, N-1\} \setminus A$ the word of the circuit is

$$W_1(\theta) = \Big(\overleftarrow{\prod}_{i\in A} U_{i,i+1}\Big)\, K\, \Big(\overrightarrow{\prod}_{j\in B} U_{j,j+1}\Big)\, L \qquad (\theta_N = +\kappa,\ \text{Lemma 1}), \tag{2}$$

$$W_2(\theta) = L\, \Big(\overleftarrow{\prod}_{i\in A} U_{i,i+1}\Big)\, K\, \Big(\overrightarrow{\prod}_{j\in B} U_{j,j+1}\Big) \qquad (\theta_N = -\kappa,\ \text{Lemma 2}), \tag{3}$$

where $\overleftarrow{\prod}_{i\in A}$ lists the sites of A in decreasing order and $\overrightarrow{\prod}_{j\in B}$ those of B in increasing order. In time, W_1 runs L, then the B-gates in decreasing site order, then K, then the A-gates in increasing site order; W_2 is the same with L last instead of first. Every such word has exactly $N+1$ gates. For $\vec{n} \subseteq \{1, \dots, N\}$ we write $M(\vec{n})$ for the word of the configuration with $-\kappa$ exactly on $\vec{n}$, and

$$d_{\min}(N, \kappa_-) = \min\{\operatorname{depth} M(\vec{n}) :\ |\vec{n}| = \kappa_-\}.$$

The source's Fig. 6 shows the three circuits $\vec{n} = (4,2), (5,2), (2,1)$ at $N = 5$ with depths $2, 3, 4$; Definition 2.1 reproduces these values (`fig6_depths`), and for $\vec{n} = \emptyset$ gives the staircase of depth $N+1$ the source describes.

When a third value ρ is present at the sites $\vec{n}^\rho \subseteq \{1, \dots, N-1\}$ and every other site carries $+\kappa$ (Sec. 3.3.1 of [1]), Lemma 1 applies and the word is

$$W_\rho(\vec{n}^\rho) = \Big(\overleftarrow{\prod}_{i\in\vec{n}^\rho} V_{i,i+1}\Big)\, K\, \Big(\overrightarrow{\prod}_{1\le j\le N-1} Y_{j,j+1}\Big)\, L, \qquad Y_{j,j+1} = \begin{cases} W_{j,j+1}, & j \in \vec{n}^\rho, \\ U_{j,j+1}, & j \notin \vec{n}^\rho, \end{cases} \tag{4}$$

with V, W, U two-site gates (the names are those of [1]); the right product now has a gate at every site because no site carries $-\kappa$. Eq. (73) of [1] prints the first product as $\overrightarrow{\prod}$; Section 5 returns to this.

2.3. **Encoding.** In the formal development a placement $\vec{n} \subseteq \{1, \dots, N\}$ is a bit mask $m < 2^N$ with bit $i-1$ set iff $i \in \vec{n}$; `kkWord N m` is $M(\vec{n})$ (choosing (2) or (3) by bit $N-1$), `kkDepth N m` its depth, `pc N m` $= |\vec{n}|$, and `kkTable N` is the list $[d_{\min}(N,0), \dots, d_{\min}(N,N)]$ computed by folding over all 2^N masks. Likewise `rhoWord N m` is $W_\rho(\vec{n}^\rho)$ for the mask $m < 2^{N-1}$ of $\vec{n}^\rho$, `rhoMin N` the minimum of its depth over all masks and `rhoTable N` the list of minima indexed by $\rho_+ = |\vec{n}^\rho|$. Finally

$$\texttt{closedForm N k} = \Big\lfloor \frac{N + \min(k, N-k) + 1}{\min(k, N-k) + 1} \Big\rfloor = \Big\lceil \frac{N+1}{\min(k, N-k) + 1} \Big\rceil,$$

and `conj1 N k`, `conj2 N k` are $\lfloor (N+3)/2 \rfloor - k$ and $\lfloor (N+4)/2 \rfloor - k$, the right-hand sides of (60) and (62) at the parities where they are meant. The statements are listed verbatim in Appendix A.

3. The exhaustive minima and the two counterexamples

3.1. **The tables.** The following three statements are exhaustive computations: for each N, every one of the 2^N placements is built by (2)–(3) and scheduled by Definition 2.1, and the minimum over each κ_- is recorded. They are proved in Lean by `native_decide`, i.e. by running the compiled definitions and trusting the result (Section 7).

Proposition 3.1 (the source's range)**.** *For $2 \le N \le 10$ the rows of Table 1 are the exhaustive minima $d_{\min}(N, \kappa_-)$, $0 \le \kappa_- \le N$, and equal* (1)*. On this range Conjecture 1 holds for every odd N and every $0 < \kappa_- \le (N-1)/2$, and Conjecture 2 holds for $N = 2, 4, 6$ and every $0 < \kappa_- \le N/2$; $d_{\min}(N, 0) = N+1$ throughout.*

This range is the one the source states it searched ($N = 3, \ldots, 9$ "and all possible positions" for odd N; $N = 2, \ldots, 10$ for even N), so the values are the source's own data as far as it reports them; the certification from its Lemma 1/Lemma 2 words is ours, and it reproduces every individual depth the source prints (Fig. 6; the $\kappa_- = 0$ staircase).

Theorem 3.2 ($11 \le N \le 16$)**.** *For $11 \le N \le 16$ the rows of Table 1 are the exhaustive minima $d_{\min}(N, \kappa_-)$, $0 \le \kappa_- \le N$, and they equal* (1) *at every κ_-.*

Theorem 3.3 ($17 \le N \le 19$)**.** *For $N = 17, 18, 19$ ($2^{19} = 524\,288$ placements at $N = 19$) the rows of Table 1 are the exhaustive minima $d_{\min}(N, \kappa_-)$, $0 \le \kappa_- \le N$, and they equal* (1) *at every κ_-.*

Table 1. The exhaustive minimum depth $d_{\min}(N, \kappa_-)$ over all 2^N placements, for $0 \le \kappa_- \le \lfloor N/2 \rfloor$ (each row is a palindrome, $d_{\min}(N, \kappa_-) = d_{\min}(N, N - \kappa_-)$; Proposition 6.1). Every entry is kernel-checked (Proposition 3.1, Theorems 3.2, 3.3) and equals $\lceil (N+1)/(\kappa_- + 1) \rceil$. A star marks the entries where Conjecture 1 or 2 predicts a larger value; the conjectured value is $\lfloor (N+3)/2 \rfloor - \kappa_-$ (odd N) or $\lfloor (N+4)/2 \rfloor - \kappa_-$ (even N).

$N \backslash \kappa_-$	0	1	2	3	4	5	6	7	8	9
2	3	2								
3	4	2								
4	5	3	2							
5	6	3	2							
6	7	4	3	2						
7	8	4	3	2						
8	9	5	3^*	3	2					
9	10	5	4	3	2					
10	11	6	4^*	3^*	3	2				
11	12	6	4^*	3^*	3	2				
12	13	7	5^*	4^*	3^*	3	2			
13	14	7	5^*	4^*	3^*	3	2			
14	15	8	5^*	4^*	3^*	3^*	3	2		
15	16	8	6^*	4^*	4^*	3^*	3	2		
16	17	9	6^*	5^*	4^*	3^*	3^*	3	2	
17	18	9	6^*	5^*	4^*	3^*	3^*	3	2	
18	19	10	7^*	5^*	4^*	4^*	3^*	3^*	3	2
19	20	10	7^*	5^*	4^*	4^*	3^*	3^*	3	2

Remark 3.4 (beyond the formal range). An independent C implementation of Definition 2.1 and of (2)–(3) (140 lines; 4.2 s for all $2 \le N \le 24$) reproduces Table 1 and continues it to $N = 24$, where every one of the 322 entries with $2 \le N \le 24$, $0 \le \kappa_- \le N$ again equals (1); for example the $N = 24$ row is $25, 13, 9, 7, 5, 5, 4, 4, 3, 3, 3, 3, 2, \ldots$. These five further rows are outside the formal development and are not used anywhere below; Theorem 4.1 makes them redundant.

3.2. The counterexamples.

Theorem 3.5 (Conjecture 1 is false). *At $N = 11$, $\kappa_- = 2$ (so $0 < \kappa_- \leq (N-1)/2 = 5$), the placement $\vec{n} = (8, 4)$ gives, by Theorem 1 of* [1], *the word*

$$M = U_{8,9}\, U_{4,5}\, K\, U_{1,2}\, U_{2,3}\, U_{3,4}\, U_{5,6}\, U_{6,7}\, U_{7,8}\, U_{9,10}\, U_{10,11}\, L,$$

whose ASAP levels, in operator order, are $(4, 4, 4, 3, 2, 1, 3, 2, 1, 3, 2, 1)$: the four layers

$$\{L, U_{7,8}, U_{3,4}\},\ \{U_{10,11}, U_{6,7}, U_{2,3}\},\ \{U_{9,10}, U_{5,6}, U_{1,2}\},\ \{K, U_{4,5}, U_{8,9}\}$$

(in time order) implement it, so its depth is 4, and every layering of M uses at least 4 layers. Conjecture 1 predicts $\frac{1}{2}(11+3)-2 = 5$. The placement $\vec{n} = (7, 5)$ given by Eq. (61) of [1] *for $d = 5$ does have depth 5. Moreover $d_{\min}(11, 2) = 4$ (no placement of two $-\kappa$'s on 11 sites has depth ≤ 3), the whole row predicted by Conjecture 1 at $N = 11$, namely $(12, 6, 5, 4, 3, 2, 2, 3, 4, 5, 6, 12)$, differs from the true row, and for every odd $N \in \{11, 13, 15, 17, 19\}$ one has $d_{\min}(N, 2) \neq \frac{1}{2}(N + 3) - 2$ while $d_{\min}(N, 2) = \lceil (N + 1)/3 \rceil$.*

Theorem 3.6 (Conjecture 2 is false). *At $N = 8$, $\kappa_- = 2$ (so $0 < \kappa_- \leq N/2 = 4$), the placement $\vec{n} = (6, 3)$ gives the word*

$$M = U_{6,7}\, U_{3,4}\, K\, U_{1,2}\, U_{2,3}\, U_{4,5}\, U_{5,6}\, U_{7,8}\, L,$$

with ASAP levels $(3, 3, 3, 2, 1, 2, 1, 2, 1)$: the three layers $\{L, U_{5,6}, U_{2,3}\}$, $\{U_{7,8}, U_{4,5}, U_{1,2}\}$, $\{K, U_{3,4}, U_{6,7}\}$ implement it, its depth is 3, and every layering uses at least 3 layers. Conjecture 2 predicts $\frac{1}{2}(8+4)-2 = 4$. The placement $\vec{n} = (6, 4)$ given by Eq. (63) for $d = 4$ does have depth 4. Moreover $d_{\min}(8, 2) = 3$, the row predicted by Conjecture 2 at $N = 8$, namely $(9, 5, 4, 3, 2, 3, 4, 5, 9)$, differs from the true row, and for every even $N \in \{8, 10, 12, 14, 16, 18\}$ one has $d_{\min}(N, 2) \neq \frac{1}{2}(N+4)-2$ while $d_{\min}(N, 2) = \lceil (N+1)/3 \rceil$.

Proof of Theorems 3.5 and 3.6. The words, the levels and the depths of the two witnesses and of the two placements (61), (63) are computed inside the Lean kernel by `decide` (no compiled evaluation); "every layering uses at least 4 (resp. 3) layers" is Theorem 2.2 applied to the computed depth. The statements $d_{\min}(11, 2) = 4$ and $d_{\min}(8, 2) = 3$, the two wrong rows and the recurrence at larger N are read off Table 1 (Proposition 3.1, Theorems 3.2, 3.3); the raw form "no placement of two $-\kappa$'s on 11 (resp. 8) sites has depth ≤ 3 (resp. ≤ 2)" is a separate sweep of the 2^{11} (resp. 2^8) masks (`no_depth_below_four_at_11`, `no_depth_below_three_at_8`). The witnesses are shown in Figures 1 and 2 beside the source's placements. $\square$

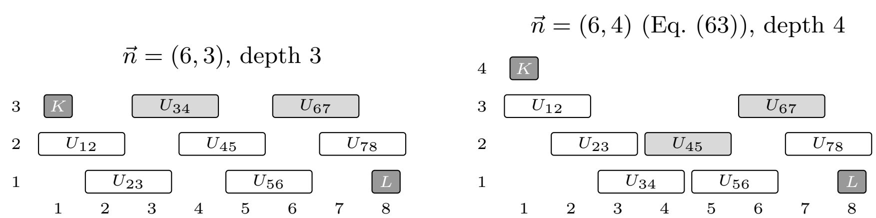

FIGURE 1. $N = 8$, $\kappa_- = 2$. Left: the counterexample to Conjecture 2, three staircases of three gates each. Right: the placement of Eq. (63) of [1], in the staircase+brickwork+staircase geometry of its Sec. 3.2.3. Time runs upwards, layer numbers at the left; shaded bulk gates are those of the left product (sites of $\vec{n}$), dark squares the boundary gates. Layers are the ASAP levels of Definition 2.1; by Theorem 2.2 no layering is shorter.

Remark 3.7 (why the conjectured placements are not optimal). Fix $\theta_N = +\kappa$ and let $\vec{n} = \{s_1 < \cdots < s_\kappa\} \subseteq \{1, \ldots, N-1\}$ cut $\{1, \ldots, N-1\}$ into the $\kappa + 1$ *gaps*

$$g_0 = s_1 - 1, \qquad g_i = s_{i+1} - s_i - 1\ (1 \leq i \leq \kappa - 1), \qquad g_\kappa = N - 1 - s_\kappa, \qquad \textstyle\sum_i g_i = N - 1 - \kappa,$$

each a run of consecutive $+\kappa$ sites, i.e. a staircase of U-gates in (2). Section 4 shows that the depth is at least $\max(g_0 + 1, \ldots, g_{\kappa-1} + 1, g_\kappa + 2)$, with equality when no two sites of $\vec{n}$ are adjacent. The

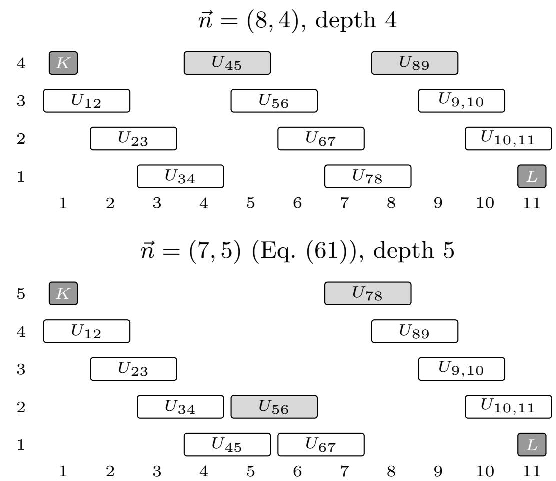

FIGURE 2. $N = 11$, $\kappa_- = 2$. Top: the counterexample to Conjecture 1, three staircases of four gates each. Bottom: the placement of Eq. (61) of [1], in the staircase+brickwork+staircase geometry. Conventions as in Figure 1.

placement (61) has $g_0 = d - 1$, $g_1 = \cdots = g_{\kappa-1} = 1$, $g_\kappa = d - 2$: the constraint $\sum g_i = N - 1 - \kappa$ then reads $2d + \kappa - 4 = N - 1 - \kappa$, which is exactly $d = \frac{1}{2}(N + 3) - \kappa$; likewise (63) has $g_\kappa = d - 3$ and gives $d = \frac{1}{2}(N + 4) - \kappa$. So the conjectured formulas are what one gets by insisting that every interior gap be a single site — "an initial block of $d - 1$ inhomogeneities equal to κ, followed by an alternating block of $(-\kappa, \kappa)$ values, and finally a second block containing only κ values", as Sec. 3.2.3 of [1] describes them — which builds two staircases with a brickwork between them. Spreading the $N - 1 - \kappa$ sites over all $\kappa + 1$ gaps gives $\lceil (N + 1)/(\kappa + 1) \rceil$ instead. The two agree when two staircases are as good as $\kappa + 1$ (at $\kappa = 1$ and at the staggered end $\kappa = \lfloor (N - 1)/2 \rfloor$) and for every N too small for a third staircase to help; the first disagreements are $(N - 1)/2 > \lceil (N + 1)/3 \rceil$ at $N = 11$ and $N/2 > \lceil (N + 1)/3 \rceil$ at $N = 8$, at $\kappa = 2$. Beyond the two smallest counterexamples, the circuits of the source's Fig. 8 ($N = 13$) with $\kappa_- = 2, 3, 4$, whose captions give depths $6, 5, 4$ for $\vec{n} = (8, 6), (9, 7, 5), (10, 8, 6, 4)$, are not of minimum depth either: by Theorem 3.2, $d_{\min}(13, \kappa_-) = 5, 4, 3$ there.

Remark 3.8 (the source's Appendix D). Appendix D of [1] ("Almost a good choice") presents, in its Fig. 23, an alternative family of circuits for $N = 9$, with sub-captions

$$\vec{n} = (8, 6, 4, 2),\ (8, 5, 2),\ (8, 4),\ (5),\ \emptyset, \qquad \text{depth } 2,\ 3,\ 4,\ 5,\ 10,$$

of which, as the appendix says, "the case for $\kappa_- = 0, 1$ and $\kappa_- = \frac{N-1}{2}$ coincide with the ones in section 3.2.2", so that the members at $\kappa_- = 2$ and $\kappa_- = 3$ are placements genuinely outside the family (61). All five have the minimum depth: they are the row $N = 9$ of Table 1, entry by entry. The appendix adds that "the distribution of the gates looks much more like the choice made in the periodic case" of Paletta, Duh, Pozsgay and Zadnik [6], in which "minimising the depth was not one of their goals". The appendix explains why the family was not adopted — "It happens that simultaneously keeping the very regular gate distribution shown in Figure 23 and minimizing the depth is not possible beyond $N = 11$" — and argues it by counting sites: "for $N = 13$, we need five gates (that take 10 sites) plus four sites in between … we need 14 sites for this configuration but we only have 13 available". That argument is correct, and it is about the *regular picture*, not about the minimum: $d_{\min}(13, 4) = 3$ (Theorem 3.2) is attained by the uneven gap profile $(1, 2, 2, 2, 1)$, i.e. by $\vec{n} = (11, 8, 5, 2)$. Section 4 is exactly the statement that giving up the regularity costs nothing, at any N; so Appendix D is where [1] comes closest to the answer, and the credit for looking outside the family of Conjectures 1 and 2 belongs there.

4. The closed form

Theorem 4.1 (not formalized; proved on paper below)**.** *For every $N \geq 2$ and every $0 \leq \kappa_- \leq N$,*

$$d_{\min}(N, \kappa_-) = \left\lceil \frac{N+1}{\min(\kappa_-, N-\kappa_-)+1} \right\rceil.$$

Throughout the proof a configuration θ is identified with its word $W_1(\theta)$ or $W_2(\theta)$ of (2)–(3), A, B are as there, $a = |A|$, and for $A = \{s_1 < \cdots < s_a\}$ the gaps $g_0, \ldots, g_a$ are those of Remark 3.7 (with the convention $g_0 = N-1$ when $a = 0$). Depths are lower-bounded by chains (Lemma 2.3).

Step (a): flipping every sign reverses the word. Let $\bar\theta$ be θ with every $\pm\kappa$ replaced by $\mp\kappa$; it has $\bar A = B$, $\bar B = A$ and $\bar\theta_N = -\theta_N$. If $\theta_N = +\kappa$ then

$$W_2(\bar\theta) = L\Big(\overleftarrow{\prod}_{j\in B} U_{j,j+1}\Big) K \Big(\overrightarrow{\prod}_{i\in A} U_{i,i+1}\Big)$$

is letter for letter the reversal of $W_1(\theta)$, and symmetrically if $\theta_N = -\kappa$. By Lemma 2.3, $\operatorname{depth} W(\bar\theta) = \operatorname{depth} W(\theta)$; since $|\bar{\vec n}| = N - |\vec n|$, this gives $d_{\min}(N, \kappa_-) = d_{\min}(N, N-\kappa_-)$, the duality asserted after Fig. 8 of [1] ("the circuit for $\frac{N-1}{2} < \kappa_- \leq N$ is obtained by reversing the order of the time layers"). Per configuration and for $N \leq 12$ this is also kernel-checked (`kkDepth_sign_flip`, Proposition 6.1). $\square$

Step (b): gap chains in W_1. Let $\theta_N = +\kappa$. In time, W_1 runs L, then the B-gates in decreasing site order, then K, then the A-gates in increasing site order. Writing U_j for $U_{j,j+1}$, each gap yields a chain:

- the bottom run $\{1, \ldots, s_1 - 1\} \subseteq B$ gives $U_{s_1-1}, \ldots, U_1, K$, of length $g_0 + 1$ (consecutive bulk gates share a site; U_1 and K share site 1);
- the interior run $\{s_i + 1, \ldots, s_{i+1} - 1\} \subseteq B$, $1 \leq i \leq a-1$, gives $U_{s_{i+1}-1}, \ldots, U_{s_i+1}, U_{s_i}$, of length $g_i + 1$ (the last gate belongs to A and acts after every B-gate);
- the top run $\{s_a + 1, \ldots, N-1\} \subseteq B$ gives $L, U_{N-1}, \ldots, U_{s_a+1}, U_{s_a}$, of length $g_a + 2$ (also when $g_a = 0$, where it is L, U_{N-1});
- if $a = 0$, the single chain $L, U_{N-1}, \ldots, U_1, K$ has length $N+1$.

Hence with $d = \operatorname{depth} W_1(\theta)$: $g_i \leq d-1$ for $i < a$ and $g_a \leq d-2$, so $N-1-a = \sum g_i \leq a(d-1)+(d-2)$, i.e. $(a+1)d \geq N+1$:

$$\operatorname{depth} W_1(\theta) \ \geq\ \left\lceil \frac{N+1}{a+1} \right\rceil, \qquad a = |A|, \tag{5}$$

which for $a = 0$ is the chain of length $N+1$. $\square$

Step (c): gap chains in W_2. Let $\theta_N = -\kappa$. The same chains exist in W_2 except that L now acts last, so the top run contributes only $U_{N-1}, \ldots, U_{s_a+1}, U_{s_a}$, of length $g_a + 1$ (and for $a = 0$ the chain $U_{N-1}, \ldots, U_1, K$ of length N). Thus $g_i \leq d-1$ for every $i \leq a$, $N-1-a \leq (a+1)(d-1)$, and

$$\operatorname{depth} W_2(\theta) \ \geq\ \left\lceil \frac{N}{a+1} \right\rceil, \qquad a = |A|. \tag{6}$$

$\square$

Step (d): the lower bound. Let $|\vec n| = \kappa$ and $d = \operatorname{depth} M(\vec n)$. We use that for $1 \leq m \leq N$, $\frac{N}{m} \geq \frac{N+1}{m+1}$ (equivalent to $N \geq m$), hence $\lceil N/m \rceil \geq \lceil (N+1)/(m+1) \rceil$.

If $\theta_N = +\kappa$, then $a = \kappa \leq N-1$ and (5) gives $d \geq \lceil (N+1)/(\kappa+1) \rceil$; by Step (a), $d = \operatorname{depth} W_2(\bar\theta)$, whose A-set is B, of size $N-1-\kappa$, so (6) gives $d \geq \lceil N/(N-\kappa) \rceil \geq \lceil (N+1)/(N-\kappa+1) \rceil$.

If $\theta_N = -\kappa$, then $\kappa \geq 1$, $a = \kappa - 1$ and (6) gives $d \geq \lceil N/\kappa \rceil \geq \lceil (N+1)/(\kappa+1) \rceil$; by Step (a), $d = \operatorname{depth} W_1(\bar\theta)$, whose A-set is B, of size $N-\kappa$, so (5) gives $d \geq \lceil (N+1)/(N-\kappa+1) \rceil$ (this includes $\kappa = N$, where $B = \emptyset$ and the chain has length $N+1$).

In both cases $d \geq \max\big(\lceil \frac{N+1}{\kappa+1} \rceil, \lceil \frac{N+1}{N-\kappa+1} \rceil\big) = \lceil \frac{N+1}{\min(\kappa, N-\kappa)+1} \rceil$. $\square$

Step (e): the construction. By Step (a) it suffices to attain the bound for $\kappa \le N/2$, and we take $\theta_N = +\kappa$ (possible since $\kappa \le N/2 \le N-1$). Put $d = \lceil (N+1)/(\kappa+1) \rceil$; as $\kappa < N$, $d \ge 2$. If $\kappa = 0$ the only placement is the staircase, of depth $N+1 = d$. Let $\kappa \ge 1$. Choose gaps

$$g_\kappa = d-2, \qquad g_1,\dots,g_{\kappa-1} \ge 1, \qquad g_0 \ge 0, \qquad g_i \le d-1 \ (0 \le i < \kappa), \qquad \textstyle\sum_{i=0}^{\kappa} g_i = N-1-\kappa.$$

This is feasible. After the top gap and the minimum 1 of each interior gap, $R = (N-1-\kappa) - (d-2) - (\kappa-1) = N+2-2\kappa-d$ units remain to be distributed over $g_0,\dots,g_{\kappa-1}$; the caps give room for them iff $N+1-\kappa-d \le \kappa(d-1)$, i.e. $(\kappa+1)d \ge N+1$, which holds by the choice of d; and $R \ge 0$ iff $2\kappa + d \le N+2$. For the latter, $d \le \frac{N+1+\kappa}{\kappa+1}$, and $\frac{N+1+\kappa}{\kappa+1} \le N+2-2\kappa$ reduces to $N\kappa \ge 2\kappa^2+\kappa-1$, i.e. $N \ge 2\kappa+1-\frac{1}{\kappa}$: true for $\kappa = 1$ (as $N \ge 2$) and for every $\kappa \ge 2$ with $N \ge 2\kappa+1$; in the remaining case $N = 2\kappa$, $\kappa \ge 2$, one has $d = \lceil (2\kappa+1)/(\kappa+1) \rceil = 2$ and $2\kappa + d = N+2$, so $R = 0$. Let $\vec{n} = \{s_1 < \cdots < s_\kappa\}$ be the placement with these gaps; no two of its sites are adjacent.

The following assignment is a layering of $W_1(\theta)$ (the conflicts of the word are only $U_{j,j+1}$ with $U_{j\pm1,j\pm2}$, K with $U_{1,2}$, and L with $U_{N-1,N}$): $L \mapsto 1$; in the top run, $U_{j,j+1} \mapsto N-j+1$ (from 2 at $j = N-1$ up to $g_\kappa + 1$); in the interior run between s_i and s_{i+1}, $U_{j,j+1} \mapsto s_{i+1} - j$ (from 1 up to g_i); in the bottom run, $U_{j,j+1} \mapsto s_1 - j$ (from 1 up to g_0); $K \mapsto g_0 + 1$; $U_{s_i,s_i+1} \mapsto g_i + 1$ for $i < \kappa$ and $U_{s_\kappa,s_\kappa+1} \mapsto g_\kappa + 2$. Indeed, within a run consecutive gates are earlier in time as j increases and get smaller layers; gates of different runs of B do not conflict; L precedes $U_{N-1,N}$ with layers $1 < 2$; $U_{1,2}$ precedes K with layers $g_0 < g_0+1$ when $g_0 \ge 1$, while for $g_0 = 0$ the gate $U_{1,2} = U_{s_1,s_1+1}$ is in A, acts after K, and gets layer $g_1 + 1 \ge 2 > 1$ (or $g_\kappa + 2 \ge 2$ if $\kappa = 1$); each U_{s_i,s_i+1} acts after all B-gates and conflicts only with U_{s_i-1,s_i} (layer 1) and U_{s_i+1,s_i+2} (layer g_i, or $g_\kappa + 1$ for $i = \kappa$), so its layer exceeds both; and A-gates do not conflict with each other since no two sites of $\vec{n}$ are adjacent. The largest layer used is $\max(g_0+1,\dots,g_{\kappa-1}+1, g_\kappa+2) \le \max(d, d) = d$. By Theorem 2.2 depth $W_1(\theta) \le d$, and by Step (d) depth $W_1(\theta) \ge d$. This proves Theorem 4.1. ☐

The construction of Step (e), with the remaining R units poured into $g_{\kappa-1}, g_{\kappa-2}, \dots, g_0$ in that order (each capped at $d-1$) and the case $\kappa > N/2$ obtained by sign flip, is the function `optConfig` of Appendix A. Its correctness is checked directly, without the proof above, on a range far beyond the exhaustive one:

Theorem 4.2 (the explicit placement, $N \le 120$)**.** *For every $2 \le N \le 120$ and every $0 \le \kappa_- \le N$, the placement* `optConfig`(N,κ_-) *— for $\kappa_- \le N/2$ the $\kappa_- + 1$-staircase placement with gap profile $(g_0,\dots,g_{\kappa_- -1}, d-2)$ of Step (e), $d = \lceil (N+1)/(\kappa_-+1) \rceil$; for $\kappa_- > N/2$ the complement of the placement for $N - \kappa_-$ — has exactly κ_- sites and its circuit has depth exactly $\lceil (N+1)/(\min(\kappa_-, N-\kappa_-)+1) \rceil$.*

Proof. A single `native_decide` over the $\sum_{N=2}^{120}(N+1) = 7\,378$ pairs (N,κ_-), each building the word and scheduling it (`optConfig_attains`). Together with Theorems 3.2–3.3 this pins $d_{\min}$ on $2 \le N \le 19$ without Theorem 4.1, and certifies the upper bound alone on $2 \le N \le 120$, past the $N = 51$ (odd) and $N = 50$ (even) to which the source reports checking (61) and (63). ☐

Corollary 4.3. *For $1 \le \kappa_- \le N/2$ the difference between the conjectured value and $d_{\min}(N,\kappa_-)$ is $\frac{1}{2}(N+3) - \kappa_- - \lceil (N+1)/(\kappa_-+1) \rceil$ (odd N) or $\frac{1}{2}(N+4) - \kappa_- - \lceil (N+1)/(\kappa_-+1) \rceil$ (even N); it vanishes at $\kappa_- = 1$, at $\kappa_- = \lfloor (N-1)/2 \rfloor$ and, for even N, at $\kappa_- = N/2$, and at every fixed $\kappa_- \ge 2$ it grows without bound with N.*

5. The $(+\kappa,\rho)$ circuits and Eq. (73)

Section 3.3.1 of [1] considers configurations with $\theta_N = +\kappa$, no $-\kappa$, and ρ at the sites $\vec{n}^\rho \subseteq \{1,\dots,N-1\}$, $\rho_+ = |\vec{n}^\rho|$, whose word is (4). It states that "contrary to the case with $(\kappa,-\kappa)$, whose maximum depth is $d = N+1$, here $d = N+1$ is actually the minimum possible depth within this construction", gives in its Eq. (72) placements of depth $N+1$ for odd N and $1 \le \rho_+ \le (N-1)/2$, and adds the remark quoted in the introduction: exhaustive for $N = 5, 6, 7, 9$, sampled for $N = 11, 13, 15$.

Theorem 5.1 ($(+\kappa,\rho)$: the minimum is $N+1$)**.** *For every $3 \le N \le 20$, the minimum of* depth $W_\rho(\vec{n}^\rho)$ *over all 2^{N-1} subsets $\vec{n}^\rho \subseteq \{1,\dots,N-1\}$ equals $N+1$. In particular, for $N = 11, 13, 15$ every placement of the ρ's gives depth at least $N+1$. At $N = 7$ the minimum depth by $\rho_+ = 0,\dots,6$ is*

$(8,8,8,8,8,10,14)$*: the three circuits of the source's Fig. 13,* $\vec{n}^\rho = (6,4,2), (5,3), (4)$ *from Eq. (72), all have depth* $8 = N+1$*, and the circuit* $\vec{n}^\rho = (6,5,4,3,2,1)$ *of its Fig. 14 has depth* $14 = 2N$*.*

Proof. The sweeps are exhaustive and discharged by `native_decide`. For $3 \le N \le 15$ the statement is `rhoMin_eq_succ_le_15`; for $16 \le N \le 20$ ($2^{19} = 524\,288$ placements at $N = 20$) it is `rhoMin_eq_succ_16_20`. The inequality in the source's own form, over the $2^{10}+2^{12}+2^{14} = 21\,504$ placements of the three sampled sizes, is `rho_depth_ge_succ_11_13_15`, and the $N = 7$ row is `rhoTable_seven`. The minimum is attained (at $\vec{n}^\rho = \emptyset$; `rho_bound_sharp`), so $N+1$ is sharp. The C program of Remark 3.4 agrees to $N = 24$ (20 s). □

Remark 5.2 (a general argument, outside the formal development). The lower bound holds for every $N \ge 2$, by one chain: in (4) the right product has a gate at every site, so $L, Y_{N-1,N}, \ldots, Y_{1,2}, K$ is a chain of length $N+1$, and $\vec{n}^\rho = \emptyset$ attains it. So the exact value $N+1$ for all N follows from Lemma 2.3; the theorem above is the exhaustive, kernel-checked form of the statement the source made, on the range it named. This remark is not formalized.

Theorem 5.3 (the product order in Eq. (73))**.** *Eq. (73) of* [1] *prints the left product of the* $(+\kappa,\rho)$ *circuit as* $\overrightarrow{\prod}_{1\le r\le \rho_+} V_{n_r^\rho, n_r^\rho+1}$ *(increasing in the site index), whereas Lemma 1 (Eq. (35)) and Theorem 1 (Eq. (42)) of* [1] *give a decreasing product* $\overleftarrow{\prod}$ *there. The two readings are not interchangeable: for* $3 \le N \le 12$ *and* $\vec{n}^\rho = (N-1,\ldots,1)$*, the word with the increasing product,* $V_{1,2}\cdots V_{N-1,N}\,K\,Y_{1,2}\cdots Y_{N-1,N}\,L$*, has depth* $N+2$*, while the word* (4) *with the decreasing product has depth* $2N$*. The source itself states* $d = 2N$ *for this circuit ("for* $\vec{n}^\rho = (N-1, N-2, \ldots, 3, 2, 1)$ *the circuit has depth* $d = 2N$*", its Fig. 14 at* $N = 7$*), which only the decreasing reading produces.*

Proof. `rho_fwd_order_wrong` and `rho_all_depth_two_mul`, by `native_decide`. In words: with the decreasing product the circuit is a single chain $L, Y_{N-1,N}, \ldots, Y_{1,2}, K, V_{1,2}, \ldots, V_{N-1,N}$ of $2N$ gates, whereas with the increasing product $V_{N-1,N}$ acts right after K and can be placed in layer 4, and $V_{j,j+1}$ in layer $N+3-j$, giving $N+2$. The misprint is invisible in the source's own minimum-depth circuits, because Eq. (72) places the ρ's at pairwise distance ≥ 2, where the V-gates commute and the order is immaterial. This paper follows Lemma 1 throughout. □

Two further readings of [1] point the same way. Its Eq. (78), which applies the same Lemma 1 to the three-value $(\kappa, -\kappa, \rho)$ circuits of its Sec. 3.3.1, writes the very same V-product as $\overleftarrow{\prod}_{1\le s\le \rho_+} V_{n_s^\rho, n_s^\rho+1}$, decreasing; and the deposited notebook [2] builds the left product of Lemma 1 in decreasing site order (its `Ma` multiplies $\check{R}_{nn-i,\,nn-i+1}$ over `{i,1,nn-1}`, i.e. from site $N-1$ down to site 1), so the source's own code agrees with the decreasing reading. Only the printed Eq. (73) does not.

6. Controls

The following checks guard the model and the sweeps; none is new mathematics.

Proposition 6.1 (controls)**.** (1) *For* $2 \le N \le 12$ *and every mask* $m < 2^N$*, the word* $M(\vec{n})$ *has exactly* $N+1$ *gates.*

(2) *For* $2 \le N \le 14$*, the maximum of* $\operatorname{depth} M(\vec{n})$ *over all* 2^N *placements is* $N+1$ *(the source's "maximum depth is* $d = N+1$*"), attained by the staircase* $\vec{n} = \emptyset$ *for* $2 \le N \le 12$*.*

(3) *For* $2 \le N \le 12$ *and every mask* m*,* $\operatorname{depth} M(\vec{n}) = \operatorname{depth} M(\bar{\vec{n}})$ *where* $\bar{\vec{n}}$ *is the complement (Step (a) of Section 4, per configuration); and every row of Table 1,* $2 \le N \le 19$*, is a palindrome.*

(4) *Non-vacuity: for* $2 \le N \le 12$ *every entry of the computed row is* $\le N+1$*, so no* κ_- *is left without a placement (the fold starts from the sentinel* $2N+4$*).*

(5) *Too small refuted: no placement of two* $-\kappa$*'s on* 11 *sites has depth* ≤ 3 *and none on* 8 *sites has depth* ≤ 2*; the witnesses of depth* 4 *and* 3 *are exhibited with their schedules (Theorems 3.5, 3.6); the twelve gates of the* $N = 11$ *witness cannot all be run in one layer.*

(6) *Too large refuted for the* $(+\kappa,\rho)$ *family: for* $N = 5, 7, 9, 11, 13, 15$ *the placement* $\vec{n}^\rho = \emptyset$ *has depth exactly* $N+1$*, so "*$d \ge N+2$*" is false.*

(7) *The* $N = 7$ *circuit* $\vec{n}^\rho = (6, \ldots, 1)$ *has depth* $2N$ *(the source's Fig. 14 value), for* $3 \le N \le 12$*.*

7. Verification

Theorem 2.2, Proposition 3.1, Theorems 3.2, 3.3, 3.5, 3.6, 4.2, 5.1, 5.3 and Proposition 6.1 have been verified in Lean 4 (toolchain v4.32.0 [8]); their statements are reproduced verbatim in Appendix A. The development does not import Mathlib: its ten modules import only the core library and a one-attribute tagging module of the surrounding repository. They are `Depth` (supports, conflicts, ASAP levels, `Layering`, Theorem 2.2), `Circuit` (the words (2)–(4) from a mask, the tables, the closed form and the two conjectured formulas), `Values` ($2 \leq N \leq 10$), `ValuesExt` ($11 \leq N \leq 16$), `ValuesExt2` ($17 \leq N \leq 19$), `Optimal` (`optConfig` and Theorem 4.2), `Rho` and `RhoExt` (Section 5), `Controls` and `Refute` (the two counterexamples). Counting the sources, they hold 70 theorem declarations (9, 14, 8, 5, 1, 5, 1, 6, 21 in the modules `Depth`, `Values`, `ValuesExt`, `ValuesExt2`, `Optimal`, `Rho`, `RhoExt`, `Controls`, `Refute`) and 32 definitions, and 63 of the theorems carry the statements of this paper; those 63 and all 32 definitions are the ones reproduced in Appendix A.

Theorem 2.2 (`depth_le_of_layering`, `depth_attained`) is proved by structural induction and depends only on `propext`, `Quot.sound` and (for the second) `Classical.choice`; those same three carry the minimum-layer statements `witness11_min_layers` and `witness8_min_layers`, while the one-layer control `witness11_not_one_layer` needs only `propext` and `Quot.sound`. The words, levels, depths and κ_- of the two witnesses and of the source's placements (61), (63), and the three Fig. 6 depths, are evaluated by `decide` inside the kernel (axioms: `propext` only, or none). Every exhaustive sweep — the nineteen rows `kkTable_value_N`, the ρ minima, Theorem 4.2, the controls (1)–(4), (6), (7), the two "no depth below" sweeps and the two conjecture-agreement statements — is discharged by `native_decide`, so each such theorem carries its own axiom `<decl>._native.native_decide.ax_1_1`, asserting that the compiled evaluation returned `true`, together with `propext` (which alone among them `kkWord_length` does not need); the derived statements (equality with the closed form, the palindromes, the failing rows, Theorems 3.5, 3.6 at the row level) rewrite with those tables and inherit exactly their native axioms and no other. There is no `sorry`, and no axiom is declared by hand. One full pass over the ten modules takes 388 processor-seconds (`Depth` 3, `Circuit` 3, `Values` 3, `ValuesExt` 15, `ValuesExt2` 115, `Optimal` 14, `Rho` 10, `RhoExt` 216, `Controls` 7, `Refute` 2; elapsed seconds on one core, remeasured module by module), with peak resident memory about 1.6 GiB per module, most of it the loaded `.olean` images; the C enumerator of Remark 3.4 took 4.2 s for the $(\kappa, -\kappa)$ table to $N = 24$, 20 s for the $(+\kappa, \rho)$ table to $N = 24$ and 6 s for the maximum-depth and sign-flip checks; about 0.3 processor-hours in all. Theorem 4.1 and Lemma 2.3 are proved on paper only; the exhaustive range $N \leq 19$ and the construction range $N \leq 120$ are their machine-checked instances. The source of the development and of the C program is currently held in a private repository; repository reference to be added at submission.

Appendix A. The formal statements

The Lean statements corresponding to the results above, verbatim (statements only; proofs, docstrings and attributes omitted), preceded by the definitions they use, grouped by module. A gate is a pair `(lo, hi)` of sites; a word is a `List Sup` in operator order; `levels` are the ASAP levels, `depth` the depth, and `Layering w f` says f is a layering of w (the argument `rest.getD i (0, 0)` is the i-th gate of `rest`, `fs.getD i 0` its layer). Masks, `kkWord`, `kkTable`, `rhoWord`, `rhoMin`, `rhoTable`, `closedForm`, `conj1`, `conj2` are as in Section 2.3; `witness11 = 136` and `witness8 = 36` are the masks of $\vec{n} = (8, 4)$ and $(6, 3)$, and `80`, `40` those of $(7, 5)$ and $(6, 4)$; `List.range' 2 119` is $[2, \ldots, 120]$.

`Depth.lean`

```
abbrev Sup := Nat × Nat
def ov (a b : Sup) : Bool := decide (a.1 ≤ b.2) && decide (b.1 ≤ a.2)
def maxOv (g : Sup) : List Sup → List Nat → Nat
  | a :: as, l :: ls => if ov g a then Nat.max l (maxOv g as ls) else maxOv g as ls
  | _, _ => 0
def levels : List Sup → List Nat
  | [] => []
  | g :: rest =>
      let ls := levels rest
      (maxOv g rest ls + 1) :: ls
def maxL (f : List Nat) : Nat := f.foldr Nat.max 0
def depth (w : List Sup) : Nat := maxL (levels w)
```

```
inductive Layering : List Sup → List Nat → Prop
  | nil : Layering [] []
  | cons {g : Sup} {rest : List Sup} {a : Nat} {fs : List Nat} :
      Layering rest fs → 1 ≤ a →
      (∀ i, i < rest.length → ov g (rest.getD i (0, 0)) = true → fs.getD i 0 < a) →
      Layering (g :: rest) (a :: fs)
theorem depth_le_of_layering {w : List Sup} {f : List Nat} (h : Layering w f) :
    depth w ≤ maxL f
theorem depth_attained (w : List Sup) : Layering w (levels w) ∧ maxL (levels w) = depth w
```

`Circuit.lean`

```
def leftPart (N : Nat) (inA : Nat → Bool) : List Sup :=
  (List.range (N - 1)).foldl (fun acc t => if inA (t + 1) then (t + 1, t + 2) :: acc else acc) []
def rightPart (N : Nat) (inB : Nat → Bool) : List Sup :=
  (List.range (N - 1)).filterMap (fun t => if inB (t + 1) then some (t + 1, t + 2) else none)
def wordOf (N : Nat) (inA inB : Nat → Bool) (lem2 : Bool) : List Sup :=
  if lem2 then (N, N) :: (leftPart N inA ++ (1, 1) :: rightPart N inB)
  else leftPart N inA ++ (1, 1) :: (rightPart N inB ++ [(N, N)])
def bit (m i : Nat) : Bool := (m >>> (i - 1)) % 2 == 1
def pc (n m : Nat) : Nat :=
  (List.range n).foldl (fun a t => if (m >>> t) % 2 == 1 then a + 1 else a) 0
def kkWord (N m : Nat) : List Sup :=
  wordOf N (fun i => bit m i) (fun j => !bit m j) (bit m N)
def kkDepth (N m : Nat) : Nat := depth (kkWord N m)
def rhoWord (N m : Nat) : List Sup :=
  wordOf N (fun i => bit m i) (fun _ => true) false
def rhoDepth (N m : Nat) : Nat := depth (rhoWord N m)
def bump (best : List Nat) (k v : Nat) : List Nat :=
  best.set k (Nat.min (best.getD k v) v)
def kkTable (N : Nat) : List Nat :=
  (List.range (1 <<< N)).foldl (fun best m => bump best (pc N m) (kkDepth N m))
    (List.replicate (N + 1) (2 * N + 4))
def rhoMin (N : Nat) : Nat :=
  (List.range (1 <<< (N - 1))).foldl (fun best m => Nat.min best (rhoDepth N m)) (2 * N + 4)
def rhoTable (N : Nat) : List Nat :=
  (List.range (1 <<< (N - 1))).foldl (fun best m => bump best (pc (N - 1) m) (rhoDepth N m))
    (List.replicate N (2 * N + 4))
def closedForm (N k : Nat) : Nat := (N + Nat.min k (N - k) + 1) / (Nat.min k (N - k) + 1)
def conj1 (N k : Nat) : Nat := (N + 3) / 2 - k
def conj2 (N k : Nat) : Nat := (N + 4) / 2 - k
```

`Values.lean` (Proposition 3.1; palindromes and staircase in Proposition 6.1)

```
theorem kkTable_value_2 : kkTable 2 = [3, 2, 3]
theorem kkTable_value_3 : kkTable 3 = [4, 2, 2, 4]
theorem kkTable_value_4 : kkTable 4 = [5, 3, 2, 3, 5]
theorem kkTable_value_5 : kkTable 5 = [6, 3, 2, 2, 3, 6]
theorem kkTable_value_6 : kkTable 6 = [7, 4, 3, 2, 3, 4, 7]
theorem kkTable_value_7 : kkTable 7 = [8, 4, 3, 2, 2, 3, 4, 8]
theorem kkTable_value_8 : kkTable 8 = [9, 5, 3, 3, 2, 3, 3, 5, 9]
theorem kkTable_value_9 : kkTable 9 = [10, 5, 4, 3, 2, 2, 3, 4, 5, 10]
theorem kkTable_value_10 : kkTable 10 = [11, 6, 4, 3, 3, 2, 3, 3, 4, 6, 11]
theorem kkTable_eq_closedForm_le_10 :
    ∀ N ∈ [2, 3, 4, 5, 6, 7, 8, 9, 10], kkTable N = (List.range (N + 1)).map (closedForm N)
theorem kkTable_palindromic_le_10 :
    ∀ N ∈ [2, 3, 4, 5, 6, 7, 8, 9, 10], (kkTable N).reverse = kkTable N
theorem conj1_holds_le_9 :
    ∀ N ∈ [3, 5, 7, 9], ∀ k ∈ List.range (N + 1),
      0 < k → 2 * k ≤ N - 1 → (kkTable N).getD k 0 = conj1 N k
theorem conj2_holds_le_6 :
    ∀ N ∈ [2, 4, 6], ∀ k ∈ List.range (N + 1),
      0 < k → 2 * k ≤ N → (kkTable N).getD k 0 = conj2 N k
theorem staircase_depth :
    ∀ N ∈ [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], kkDepth N 0 = N + 1
```

`ValuesExt.lean` (Theorem 3.2)

```
theorem kkTable_value_11 : kkTable 11 = [12, 6, 4, 3, 3, 2, 2, 3, 3, 4, 6, 12]
theorem kkTable_value_12 : kkTable 12 = [13, 7, 5, 4, 3, 3, 2, 3, 3, 4, 5, 7, 13]
theorem kkTable_value_13 : kkTable 13 = [14, 7, 5, 4, 3, 3, 2, 2, 3, 3, 4, 5, 7, 14]
theorem kkTable_value_14 : kkTable 14 = [15, 8, 5, 4, 3, 3, 3, 2, 3, 3, 3, 4, 5, 8, 15]
theorem kkTable_value_15 : kkTable 15 = [16, 8, 6, 4, 4, 3, 3, 2, 2, 3, 3, 4, 4, 6, 8, 16]
theorem kkTable_value_16 : kkTable 16 = [17, 9, 6, 5, 4, 3, 3, 3, 2, 3, 3, 3, 4, 5, 6, 9, 17]
theorem kkTable_eq_closedForm_11_16 :
    ∀ N ∈ [11, 12, 13, 14, 15, 16], kkTable N = (List.range (N + 1)).map (closedForm N)
theorem kkTable_palindromic_11_16 :
    ∀ N ∈ [11, 12, 13, 14, 15, 16], (kkTable N).reverse = kkTable N
```

`ValuesExt2.lean` (Theorem 3.3)

```
theorem kkTable_value_17 :
    kkTable 17 = [18, 9, 6, 5, 4, 3, 3, 3, 2, 2, 3, 3, 3, 4, 5, 6, 9, 18]
theorem kkTable_value_18 :
    kkTable 18 = [19, 10, 7, 5, 4, 4, 3, 3, 3, 2, 3, 3, 3, 4, 4, 5, 7, 10, 19]
theorem kkTable_value_19 :
    kkTable 19 = [20, 10, 7, 5, 4, 4, 3, 3, 3, 2, 2, 3, 3, 3, 4, 4, 5, 7, 10, 20]
theorem kkTable_eq_closedForm_17_19 :
    ∀ N ∈ [17, 18, 19], kkTable N = (List.range (N + 1)).map (closedForm N)
theorem kkTable_palindromic_17_19 :
    ∀ N ∈ [17, 18, 19], (kkTable N).reverse = kkTable N
```

`Optimal.lean` (Theorem 4.2)

```
def fillGaps : Nat → Nat → Nat → List Nat
  | 0, _, _ => []
  | (n + 1), rem, cap =>
      let base := if n = 0 then 0 else 1
      let add := Nat.min (cap - base) rem
      fillGaps n (rem - add) cap ++ [base + add]
def gapsLow (N k : Nat) : List Nat :=
  let d := closedForm N k
  fillGaps k ((N + 1 - k - d) - (k - 1)) (d - 1)
def optPositions (N k : Nat) : List Nat :=
  ((gapsLow N k).foldl (fun st gi => (st.1 + gi + 1, st.2 ++ [st.1 + gi + 1])) (0, [])).2
def optMaskLow (N k : Nat) : Nat :=
  (optPositions N k).foldl (fun m s => m ||| (1 <<< (s - 1))) 0
def optConfig (N k : Nat) : Nat :=
  if 2 * k ≤ N then optMaskLow N k else ((1 <<< N) - 1) - optMaskLow N (N - k)
theorem optConfig_attains :
    ∀ N ∈ List.range' 2 119, ∀ k ∈ List.range (N + 1),
      pc N (optConfig N k) = k ∧ kkDepth N (optConfig N k) = closedForm N k
```

`Rho.lean` (Theorems 5.1, 5.3)

```
theorem rhoMin_eq_succ_le_15 :
    ∀ N ∈ [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15], rhoMin N = N + 1
theorem rho_depth_ge_succ_11_13_15 :
    ∀ N ∈ [11, 13, 15], ∀ m ∈ List.range (1 <<< (N - 1)), N + 1 ≤ rhoDepth N m
theorem rhoTable_seven : rhoTable 7 = [8, 8, 8, 8, 8, 10, 14]
theorem rho_all_depth_two_mul :
    ∀ N ∈ [3, 4, 5, 6, 7, 8, 9, 10, 11, 12], rhoDepth N ((1 <<< (N - 1)) - 1) = 2 * N
def rhoWordFwd (N m : Nat) : List Sup :=
  rightPart N (fun i => bit m i) ++ (1, 1) :: (rightPart N (fun _ => true) ++ [(N, N)])
theorem rho_fwd_order_wrong :
    ∀ N ∈ [3, 4, 5, 6, 7, 8, 9, 10, 11, 12],
      depth (rhoWordFwd N ((1 <<< (N - 1)) - 1)) = N + 2 ∧
        rhoDepth N ((1 <<< (N - 1)) - 1) = 2 * N
```

`RhoExt.lean` (Theorem 5.1)

```
theorem rhoMin_eq_succ_16_20 :
    ∀ N ∈ [16, 17, 18, 19, 20], rhoMin N = N + 1
```

`Controls.lean` (Proposition 6.1; `fig6_depths` in Proposition 3.1)

```
def kkMax (N : Nat) : Nat :=
  (List.range (1 <<< N)).foldl (fun best m => Nat.max best (kkDepth N m)) 0
theorem kkWord_length :
    ∀ N ∈ [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12],
      ∀ m ∈ List.range (1 <<< N), (kkWord N m).length = N + 1
theorem kkMax_eq :
    ∀ N ∈ [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14], kkMax N = N + 1
theorem kkDepth_sign_flip :
    ∀ N ∈ [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12],
      ∀ m ∈ List.range (1 <<< N), kkDepth N m = kkDepth N ((1 <<< N) - 1 - m)
theorem rho_bound_sharp :
    ∀ N ∈ [5, 7, 9, 11, 13, 15], rhoDepth N 0 = N + 1
theorem kkTable_not_sentinel :
    ∀ N ∈ [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12],
      ∀ v ∈ kkTable N, v ≤ N + 1
theorem fig6_depths : kkDepth 5 10 = 2 ∧ kkDepth 5 18 = 3 ∧ kkDepth 5 3 = 4
```

`Refute.lean` (Theorems 3.5, 3.6; controls (5))

```
def witness11 : Nat := 136
theorem witness11_kappa : pc 11 witness11 = 2
theorem witness11_word :
    kkWord 11 witness11 =
      [(8, 9), (4, 5), (1, 1), (1, 2), (2, 3), (3, 4), (5, 6), (6, 7), (7, 8), (9, 10),
       (10, 11), (11, 11)]
```

```
theorem witness11_schedule :
    levels (kkWord 11 witness11) = [4, 4, 4, 3, 2, 1, 3, 2, 1, 3, 2, 1]
theorem witness11_depth : depth (kkWord 11 witness11) = 4
theorem witness11_min_layers :
    (∀ f, Layering (kkWord 11 witness11) f → 4 ≤ maxL f) ∧
      Layering (kkWord 11 witness11) (levels (kkWord 11 witness11)) ∧
      maxL (levels (kkWord 11 witness11)) = 4
theorem witness11_not_one_layer :
    ¬ Layering (kkWord 11 witness11) (List.replicate 12 1)
theorem source_config_11 : kkDepth 11 80 = 5 ∧ pc 11 80 = 2
theorem conj1_false_at_11 : (kkTable 11).getD 2 0 = 4 ∧ conj1 11 2 = 5
theorem no_depth_below_four_at_11 :
    ∀ m ∈ List.range (1 <<< 11), pc 11 m = 2 → 4 ≤ kkDepth 11 m
theorem conj1_row_11_wrong :
    kkTable 11 ≠ [12, 6, 5, 4, 3, 2, 2, 3, 4, 5, 6, 12]
def witness8 : Nat := 36
theorem witness8_kappa : pc 8 witness8 = 2
theorem witness8_word :
    kkWord 8 witness8 =
      [(6, 7), (3, 4), (1, 1), (1, 2), (2, 3), (4, 5), (5, 6), (7, 8), (8, 8)]
theorem witness8_schedule :
    levels (kkWord 8 witness8) = [3, 3, 3, 2, 1, 2, 1, 2, 1]
theorem witness8_depth : depth (kkWord 8 witness8) = 3
theorem witness8_min_layers :
    (∀ f, Layering (kkWord 8 witness8) f → 3 ≤ maxL f) ∧
      Layering (kkWord 8 witness8) (levels (kkWord 8 witness8)) ∧
      maxL (levels (kkWord 8 witness8)) = 3
theorem source_config_8 : kkDepth 8 40 = 4 ∧ pc 8 40 = 2
theorem conj2_false_at_8 : (kkTable 8).getD 2 0 = 3 ∧ conj2 8 2 = 4
theorem no_depth_below_three_at_8 :
    ∀ m ∈ List.range (1 <<< 8), pc 8 m = 2 → 3 ≤ kkDepth 8 m
theorem conj2_row_8_wrong : kkTable 8 ≠ [9, 5, 4, 3, 2, 3, 4, 5, 9]
theorem conj1_false_odd_11_19 :
    ∀ N ∈ [11, 13, 15, 17, 19],
      (kkTable N).getD 2 0 ≠ conj1 N 2 ∧ (kkTable N).getD 2 0 = closedForm N 2
theorem conj2_false_even_8_18 :
    ∀ N ∈ [8, 10, 12, 14, 16, 18],
      (kkTable N).getD 2 0 ≠ conj2 N 2 ∧ (kkTable N).getD 2 0 = closedForm N 2
```

References

[1] M. García Fernández, C. Paletta and A. L. Retore, *Open-boundary integrable quantum circuits with different geometries*, arXiv:2607.02093v1 [math-ph] (2 July 2026). Section, equation and figure numbers cited in this paper are those of the compiled arXiv PDF of version 1 (58 pages), and all quotations are verbatim from it.

[2] M. García Fernández, C. Paletta and A. L. Retore, *Open-boundary integrable quantum circuits with different geometries*, Zenodo (2026), doi:10.5281/zenodo.20763606 (reference [49] of [1]; a computational-notebook deposit of 2 July 2026 holding the two Mathematica notebooks `OpenQCforDiffGeom.nb` and `AllKMatrices.nb`).

[3] M. Vanicat, *Integrable Floquet dynamics, generalized exclusion processes and "fused" matrix ansatz*, Nucl. Phys. B 929 (2018) 298–329, doi:10.1016/j.nuclphysb.2018.02.007, arXiv:1711.08884.

[4] M. Vanicat, L. Zadnik and T. Prosen, *Integrable Trotterization: Local Conservation Laws and Boundary Driving*, Phys. Rev. Lett. 121 (2018) 030606, doi:10.1103/PhysRevLett.121.030606, arXiv:1712.00431.

[5] J. Bensa and M. Žnidarič, *Fastest Local Entanglement Scrambler, Multistage Thermalization, and a Non-Hermitian Phantom*, Phys. Rev. X 11 (2021) 031019, doi:10.1103/PhysRevX.11.031019, arXiv:2101.05579.

[6] C. Paletta, U. Duh, B. Pozsgay and L. Zadnik, *Integrability and charge transport in asymmetric quantum-circuit geometries*, J. Phys. A 58 (2025) 275001, doi:10.1088/1751-8121/ade483, arXiv:2503.04673 (reference [31] of [1], the periodic-boundary construction its Appendix D compares with).

[7] G. X. Viennot, *Heaps of pieces, I: Basic definitions and combinatorial lemmas*, in: Combinatoire énumérative (Montréal 1985), Lecture Notes in Mathematics 1234, Springer, 1986, 321–350, doi:10.1007/BFb0072524.

[8] L. de Moura and S. Ullrich, *The Lean 4 theorem prover and programming language*, in: Automated Deduction — CADE 28, Lecture Notes in Computer Science, Springer, 2021, 625–635, doi:10.1007/978-3-030-79876-5_37.