Exact synthesis of small Boolean functions in three bases: an annotated atlas for n ≤ 5

Luiz Antonio Busnello (Independent Researcher, ORCID 0009-0007-5911-8141)

3 November 2026 · doi:10.5281/zenodo.23913364

1. What this document is

A map of a corpus that is mostly published; where a fact is not in a public deposit, §7 says so. It contains no new result. For each fact it says where the fact is, what certifies it, and what remains open. The cross-base laws belong to the companion paper [L], so this document stays an atlas: when it needs a law, it points to [L].

Scope: Boolean functions of n ≤ 5 variables, exact synthesis (proved minimum size, not a heuristic bound), in three bases. Out of scope: depth, unbounded fan-in, asymptotic families, multi-output functions, and anything without a checkable artifact.

Deposits are cited by letter ([A], [B], [C], [E], [L], [S], [U]; the list is at the end), and a file by its path inside the deposit.

2. The three bases, and what can be compared across them

basis	gate	free
AIG	2-input AND	inversions on edges and at the output
MIG	3-input majority	inversions on edges and at the output; constant inputs
XAG	2-input AND and 2-input XOR	inversions on edges and at the output

Two quantities per basis: **`opt`**, the minimum number of gates of a circuit, and **`tree`**, the minimum for a formula (a circuit of fan-out 1). **`gap = tree − opt`** measures what sharing buys.

A gate count in one basis is not a cost in another. The `gap` maxima `6`, `3` and `1` count AND gates, MAJ-3 gates and AND/XOR gates, and nothing here compares physical cost across bases. The counts can still be compared as numbers where one basis contains another. Every AIG circuit is an XAG circuit of the same size and, with constant inputs free, a MIG circuit of the same size (`AND(a, b) = MAJ(a, b, 0)`), and a formula stays a formula, so `opt` and `tree` in MIG and in XAG never exceed their AIG values (222 of 222 classes in the three CSVs). Neither of MIG and XAG contains the other, and `gap` has no such order: it is not monotone in the expressiveness of the basis (§3.5). The comparisons of §3.5 are of this kind, counts against counts and paying against not paying, never cost against cost.

3. n ≤ 4: complete, and published as an atlas

3.1 The table

The 222 NPN classes of 4-variable functions are resolved in all three bases, with `opt`, `tree`, `gap` and twelve structural columns computed from the truth table. Published as the dataset [C], **`10.5281/zenodo.23353481`** (v1.1; v1.0 is `10.5281/zenodo.23329555`), in the table `atlas_n4_three_bases.csv` with a checker, `verify_atlas.py`. The table joins three primary sources with the same 222 keys: the AIG table `theorem7/verify/atlas_n4.csv`

and the XAG table `census/xag-n4/npn4_xag_gap.csv` of the artifact bundle [B], and a MIG table whose `tree` column comes from a formula dynamic program, which is not in a public deposit (its values are the MIG columns of [C]).

gap	0	1	2	3	4	5	6	gap = 0	max
AIG	93	57	40	13	14	2	3	93 / 222	6
MIG	168	50	3	1	—	—	—	168 / 222	3
XAG	218	4	—	—	—	—	—	218 / 222	1

The maximum `opt` is 10 in AIG (4 classes), 7 in MIG (1 class) and 7 in XAG (12 classes).

The classes at each maximum are sets, not single witnesses:

- `gap = 6` in AIG: `0x1668`, `0x16e9`, `0x6996`, all with `opt = 9` and `tree = 15`.
- `gap = 3` in MIG: `0x6996`, the parity of four variables.
- `gap = 1` in XAG: `0x01e9`, `0x0661`, `0x1668`, `0x167e`.
- `opt = 10` in AIG: `0x1669`, `0x166b`, `0x1681`, `0x1683`.

In the AIG table, 48 of the 222 classes are `forced_multi` and 22 are canalizing (columns of `theorem7/verify/atlas_n4.csv` in [B]; [C] carries the canalizing column). The six forced-multi classes with `gap = 1` are F_4 `= 6` (`theorem7/VERIFY.md` of [B] derives the count from that table).

3.2 What backs each column

The README of [C] states the limits: `opt` comes from SAT-based exact synthesis, whose satisfiable side is a circuit checked by simulation and whose unsatisfiable side is a solver answer, not DRAT-certified class by class; `tree` comes from exact dynamic programs over formulas.

column	upper side	lower side	independent route	in [C]
AIG `opt`	circuit, simulated	solver UNSAT of our own encoder (222 of 222), no DRAT per class	Krinkin/bounds catalog (220 values imported, rederived 222 of 222); mockturtle `subgraphs_aig`, 221 of 222; DRAT for `0x1669`, `0x166b` (§3.4); A401174 rows `n ≤ 4` are ours, not independent	table only
AIG `tree`	formula, simulated	dynamic program, no solver	Madsen et al. counts; uniqueness proof; 220 witnesses (§3.3)	DP values and `frontier/`
MIG `opt`	circuit, simulated	solver UNSAT	mockturtle `mig_npn`, 222 of 222 per class (2026-10-14); Soeken et al. (DATE 2016) Table I marginals	table only
MIG `tree`	formula	dynamic program	SAT census in 220 of 222; Soeken et al. Table II marginals	table only
XAG `opt`	circuit, simulated	solver UNSAT, kissat without DRAT	optimal5 restricted to `n = 4`, class by class; mockturtle `subgraphs_xag`, 221 of 222	table only
XAG `tree`	formula	dynamic program	Lflib5 (IWLS 2019) restricted to `n = 4`, class by class	table only

The XAG agreement with optimal5 and Lflib5 is per class (222 of 222, the `n = 4` control of the join of §4.2). The MIG `opt` agreement with the mockturtle `mig_npn` database is per class, 222 of 222, and the mockturtle AIG and XAG databases agree per class in 221 of 222, the remaining class having no output there (our decoding of the three databases, 2026-10-14, not in a public deposit). The agreement with Soeken et al. is only in distribution,

because that paper publishes marginals and not per-class values; the per-class pairing of the MIG `tree` column is ours. All of these are third-party data (§5), used as agreement checks.

3.3 The AIG **`tree`** column: three routes and a proof

The independent validation of formula mode (a SAT encoder asked for a formula of `tree` gates and refuted at `tree − 1`) ends at 203 PASS, 17 UNDECIDED and 2 trivial over the 222 classes. The undecided ones are exactly the 17 classes with `tree ≥ 12` (3, 8, 2 and 4 classes at `tree` = 12, 13, 14 and 15), so that instrument covers `tree ≤ 11` and nothing above. [C] v1.1 publishes those runs in `frontier/` and says, correctly for what it contains, that for these 17 the value rests on the dynamic program alone.

The value does not rest on one route. Three were established on 2026-10-27:

1. an exhaustive dynamic program over formulas (2026-08-31), which decides all 65 536 functions without a solver and agrees with the atlas on 222 of 222 classes;
2. the published distribution of Madsen, van de Pol and Henriksen (OOPSLA 2023), whose formula size in negation normal form equals `tree_AIG` when inversions are free; expanding the 222 classes by their NPN orbits reproduces their counts, 16 of 16;
3. explicit witnesses: 220 formulas, one per non-trivial class, checked by a verifier that does not import the encoder: topology, fan-out 1, the function on all 16 rows, and the gate count equal to the atlas (1 782 gates in total).

Routes 2 and 3 and NPN invariance give a proof, not a search, conditional on the published counts of Madsen et al. (hypothesis H2: that those counts are correct): by descending induction from `t = 15` to `t = 0`, the witnessed upper bounds and the published counts force `{tree = t}` to be exactly the set of classes with witness value `t`, so no UNSAT call enters the column. The classical Khrapchenko bound does not do this job: computed class by class, it is tight in 19 of the 221 non-constant classes and closes 1 of the 17 (the parity). The values of the dynamic program are those of [C]; the program's full output, the witnesses and the uniqueness check are not in a public deposit (§7).

What stays open here is the instrument, not the values: formula-mode SAT has never decided a class above `tree = 11`.

3.4 The AIG **`opt`** ceiling

All four classes at `opt = 10` are decided in the atlas, but they are not certified alike.

- `0x1669` and `0x166b`: the `k = 9` UNSAT has a DRAT proof verified by `drat-trim` in July 2026 (4.79 GB and 3.87 GB, on two machines), and the 10-gate circuit is simulated. For `0x1669` a second `k = 9` proof (3.08 GB), regenerated from the same CNF (same sha256), was verified on 2026-10-15 and deleted; the value `opt_AIG(0x1669e996) = 13` of §4.4 rests on it. The dataset [A], `10.5281/zenodo.23329385`, holds the 18 CNFs for `k = 1..9` (`opt10-n4-certificates/cnf/`), the DRAT proofs for `k ≤ 8`, the witnesses (`witnesses.json`) and a normalization lemma (`normalization_lemma.md`). The **`k = 9`** proofs are not archived: only the CNFs, the sha256 of CNFs and proofs, and the July `drat-trim` logs are (`history/drat_hashes.txt`, `history/dt_0x1669.out`, `history/dt_0x166b.out`), so the decisive step is reproducible but not re-checkable without regenerating the proof.
- `0x1681` and `0x1683`: the value 10 is in the Krinkin/bounds catalog and was rederived by our encoder, with solver UNSAT at `k = 9`, and we hold no DRAT proof of either (searched 2026-10-27). They are measured, not certified. A different question, whether every optimum of these classes is forced-multi, was decided with symmetry breaking (3 437.6 s for `0x1681`, column `t_sec` of `theorem7/verify/atlas_n4.csv` in [B]) and closed again without symmetry breaking on 2026-10-13 (solver UNSAT without DRAT, not in a public deposit); that leg does not bear on `opt`.

The four `opt = 10` classes have `tree ≥ 12`, so their `tree` values are covered by §3.3.

3.5 Cross-base structure: see [L]

These are results of the companion paper [L]. The atlas only records where each one lives. The counts are checked against the three CSVs (§9), and the arguments are in [L].

fact	section of [L]
the gap is non-monotone in the expressiveness of the basis: 14 of the 222 classes	§2
XAG gap below AIG gap in 127 classes and above it in 0	§1
0x1668 (E_2) is the only class with a positive gap in all three bases	§3
sharing term of parity, s_n = 3(n − 2), exact for n = 3..12	§4
m*_XAG = 1 at n = 3, but not at n = 4	§5
a substitution cost law (not claimed)	not stated; §5 reports a measurement, not a law

4. n = 5: partial, and this is where the frontier is

Nothing is complete at n = 5. This section says in what sense each piece is partial.

4.1 AIG, by size

The number of 5-variable functions of exact AIG size k, for k = 0..8, is 12, 80, 640, 4420, 25744, 141680, 662820, 3071048, 12216080. Two enumerations written independently agree bit for bit on the sets for k = 6, 7, 8, and the NPN class counts at those sizes, 198, 667 and 2311, agree with Stevenson's A057241 table. Published as OEIS A401174 (approved 2026-10-22) with the dataset [E], 10.5281/zenodo.23406952 (the counts in aig5-size-counts/data/terms.tsv, the run records under aig5-size-counts/data/records/). The evidence is exhaustive enumeration by two independent programs, which needs no UNSAT; it is computation, not a checked proof (§5). Above k = 8 the enumeration does not close.

4.2 XAG over all classes, and the unit gap

Over the 616 126 NPN classes of n = 5, gap_XAG is {0: 365053, 1: 244245, 2: 6815, 3: 13}. That distribution is third-party data without a certificate: opt_XAG is C(f) of optimal5 (Goucher, from Knuth's BOOLCHAINS) and tree_XAG is L(f) of Lflib5 (IWLS 2019), valid under the hypotheses H1/H2 that those tables are optimal. Our own part is the join, published per class as gap_xag_n5_per_class.csv.gz in the artefact archive of [U], and the n = 4 control (222 of 222, §3.2).

What we establish without third-party tables is existence: the separator 0x00063357 has opt = 6 and tree = 8, by a simulated 6-gate circuit, solver UNSAT for k = 1..5 (kissat, no DRAT), a formula dynamic program for tree ≥ 8 and formula-mode SAT, UNSAT for k = 1..7, as a second route, and for tree ≤ 8 an 8-gate formula returned by the formula-mode encoder at k = 8, whose model is re-simulated (the logs sat_check_sep.log, sep_form_k1a6_vps.log, sep_form7_vps.log and tree_dp_check.log of the artefact archive of [U]). Two further n = 5 XAG results follow the same pattern and are not in a public deposit: exact opt_XAG for the 2 289 classes of five essential variables with opt ≤ 6, by exhaustive enumeration; and the three Ashenhurst separators 0x011e1ffe, 0x011fe0e1, 0x06696ff9 with (opt_XAG, tree_XAG) = (7, 9), using a read-once substitution lemma that is not yet claimed and solver UNSAT without DRAT.

So the unit gap has two refutations resting on different kinds of evidence (§5). In AIG it fails already at n = 3: gap($\oplus_3$) ≥ 3, with opt($\oplus_3$) = 6 by a DRAT-verified UNSAT and the gap bound in Lean (UnitGap.unit_gap_refuted, in formal/UnitGap.lean of [B]; its trust base includes a native_decide axiom, which work in progress is removing). In XAG it holds through n = 4 and fails at n = 5, by the simulated 6-gate circuit (opt ≤ 6) and the exact formula dynamic program (tree ≥ 8) of 0x00063357, with no UNSAT and no third-party table; the dynamic program is not machine-checked. The solver UNSAT (no DRAT) only makes opt = 6 exact, which the refutation does not need. Both are in [U], v2.1, 10.5281/zenodo.23328820.

4.3 The forced-multi family F_5

$F_5 \geq 78$ (11 lift classes and 67 non-lift classes). The table `theorem7/verify/f5_classes.csv` of [B] has 79 rows because `0x0000178e` arises as both an AND-lift and an OR-lift. The evidence differs, and `theorem7/README.md` of [B] keeps the three kinds apart: 69 of the 78 carry an `n = 5` DRAT chain of their own, five arrive by the Lean lift theorem (`LiftTower.forced_family_nonempty_at_every_arity`) applied to a certified `n = 4` base, and four, the level-9 slice, carry a DRAT-checked forced-multi refutation with `opt` and `tree` taken from the solver without proofs. Published in the artifact bundle [B], v1.2.0, `10.5281/zenodo.23597764` (concept `10.5281/zenodo.21630762`, "at least 78").

The four level-10 candidates. `0x00187e18`, `0x00192d19`, `0x00192d2d` and `0x001e9b1e` closed as solver UNSAT without proof on the forced-multi leg, 139.2 h of kissat in total, finished 2026-10-25. They are candidates, not classes: without a certificate they do not enter `f5_classes.csv`, and the floor does not move. On 2026-10-27 the memory cost of checking was measured: `drat-trim` needs 1.18× the proof size in RAM and `dsr-trim` 2.60× (the slope between the two largest of six real pairs). With 121 GB available on the server and a 3 GB margin, that puts the ceilings at proofs of about 100.3 GB and 45.5 GB respectively, an estimate from six points and not a guarantee, while these proofs are projected at about 107 to 146 GB each. The model reproduces the two real outcomes of this project (69.7 GB predicted for the 59.19 GB proof that verified, 143.4 GB for the 121.8 GB proof that died of OOM). They are left as measured: certifying them would need a host with at least 256 GB of RAM (O12).

4.4 Single named values

- `opt_MIG(PAR_5) = 6`: DRAT-verified UNSAT for `k = 1..5` without symmetry breaking (`logs/par5_drat.log` of [S]), and a simulated 6-gate circuit; the dataset [S], `10.5281/zenodo.23382607`, is published, and the note on SUM_n that states the value is an unpublished draft. On 2026-10-31 the EPFL tool percy found the same value; percy is also SAT-based, so what is independent is the formulation and the solver, not the method. percy searches MIGs with at most one complemented fanin per node, and whether that restriction is WLOG was not checked, so its agreement confirms that 6 is reachable, not the lower bound.
- `opt_AIG(0x1669e996) = 13`, conditional on Lemma S (`opt(g ⊕ y) = opt(g) + 3`, a paper proof without Lean; O13). The `k = 9` DRAT for `0x1669` it uses is the 2026-10-15 proof of §3.4, verified and then deleted; only the CNF is reproducible.

5. Certified, measured, alleged

Three states, and the atlas depends on keeping them apart.

Certified means UNSAT with a checked proof. On 2026-10-27 every DRAT proof we keep went through a second family of checker: `dsr-trim` re-emits it in LSR and `lsr-check` checks it; 57 of 57 gave `s VERIFIED UNSAT` (45.3 MB of DRAT, 111.7 MB of LSR; the outputs are not in a public deposit). For a wrong proof to pass both, each code base would need a defect of its own that accepts it, so the second family lowers the dependence on one implementation without removing it; and no checker says whether the CNF states the synthesis question, which rests on the encoder. Neither checker is verified. One rung above, `cake_lpr` (Tan, Heule and Myreen, TACAS 2021), whose correctness is proved and whose code is produced by the verified CakeML compiler, accepted the 31 of 31 certificates of `theorem7/verify/sample_certs/` in [B], in LRAT form, on 2026-10-26. The verified checker for LSR, `srcheck` (Lean 4, FormalSAT/trestle), does not compile upstream, and patching someone else's verified checker would move the trust back to us, so it is not done.

Traps for anyone reproducing this. `lsr-check` returns `rc = 0` and prints `s VALID` for a truncated proof, half a proof and an empty file; `cake_lpr` returns `rc = 0` for everything, including a missing file. The only verdict is the exact line `s VERIFIED UNSAT`, so our checks go through wrappers that require that line and never rely on the exit code of the binaries.

Measured means solver UNSAT without a proof: the AIG `opt` lower side for most classes, the XAG lower sides, `0x1681` and `0x1683`, and the four level-10 candidates. It counts as knowledge, not as a certificate.

Alleged means a third party states it and we have not certified it. In this atlas that is concrete: the optimal5 and Lflib5 tables behind the `n = 5` XAG distribution, the Krinkin/bounds catalog of AIG `opt` at `n = 4` (rederived by

our encoder), the Soeken et al. marginals, the Madsen et al. counts (H2 of §3.3), the mockturtle `mig_npn`, `subgraphs_aig` and `subgraphs_xag` databases and the `tree = 28` of Cox and Healy used for `opt_AIG(0x1669e996) = 13` (§4.4). Each is used either as an agreement check or under a named hypothesis, never as the only leg of a certified claim.

Other evidence. The three states classify answers to SAT queries, and some evidence in this atlas is none of them. Exhaustive computation, without a solver and without a formal proof: the two `n = 5` AIG enumerations (§4.1), the formula dynamic programs behind every `tree` column and behind `tree ≥ 8` for `0x00063357` (§4.2), and the enumeration of the 2 289 classes of §4.2. Formal proof in Lean, with the trust base stated in §6. Simulation of an explicit circuit or formula, which proves an upper bound and nothing else. Where this document calls a value certified it means the first state; otherwise it names the kind.

6. Lean

What is in Lean and touches the atlas, all checked by a statement comparator, which checks each proof against a separately written statement of the theorem (four configurations also passed with a real sandbox on 2026-10-07):

- `ThresholdTreeN4.threshold_n4` and `ThresholdTreeN4.tree_thm_n4`: statements about the 222 classes of `n = 4`, proved over a table imported from the census (`npnData`), so they inherit the census's trust.
- `UnitGap.unit_gap_refuted`: `gap(⊕₃) ≥ 3` in AIG (§4.2).
- `LiftTower.forced_family_nonempty_at_every_arity`: the lift theorem, whose paper proof is `theorem7/proofs/proof_lift_lemma.md` of [B], used for five of the 78 classes of F_5 (§4.3).

`UnitGap.lean` is in [B] (`formal/UnitGap.lean`); the other Lean files are not in a public deposit. Nine of the 17 statements under the comparator depend on `native_decide`, which puts the Lean compiler in the trust base and blocks any external kernel. Work in progress replaces it (10 of 12 bridge obligations proved on 2026-10-31).

7. Provenance, evidence and deposit for each fact

fact	evidence (§5)	deposit, and file inside it	open (§8)
three-base atlas `n = 4`, 222 rows	`opt`: simulated circuit + solver UNSAT; `tree`: exact DP	[C] v1.1, `atlas_n4_three_bases.csv`	O2, O3, O4
gap distributions, maxima and their witness sets	join of the three CSVs, rechecked mechanically (§9)	same	—
AIG `tree` column	DP + Madsen et al. 16/16 + uniqueness proof (under H2) + 220 witnesses	DP values and formula-mode runs (`frontier/`) in [C]; proof and witnesses not in a public deposit	O1, O5
`opt_AIG = 10` for `0x1669`, `0x166b`	DRAT verified by `drat-trim` (July 2026); proofs not archived	[A] (CNFs `k ≤ 9`, DRAT `k ≤ 8`, July logs in `history/`)	O6
`opt_AIG = 10` for `0x1681`, `0x1683`	Krinkin/bounds catalog + solver UNSAT of our encoder, no DRAT	rows of [C]	O3
MIG columns	`opt`: solver UNSAT, mockturtle `mig_npn` 222/222; `tree`: DP and SAT 220/222; marginals of Soeken et al.	rows of [C]	O4
XAG columns	`opt`: kissat UNSAT without DRAT; `tree`: DP; optimal5/Lflib5 per class	rows of [C]; source table in [B], `census/xag-n4/`	O4
$F_4 = 6$	one DRAT-verified UNSAT per class, with and without symmetry breaking (`theorem7/VERIFY.md` of [B])	[B], column `forced_multi` of `theorem7/verify/atlas_n4.csv`; not rechecked for this row	—
unit gap refuted in AIG, $\oplus_3$	DRAT + Lean (`native_decide` in trust base)	[U] v2.1, `10.5281/zenodo.23328820`; DRAT in `certificates/` and Lean in `formal/UnitGap.lean` of [B]	O7

fact	evidence (§5)	deposit, and file inside it	open (§8)
`n = 5` AIG counts, `k = 0..8`	two exhaustive enumerations; Stevenson A057241	[E]; OEIS A401174	O9
`n = 5` XAG gap distribution	third-party tables under H1/H2	[U] v2.1, `gap_xag_n5_per_class.csv.gz`	O10
separator `0x00063357`	simulated circuit; solver UNSAT without DRAT; DP	[U], the logs named in §4.2	O10
exact `opt_XAG` for 2 289 classes; three Ashenhurst separators	enumeration; solver UNSAT without DRAT; lemma not yet claimed	not in a public deposit	O10
$F_5 \geq 78$	69 `n = 5` DRAT chains, 5 Lean lift + certified base, 4 DRAT forced-multi with solver `opt/tree`	[B] v1.2.0, `theorem7/verify/f5_classes.csv`	O11
four level-10 candidates	solver UNSAT without proof	not in a public deposit	O12
checker memory model, 1.18× and 2.60×	measured on six real CNF/DRAT pairs	not in a public deposit	O12
`opt_MIG(PAR_5) = 6`	DRAT verified, no symmetry breaking; percy reaches 6 in its restricted space	[S], `logs/par5_drat.log`; the SUM_n note is an unpublished draft	—
`opt_AIG(0x1669e996) = 13`	conditional on Lemma S (paper proof); DRAT historical	not in a public deposit	O13
57 of 57 in a second checker; 31 of 31 under `cake_lpr`	`lsr-check` (unverified); `cake_lpr` (verified)	the 31 certificates in [B], `theorem7/verify/sample_certs/`; the checker outputs are not in a public deposit	O6

8. Open

#	what is open	why it matters
O1	formula-mode SAT has never decided an AIG class above `tree = 11`	the values are settled (§3.3); the instrument is unexercised in that range
O2	[C] (v1.1, 2026-10-14) predates the routes of §3.3 and says the 17 classes rest on the DP alone; the uniqueness proof and the 220 witnesses are not in a public deposit	a reader of [C] sees less than exists; a new version of [C] would close this
O3	the AIG `opt` lower side is solver UNSAT without DRAT for all classes but `0x1669` and `0x166b`, with the Krinkin catalog and the mockturtle database as third-party agreement; `0x1681` and `0x1683` have no DRAT for `opt` (their forced-multi leg was closed without symmetry breaking)	the `opt = 10` ceiling is certified for two of its four classes
O4	the MIG and XAG `opt` lower sides are solver UNSAT without DRAT; their per-class second witnesses (mockturtle 222/222 for MIG; optimal5 and mockturtle for XAG) are third-party data	the agreement is with uncertified databases
O5	[C] does not carry the circuits behind `opt` and `tree`, so its values cannot be re-derived from it (its README says so)	reproducibility of the atlas from the deposit alone
O6	`srcheck` does not compile; the `k = 9` proofs of `0x1669` and `0x166b` are not on disk, so they cannot go through `cake_lpr` without being regenerated	the two certified `opt = 10` values are checked by an unverified checker
O7	`native_decide` in the trust base of nine Lean statements	blocks an external kernel
O8	certified symmetry breaking (VeriPB or LSR)	removes "under symmetry breaking" from AIG `opt` and from part of F_5

#	what is open	why it matters
O9	`n = 5` AIG above 8 gates	the enumeration does not close
O10	the `n = 5` XAG distribution rests on optimal5/Lflib5 under H1/H2; novelty against TAOCP §7.1.2 is unchecked	the distribution and the maximum 3 are not ours to certify
O11	exact F_5	78 is a floor
O12	the four level-10 candidates	left as measured; they reopen only with a host of at least 256 GB of RAM
O13	Lemma S (`opt(g ⊕ y) = opt(g) + 3`; distinct from the symmetry-breaking condition S of the SUM_n work) has no Lean proof and no human review	the value 13 of §4.4 is conditional on it
O14	a substitution cost law: [L] does not state one, and its §5 reports only a measurement within its scope	the cross-base structure of §3.5 has no law for substitution

9. How the numbers were checked

Every number in this document comes either from a public deposit listed below or from an artifact in our private records; §7 says which for each fact. A script recomputes from the artifacts, and requires to appear in this text exactly as recomputed: the `n = 4` distributions, maxima and witness sets of §3.1; the `forced_multi` and canalizing counts; the `tree ≥ 12` split and the formula-mode tally; the Madsen counts, the witnesses and their total gate count; the counts of §3.5; the `n = 5` AIG counts and XAG distribution; $F_5 \geq 78$ and its split into lift and non-lift classes; the level-10 total; the two checker counts of §5; the 222 of §2; and the two checker slopes and memory ceilings of §4.3. It also fails on a numeric variant of any of these anywhere in the text, and it was tested by mutating each number in turn. The other numbers (DOIs, run times, proof sizes, the 69/5/4 split of F_5) and the Lean counts of §6 were checked by hand against their sources on 2026-11-01; $F_4 = 6$ was not rechecked for this document.

Use of generative AI

This work used AI models extensively, under the author's direction: Anthropic's Claude for the analysis, code and drafting, and models of other families for read-only review. The author is responsible for all claims. Correctness rests on the artifacts this document cites (checked DRAT proofs, exhaustive computations, simulated circuits and the Lean development), not on the models.

Data availability

The atlas of §3 is the dataset [C]; the other public evidence is in the deposits listed below, and §7 names, for each fact, the deposit and the file inside it. The facts that §7 marks as not in a public deposit (the uniqueness proof and witnesses of §3.3, the decoding of the mockturtle databases, two of the `n = 5` XAG results of §4.2, the level-10 runs and the checker memory model of §4.3, the second-checker outputs of §5, the Lean files other than `UnitGap.lean`, and Lemma S) are not yet in a public deposit. Third-party data are not redistributed.

Deposits cited

- [A] L. A. Busnello. opt(0x1669) = opt(0x166b) = 10: CNFs, DRAT proofs for k ≤ 8 and a normalization lemma for the last two open entries of the NPN-4 exact AIG catalog. Dataset, *Zenodo*, 2026, doi:10.5281/zenodo.23329385.
- [B] L. A. Busnello. unit-gap-artefacts: machine-checkable artefacts for refutations of Theorems 2 and 7 of arXiv:2603.08033. Version v1.2.0, *Zenodo*, 2026, doi:10.5281/zenodo.23597764 (all versions: doi:10.5281/zenodo.21630762).
- [C] L. A. Busnello. Atlas of the 222 NPN classes of 4-variable Boolean functions in three bases (AIG, MIG, XAG): circuit size, formula size, gap and structural invariants. Dataset, version 1.1, *Zenodo*, 2026, doi:10.5281/zenodo.23353481.

- [E] L. A. Busnello. Boolean functions of five variables by minimum AND-inverter graph size, sizes 0 to 8: two enumeration programs, counts and run records. Dataset, *Zenodo*, 2026, doi:10.5281/zenodo.23406952.
- [L] L. A. Busnello. Base laws for the formula–circuit gap at four inputs: AIG, MIG and XAG compared class by class. Version 1.0, *Zenodo*, 2026, doi:10.5281/zenodo.23913155.
- [S] L. A. Busnello. Exact majority-inverter graph size of SUM_5 and PAR_5: encoders, CNF hashes, DRAT records and checks. Dataset, *Zenodo*, 2026, doi:10.5281/zenodo.23382607.
- [U] L. A. Busnello. The Unit Gap Is Base-Dependent: A Mechanically Verified Refutation and a Cross-Base Census. Version 2.1, *Zenodo*, 2026, doi:10.5281/zenodo.23328820.