GPU-Accelerated Synthesis of Mixed-Boolean Arithmetic: Beyond Caching

Gabriel Bathie, Nathanaël Fijalkow
LaBRI, CNRS, Université de Bordeaux {gabriel.bathie, nathanael.fijalkow}@labri.fr

1. MBA & Learning from Examples

Mixed-Boolean Arithmetic (MBA) expressions: variables $(x_1, \dots, x_m)$, arithmetic operations $(+, \times, -)$ and bitwise operators $(\wedge, \vee, \oplus, \sim)$ over machine integers (e.g. 32 bits integers).

MBA Learning

Given a set of input-output examples $\{((x_1^{(i)}, \dots, x_m^{(i)}), y^{(i)})\}$, find an MBA expression E of minimal size that maps each input to the corresponding output:

$$\forall i, E\left(x_1^{(i)}, \dots, x_m^{(i)}\right) = y^{(i)}$$

State of the art:
- Heuristic-based local search solvers: not always optimal
- SyGuS/SMT solvers: scale poorly on larger instances

2. Example application: Code deobfuscation

Obfuscated Code: simple operations are replaced with more complex, equivalent sequences to make the program behavior hard to understand by looking at the (assembly) code.
Example:

$$x_1 + x_2 \equiv (x_1 \oplus x_2) + 2 \times (x_1 \wedge x_2)$$

Deobfuscation workflow:
- **Obfuscated function:** $f(x_1, x_2, x_3) = (x_1 \vee (2 \times (x_2 \vee x_3) - (x_2 \oplus x_3))) - (x_1 \wedge (2 \times (x_2 \vee x_3) - (x_2 \oplus x_3)))$
- **Sample inputs and evaluate** (black box)

$(2,1,9) \xrightarrow{f} 8$ $\quad (5,5,3) \xrightarrow{f} 13$ $\quad (8,7,2) \xrightarrow{f} 1$

$(7,6,8) \xrightarrow{f} 9$ $\quad (2,5,3) \xrightarrow{f} 10$ $\quad (5,4,3) \xrightarrow{f} 2$

- **Use MBA Learning to recover** f: $f(x_1, x_2, x_3) = x_1 \oplus (x_2 + x_3)$.

3. Locally Consistent Integer Encoding of MBAs

Bijection from $\{0, \dots, N_s - 1\}$ to formulas of size s: Ranking/Unranking

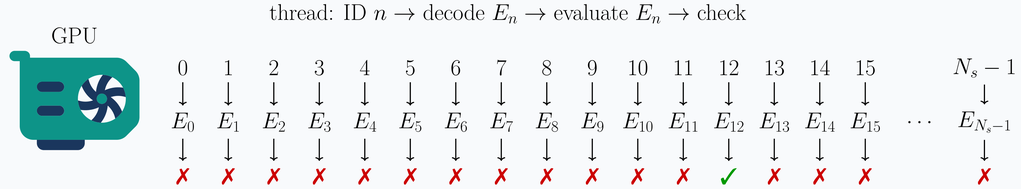
- Easy to compute
- Nearby integers are mapped to similar formulas



On GPU:
- One expression per GPU thread (determined by thread ID).
- Locally consistent encoding ⇒ minimal thread divergence.

4. Proposed Architecture: SIMBA 🦁

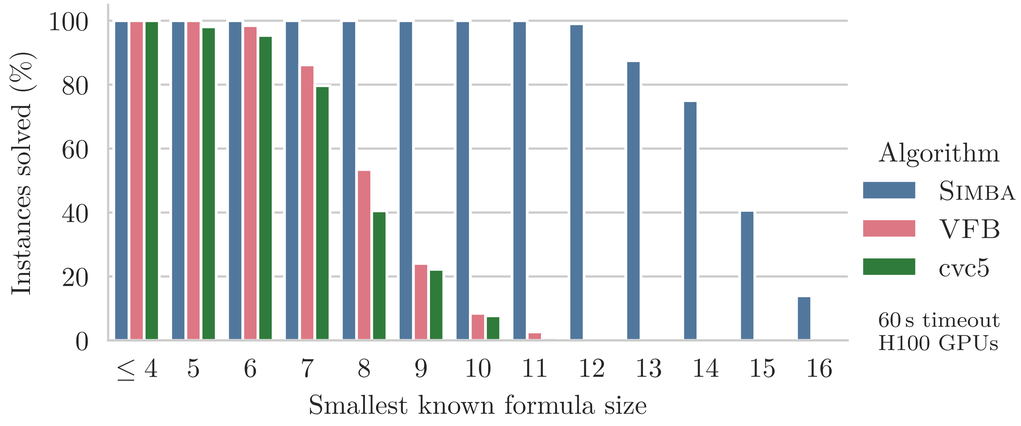
for size $s = 1, \dots,$ do in parallel for $n = 0, \dots, N_s - 1$:
thread: ID $n \to$ decode $E_n \to$ evaluate $E_n \to$ check



Solution found!
Optimality guaranteed ✓

5. Experimental results

SIMBA (ours) vs other exact MBA synthesizers: CVC5 (CPU-based SMT Solver) and VFB (GPU-accelerated, domain specific).



Ablation Study: Randomized Encoding
- **Goal:** validate the importance of locally consistent encoding.
- **Idea:** apply a random permutation to thread IDs to break locality.
- **Result:** 4× slower.

6. Summary & Key Takeaways

- **The Bitter Lesson:** In high-dimensional output spaces, using brute force is faster than using a hash table for caching.
- **Locally Consistent Encoding:** Reduces thread divergence and improves GPU utilization.
- **SIMBA:** Open-source GPU-accelerated tool for MBA synthesis.

Code & Paper

LaTeX TikZposter