

Hardware accelerated intelligent theorem proving

Jan Heemstra

Eindhoven University of Technology, the Netherlands

GPU accelerated theorem proving

- GPUs really not made for theorem proving
- Are they still useful?
- Can we apply machine learning?

Videogames

- Problem: draw triangles fast
- Observation: we are doing the same operations many times
- Solution: stream multiprocessors

Stream multiprocessors

- Multiple ALUs with single instruction pointer
- Must ALL perform the same elementary instruction
 - Or run idle
- Sequentially 1 to 2 orders of magnitude slower than CPU
 - Ballpark estimate for similarly dated and priced hardware
- In parallel all SMs are 1 to 2 orders of magnitude faster than a CPU
 - Ballpark estimate for similarly dated and priced hardware

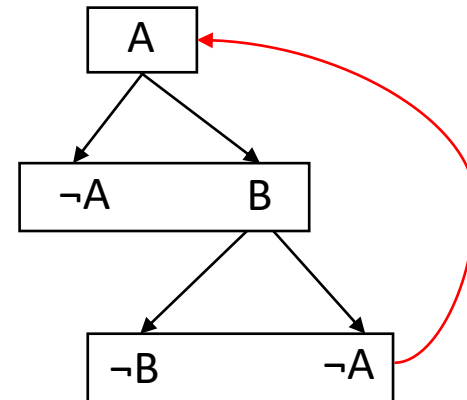
Thread divergence

- Goal: minimize thread divergence
- Easier for machine learning and physics simulations
- Hard for complex tasks (such as first order theorem proving)

Tableaux calculus

Unification

$f(x, B) \quad f(A, y) \quad \rightarrow f(A, B)$
 $f(x, B, x) \quad f(A, y, y) \quad \rightarrow \text{☹}$
 $f(x, g(x)) \quad f(g(A), y) \quad \rightarrow f(g(A), g(g(A)))$
 $f(x, g(x)) \quad f(y, y) \quad \rightarrow \text{☹} \quad \text{Occurs check}$



Tableaux calculus

- SAT \rightarrow 1st order
- Predicate logic has unique* most general unifiers
- Predicate logic is relatively complex
 - Predicate symbols, function symbols, different arities
- Lambda-free higher order logic also has unique* most general unifiers
- Lambda-free higher order logic is simple
 - Application (and of course constants and variables)
 - (hopefully) less thread divergence!

Tableaux calculus

- Substitutions are stored in small local dictionary
- Unification uses a custom stack (CUDA recursion bad)
- History is kept track of, proof steps are reversible

Proof exploration

- Iterative deepening is relatively hard and requires synchronization
- Randomly try steps instead
 - Far worse than iterative deepening, but easier to implement
- 1 million runs
 - Max 64 random steps for each run
 - If no proof is found: start over

Intermediate results

- M40 dataset
- Our method
 - 4783 out of 32444 proofs found (some theorems were excluded due to size)
 - On average 1 second is spent on each theorem (using an RTX 3080)
 - 64 million inferences per second

Machine learning

- Tree neural network
 - Less advanced than graph neural network
- Neural Network evaluation is slow
- May detriment inferences per second
- Solution: evaluate neural network ahead of time

Machine learning

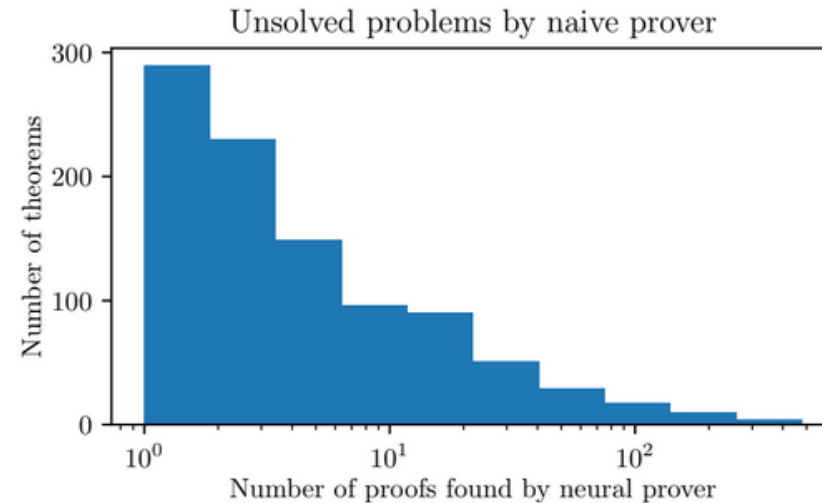
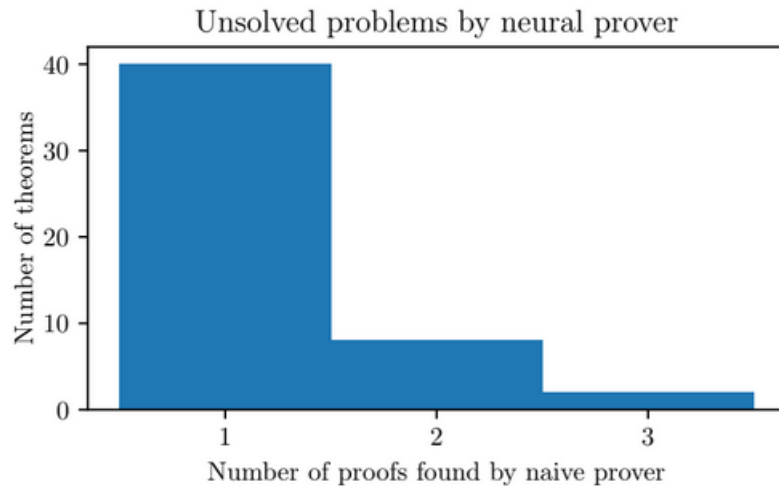
- Run prover
 - save proof steps of successful proofs
- Tally extension steps as pairs of literals, and normalize
- Train neural network on pairs of literals
 - Network has no state information
- Evaluate neural network on pairs of literals
- Run prover again, using evaluation as weights

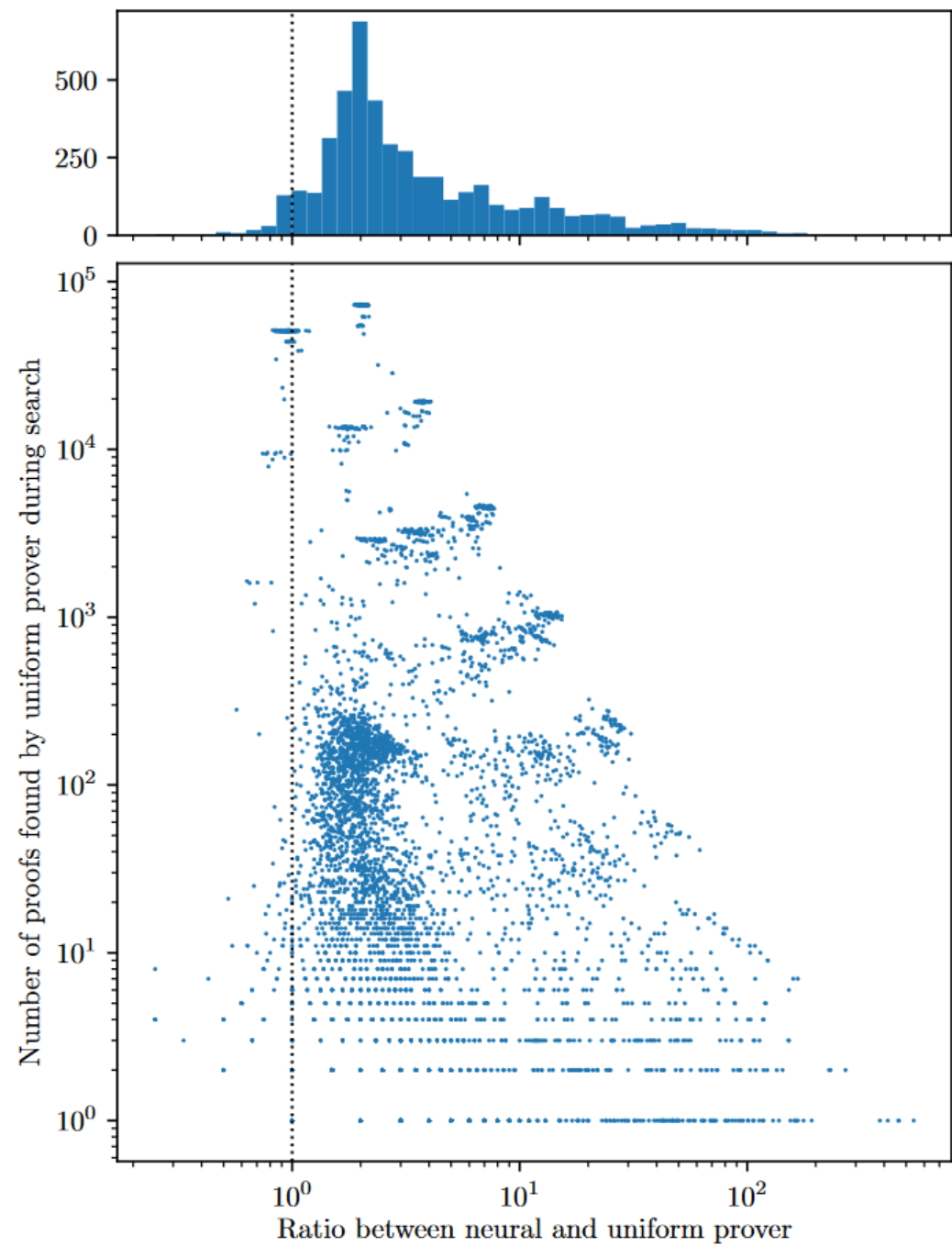
Machine learning

- Uniform prover: 4783 out of 32444 proofs found
- Neural prover: 5698 out of 32444 proofs found
- 965 newly proven problems
- 50 out of 4783 problems not proven by neural prover
- Any regressions?

(no) regressions

- Number of proofs found corresponds to Bernoulli distribution
- No regression hypothesis is plausible
- No improvement hypothesis is implausible





Future work

- Implement iterative deepening
 - Monte Carlo
- Better machine learning
 - Graph neural networks
 - Clause selection