Can Pigeonhole Principle Definitions Be Learned?

Chad E. Brown, Mikoláš Janota

Czech Technical University in Prague

September 2025

Outline

Motivation

- Pigeonhole problems are challenging for SAT solvers.
- (Haken 85) Expenential Resolution Proofs
- (Cook 76) Relatively Short Proofs Extended Resolution Proofs
- We solve them using Cook's approach with Extended SAT solving
- Can this be learned?

Goals

- Benchmark higher order ATPs on realistic higher-order mathematical problems.
- Replace large parts of proof scripts with automated calls.
- Study proof reconstruction for ATP-generated proofs.

Pigeonhole in SAT

- n pigeons, $n-1$ holes
- $P_{i,j}$ atom meaning Pigeon i is in Hole j
- $(n-1)\binom{n}{2}$ 2-literal clauses – no 2 pigeons in the same hole
- n clauses with $n-1$ literals saying each pigeon is in at least one hole

CaDiCaL Results

- CaDiCaL 2.1.3: 10 pigeons in 3.2s; 11 in 19.6s and 12 in 5.5m
- CaDiCaL 1.3.1: 10 pigeons in 0.4s; . . . 14 in about 5m

Operations

- **Definition$_2$:** Define q as $p \vee p'$ for 2 propositional variable p and p'.
- **Definition$_3$:** Define q as $p \vee p' \wedge p''$ for 3 propositional variable p, p' and p''.
- **Lemma** Prove a new clause and add it to the clause set.
- **Delete** Delete a clause from the clause set.

Extended SAT Solving

- Start with a clause set S to refute.
- **Definition$_2$:** Add clauses for $q \Leftrightarrow p \vee p'$.
- **Definition$_3$:** Add clauses for $q \Leftrightarrow p \vee p' \wedge p''$.
- **Lemma** Prove a lemma clause $l_1 \vee \cdots \vee l_n$ from S by calling a SAT solver with S and unit clauses $\neg l_i$. If successful, add the lemma clause to S.
- **Delete** Delete a clause from S.
- Call a SAT solver on the clause set S.

Cook's Recipe

- Use a sequence of definitions
- and a sequence of lemmas
- to reduce the problem with n pigeons and $n-1$ holes
- to the problem with $n-1$ pigeons and $n-2$ holes.
- Then delete the old clauses.

Cook's Recipe at 15

- Assume 15 pigeons. Want to reduce to 14.
- Start with 1485 clauses.
- Define $Q_{i,j}$ as $P_{i,j} \vee (P_{i,13} \wedge P_{14,j})$.
- Q would inject the first 14 pigeons into the first 13 holes.
- Prove lemmas with Q corresponding to the 14 case.
- Delete the old clauses.

Modified Recipe at 15

- Assume 15 pigeons. Want to reduce to 14.
- Define $Q_{i,j}$ as $P_{i,j} \vee (P_{i,13} \wedge P_{14,j})$.
- Prove lemmas with Q corresponding to the 14 case.
- Plus symmetry reduction:
  - Make extra definitions with just disjunctions to state and prove lemmas
  - $(\bigvee_{i' \in \{0,\ldots,i-1\}} Q_{i',j}) \Rightarrow \neg Q_{i,j}$.
- Repeat until we're at 11 pigeons and then call the SAT solver.
- Total time: Just over 3 minutes

Can This Be Learned?

- Is it possible to "learn" something to suggest the appropriate operations?
- Given the current clause set, the suggestor would suggest either:
  - Make a **definition** $p \vee p'$ (suggesting the p and p').
  - Make a **definition** $p \vee p' \wedge p''$ (suggesting the p, p' and p'').
  - Suggest and prove a **lemma** clause.
  - **Delete** a clause.
  - Just call a SAT solver to **finish**.

Training Data

- Training data is straightforward.
- While following the recipe at each step:
- We know the current clause set.
- We know the option the recipe chooses.
- Issue: sometimes the order matters and sometimes it doesn't.

Training Data

- For 30 pigeons:
- 70000 instances of training data.
- 15219 for definitions.
- 7809 for lemmas.
- 51063 for deleting clauses.

Possible Tests

- There are obvious variants of Pigeonhole to test on.
- Reorder literals and clauses.
- Drop literals from long clauses (Pigeon 5 is not allowed in Hole 7).
- Does the learned suggestor work more generally?

Conclusion

- Cook's recipe allows us to solve Pigeonhole problems with Extended SAT solving.
- The recipe also gives training data.
- **Question:** Can the recipe be learned from the training data?
- **Answer:** I don't know.

Acknowledgments

Supported by the Czech Ministry of Education, Youth and Sports within the dedicated program ERC CZ under the project POSTMAN no. LL1902.