Fast Normalization in the HOL Theorem Prover

Joe Hurd

joe.hurd@cl.cam.ac.uk

University of Cambridge

The Problem with CNF

Sometimes converting terms to CNF makes their size explode:

$$\text{CNF}\left(\begin{array}{l} (a_0 \wedge a_1 \wedge a_2 \wedge a_3) \vee (b_0 \wedge b_1 \wedge b_2 \wedge b_3) \vee \\ (c_0 \wedge c_1 \wedge c_2 \wedge c_3) \vee (d_0 \wedge d_1 \wedge d_2 \wedge d_3) \end{array}\right)$$

$$=$$

$$(a_3 \vee b_3 \vee c_3 \vee d_0) \wedge (a_2 \vee b_3 \vee c_3 \vee d_0) \wedge$$
$$(a_1 \vee b_3 \vee c_3 \vee d_0) \wedge (a_0 \vee b_3 \vee c_3 \vee d_0) \wedge$$
$$\ldots \text{992 more atoms} \ldots$$
$$(a_0 \vee b_3 \vee c_3 \vee d_3) \wedge (a_1 \vee b_3 \vee c_3 \vee d_3) \wedge$$
$$(a_2 \vee b_3 \vee c_3 \vee d_3) \wedge (a_3 \vee b_3 \vee c_3 \vee d_3)$$

Disastrous if we're converting to CNF for a SAT solver

Definitional CNF

Definitional CNF guarantees the size of normalized terms will be linear in the size of original terms:

$$\text{DEF\_CNF}\left(\begin{array}{l}(a_0 \land a_1 \land a_2 \land a_3) \;\lor\; (b_0 \land b_1 \land b_2 \land b_3) \lor \\ (c_0 \land c_1 \land c_2 \land c_3) \;\lor\; (d_0 \land d_1 \land d_2 \land d_3)\end{array}\right)$$

$$=$$

$$\begin{array}{l}\exists\, v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}. \\ \quad (v_{11} \lor \neg d_0 \lor \neg v_{10}) \;\land\; (v_{10} \lor \neg v_{11}) \;\land\; (d_0 \lor \neg v_{11}) \land \\ \quad (v_{10} \lor \neg d_1 \lor \neg v_9) \;\land\; (v_9 \lor \neg v_{10}) \;\land\; (d_1 \lor \neg v_{10}) \land \\ \qquad\qquad \ldots \text{59 more atoms} \ldots \\ \quad (v_0 \lor \neg v_1) \;\land\; (a_1 \lor \neg v_1) \;\land\; (v_0 \lor \neg a_2 \lor \neg a_3) \land \\ \quad (a_3 \lor \neg v_0) \;\land\; (a_2 \lor \neg v_0) \;\land\; (v_2 \lor v_5 \lor v_8 \lor v_{11})\end{array}$$

Definitional CNF by Inference

- Given an input term t, it's easy to generate the definitional CNF normalized term t'.
- This allows a fast oracle implementation of normalization into definitional CNF:

$$\{\mathit{ORACLE\_SAYS}\} \vdash t \iff t'$$

- We'd prefer a fully-expansive HOL proof that t and t' are logically equivalent:

$$\vdash t \iff t'$$

- Unfortunately, the naive algorithm to derive this theorem is (at least) quadratic in the size of t.

Fast Definitional CNF

- Can use a technique invented by Harrison to perform BDD operations by fully-expansive proof.
- Runs in (nearly) linear time, by making extensive use of proforma theorems.
- Essential part of this: instead of introducing many new boolean variables, we use one variable vector:

$$
\begin{array}{l}
\mathsf{FAST\_DEF\_CNF}\,(\cdots) \\
\qquad = \\
\exists\, v : \mathbb{N} \to \mathbb{B}. \\
\quad (v(11) \lor \neg d_0 \lor \neg v(10)) \;\land\; (v(10) \lor \neg v(11)) \;\land \\
\quad (d_0 \lor \neg v(11)) \;\land\; (v(10) \lor \neg d_1 \lor \neg v(9)) \;\land\; \cdots
\end{array}
$$

Performance Results

We compare the different methods on the `ADD4` term (from hardware verification) containing 1111 atoms.

Operation on the `ADD4` Term	**Time (s)**	**Infs.**
Definitional NNF	10.610	7677
Oracle definitional CNF	0.390	0
Naive definitional CNF	122.620	12034
Fast definitional CNF	28.800	238258
Applying the `zCHAFF` solver	4.680	0

Observe that the fast method uses more HOL inference steps than the naive method, but takes much less time.