Hard Cases for SAT Solvers

William Gasarch-U of MD

Example

The AND of the following:

1. $x_{11} \vee x_{12}$
2. $x_{21} \vee x_{22}$
3. $x_{31} \vee x_{32}$
4. $\neg x_{11} \vee \neg x_{21}$
5. $\neg x_{11} \vee \neg x_{31}$
6. $\neg x_{21} \vee \neg x_{31}$
7. $\neg x_{12} \vee \neg x_{22}$
8. $\neg x_{12} \vee \neg x_{32}$
9. $\neg x_{22} \vee \neg x_{32}$

Example

The AND of the following:

1. $x_{11} \vee x_{12}$
2. $x_{21} \vee x_{22}$
3. $x_{31} \vee x_{32}$
4. $\neg x_{11} \vee \neg x_{21}$
5. $\neg x_{11} \vee \neg x_{31}$
6. $\neg x_{21} \vee \neg x_{31}$
7. $\neg x_{12} \vee \neg x_{22}$
8. $\neg x_{12} \vee \neg x_{32}$
9. $\neg x_{22} \vee \neg x_{32}$

This is Pigeonhole Principle: x_{ij} is putting ith pigeon in j hole!

Example

The AND of the following:

1. $x_{11} \vee x_{12}$
2. $x_{21} \vee x_{22}$
3. $x_{31} \vee x_{32}$
4. $\neg x_{11} \vee \neg x_{21}$
5. $\neg x_{11} \vee \neg x_{31}$
6. $\neg x_{21} \vee \neg x_{31}$
7. $\neg x_{12} \vee \neg x_{22}$
8. $\neg x_{12} \vee \neg x_{32}$
9. $\neg x_{22} \vee \neg x_{32}$

This is Pigeonhole Principle: x_{ij} is putting ith pigeon in j hole!
Can't put 3 pigeons into 2 holes! So Fml is NOT satisfiable.

PHP: Pigeon Hole Principle

Let $n < m$. n is NUMBER OF HOLES, m is NUMBER OF PIGEONS. x_{ij} will be thought of as Pigeon i IS in Hole j.

Definition

PHP_n^m is the AND of the following:

1. For $1 \le i \le m$

$$x_{i1} \vee x_{i2} \vee \cdots \vee x_{in}$$

   (Pigeon i is in SOME Hole.)

2. For $1 \le i_1 < i_2 \le m$ and $1 \le j \le n$

$$\neg x_{i_1 j} \vee \neg x_{i_2 j}$$

   (Hole j does not have BOTH Pigeon i_1 and Pigeon i_2.)

NOTE: PHP_n^m has nm **VARS** and $O(mn^2)$ **CLAUSES** and is NOT satisfiable.

What is Known

1. If $n < m$ then PHP_n^m is not satisfiable.
2. The proof of this is by the Pigeon hole principle and not by Truth Table, it was by mathematical reasoning.
3. There is a proof technique called **Resolution** that is used to show formulas are **not satisfiable** . It is known that resolution proofs that PHP_n^m is not satisfiable are large.
4. Our speculation is that the SAT Solvers we have been studying will take a long time on PHP_n^m.
5. Try our out SAT solvers on PHP_n^{n+1}, PHP_n^{n+2}, ... and see if it takes a long time. See what happens as the m gets bigger.