Example of SMT Eager Evaluation

November 14, 2010

A gap I left in my presentation on eager evaluation of SMT was an example of translating from F_{arith} (that is, our integer linear program) to F_{bool}.

This document will fill that gap, and explain how to translate those inequalities into CNF.

1 SAT Example

Consider the following simple example (all clauses implicitly ANDed together):

$$
\begin{aligned}
P \vee (x \leq 4) \\
Q \vee (y \leq 3) \\
R \vee (x + y \geq 8) \\
(x \leq 2)
\end{aligned}
$$

The core idea is that we replace each inequality with a new CNF variable, and then ensure that those variables are true **iff** there exists assignments to the variables such that those inequalities hold.

First we can make the new variables:

$$
\begin{aligned}
P \vee A \\
Q \vee B \\
R \vee C \\
D
\end{aligned}
$$

To establish the relationship, we could try to say something like $A \leftrightarrow x \leq 5$, but that puts us back in square 1. But wait – the whole point is that all of these operators are *transitive*, so we can simply phrase things entirely in terms of each other. For example, we know that if $x \leq 2$ then surely $xleq4$. That gives us a clause: $D \rightarrow A$. Not the only one needed, but a starting point.

$D \rightarrow A$: As stated before, if x is less than or equal to 2, then it must be less than or equal to 4.

$A \wedge B \rightarrow \neg C$: If both x and y are less than 4 and 3, then they could not possible add to something greater than eight. This *also* captures the idea that if they *do* exceed 8, then either A or B must be false. After CNF conversion, $A \wedge B \rightarrow \neg C$ is the same as $C \rightarrow \neg A \vee \neg B$.

So our new big pile of CNF statements are as follows:

$$\begin{aligned} P \vee A \\ Q \vee B \\ R \vee C \\ D \\ \neg D \vee A \\ \neg A \vee \neg B \vee \neg C \end{aligned}$$

Clearly the original statement was SAT, and this one is no different.

2 UNSAT Example

As a quick example of an UNSAT example, consider the following situation:

$$\begin{aligned} x \leq 2 \\ x \geq y \\ y \geq 3 \end{aligned}$$

We would re-write the statements as A, B, and C, and then establish the following facts (our translation program knows how to do this by definition):

$$\begin{aligned} A \wedge B \rightarrow \neg C \\ C \wedge B \rightarrow \neg A \end{aligned}$$

We see that those are really the same statement in CNF form: $\neg A \vee \neg B \neg C$. We end up with the final CNF:

$$\begin{aligned} A \\ B \\ C \\ \neg A \vee \neg B \vee \neg C \end{aligned}$$

This is clearly UNSAT, and intuitively it captures the idea that for the final clause to be satisfied, we would have to ignore at least one of our original inequalities.