Discussion on Octagon

March 23, 2021

1 Draft

Synthesizing an Octagon Predicate p

Definition 1 (Octagon)**.** *Given a set of variables X where all variables in the set belongs to a numerical set $\mathbb{I}$, which can be $\mathbb{Z}, \mathbb{R}$ or $\mathbb{Q}$. We call* octagonal constraint *any constraint of the form $\pm x_i \pm x_j \geq c$ where $c \in \mathbb{I}$ and $x_i, x_j \in X$. An* octagon *is the set of points that satisfies the conjunction of all octagonal constraints.*

Assume the program we consider is affine linear. From the definition of incorrectness logic and the iteration rule, our target is to synthesize a predicate $p(\mathbf{x}, n)$ for a loop program where the update of the loop body can be expressed as $\mathbf{x}' = M\mathbf{x}$, s.t.

$$\models \forall \mathbf{x}.n.(p(\mathbf{x}, n+1) \implies \exists \mathbf{y}.\mathbf{x} = M\mathbf{y} \wedge p(\mathbf{y}, n))$$

After the elimination of the existential quantifier we get:

$$\models \forall \mathbf{x}.n.(p(\mathbf{x}, n+1) \implies p(k_0(\mathbf{x} - \mathbf{c}) + k_1\mathbf{v}_i, n))$$

Example 1. *We first consider the simplest example where $X = \{x, n\}$, i.e. $\mathbf{x}$ only contains one variable. We assume the update of the program is $x' = ax + b$. The octagon is equivalently given by the form:*

$$\begin{aligned} x + y \geq& \mathcal{A}_{x,y} \\ x - y \geq& \mathcal{B}_{x,y} \\ -x + y \geq& \mathcal{C}_{x,y} \\ -x - y \geq& \mathcal{D}_{x,y} \end{aligned}$$

where $x, y \in X$.

For this example, then the constraint system S_1 of $p(\mathbf{x}, n+1)$ can be given as:

$$\begin{aligned}
2x & \qquad & \geq \mathcal{A}_{x,x}\\
 & & 0 \geq \mathcal{B}_{x,x}\\
 & & 0 \geq \mathcal{C}_{x,x}\\
-2x & & \geq \mathcal{D}_{x,x}\\
x+n & & +1 \geq \mathcal{A}_{x,n}\\
x-n & & +1 \geq \mathcal{B}_{x,n}\\
-x+n & & +1 \geq \mathcal{C}_{x,n}\\
-x-n & & +1 \geq \mathcal{D}_{x,n}\\
+2n & & +2 \geq \mathcal{A}_{n,n}\\
 & & 0 \geq \mathcal{B}_{n,n}\\
 & & 0 \geq \mathcal{C}_{n,n}\\
-2n & & -2 \geq \mathcal{D}_{n,n}
\end{aligned}$$

Similarly, from the fact that $\mathbf{y} = [y] = [\frac{1}{a}x - \frac{b}{a}]$, *we can also derive a system* S_2 *for* $p(\mathbf{y}, n)$*:*

$$\begin{aligned}
\frac{2}{a}x & \qquad & -\frac{2b}{a} \geq \mathcal{A}_{x,x}\\
 & & 0 \geq \mathcal{B}_{x,x}\\
 & & 0 \geq \mathcal{C}_{x,x}\\
-\frac{2}{a}x & & \frac{2b}{a} \geq \mathcal{D}_{x,x}\\
\frac{1}{a}x+n & & -\frac{b}{a} \geq \mathcal{A}_{x,n}\\
\frac{1}{a}x-n & & -\frac{b}{a} \geq \mathcal{B}_{x,n}\\
-\frac{1}{a}x+n & & +\frac{b}{a} \geq \mathcal{C}_{x,n}\\
-\frac{1}{a}x-n & & +\frac{b}{a} \geq \mathcal{D}_{x,n}\\
+2n & & \geq \mathcal{A}_{n,n}\\
 & & 0 \geq \mathcal{B}_{n,n}\\
 & & 0 \geq \mathcal{C}_{n,n}\\
-2n & & \geq \mathcal{D}_{n,n}
\end{aligned}$$

Target of the synthesis is to synthesize the unknown parameter $\{\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}\}$ *s.t.* $\models \forall x.n.(S_1 \implies S_2)$.

According to the method in previous SAS'04, e.g. if we consider the implication

$$\forall \mathbf{x}.(S_1 \implies \frac{2}{a}x - \frac{2b}{a} \geq \mathcal{A}_{x,x})$$

By introducing the symbol $\lambda^{\mathcal{A}_{x,x}}_{\mathcal{B}_{x,x}}$, *we mean the parameter introduced during the application of Farkas' Lemma from the equation with parameter* $\mathcal{B}_{x,x}$ *to the equation in* S_2 *with parameter* $\mathcal{A}_{x,x}$. *Then for this instance of implication we have* $\exists \lambda^{\mathcal{A}_{x,x}}_{\mathcal{A}_{x,x}} \ldots \lambda^{\mathcal{A}_{x,x}}_{\mathcal{D}_{n,n}}$.

$$
\begin{array}{llllllll}
+\,2\lambda^{\mathcal{A}_{x,x}}_{\mathcal{A}_{x,x}} & & -2\lambda^{\mathcal{A}_{x,x}}_{\mathcal{D}_{x,x}} + \lambda^{\mathcal{A}_{x,x}}_{\mathcal{A}_{x,n}} & +\lambda^{\mathcal{A}_{x,x}}_{\mathcal{B}_{x,n}} - \lambda^{\mathcal{A}_{x,x}}_{\mathcal{C}_{x,n}} & -\lambda^{\mathcal{A}_{x,x}}_{\mathcal{D}_{x,n}} & & & = \dfrac{2}{a} \\
 & & +\,\lambda^{\mathcal{A}_{x,x}}_{\mathcal{A}_{x,n}} & -\lambda^{\mathcal{B}_{x,x}}_{\mathcal{B}_{x,b}} + \lambda^{\mathcal{A}_{x,x}}_{\mathcal{C}_{x,n}} & -\lambda^{\mathcal{A}_{x,x}}_{\mathcal{D}_{x,n}} + 2\lambda^{\mathcal{A}_{x,x}}_{\mathcal{A}_{n,n}} & & -2\lambda^{\mathcal{A}_{x,x}}_{\mathcal{D}_{n,n}} & = 0 \\
-\,\lambda^{\mathcal{A}_{x,x}}_{\mathcal{A}_{x,x}}\mathcal{A}_{x,x} & -\lambda^{\mathcal{A}_{x,x}}_{\mathcal{B}_{x,x}}\mathcal{B}_{x,x} - \lambda^{\mathcal{A}_{x,x}}_{\mathcal{C}_{x,x}}\mathcal{C}_{x,x} & -\lambda^{\mathcal{A}_{x,x}}_{\mathcal{D}_{x,x}}\mathcal{D}_{x,x} + (1-\mathcal{A}_{x,n})\lambda^{\mathcal{A}_{x,x}}_{\mathcal{A}_{x,n}} & +(1-\mathcal{B}_{x,n})\lambda^{\mathcal{A}_{x,x}}_{\mathcal{B}_{x,n}} + (1-\mathcal{C}_{x,n})\lambda^{\mathcal{A}_{x,x}}_{\mathcal{C}_{x,n}} & +(1-\mathcal{D}_{x,n})\lambda^{\mathcal{A}_{x,x}}_{\mathcal{D}_{x,n}} + (2-\mathcal{A}_{n,n})\lambda^{\mathcal{A}_{x,x}}_{\mathcal{A}_{n,n}} & -\mathcal{B}_{n,n}\lambda^{\mathcal{A}_{x,x}}_{\mathcal{B}_{n,n}} - \mathcal{C}_{n,n}\lambda^{\mathcal{A}_{x,x}}_{\mathcal{C}_{n,n}} & +(2-\mathcal{D}_{n,n})\lambda^{\mathcal{A}_{x,x}}_{\mathcal{D}_{n,n}} & = -\dfrac{2b}{a} - \mathcal{A}_{x,x} \\
-\,\lambda^{\mathcal{A}_{x,x}}_{\mathcal{A}_{x,x}}\mathcal{A}_{x,x} & & -\lambda^{\mathcal{A}_{x,x}}_{\mathcal{D}_{x,x}}\mathcal{D}_{x,x} + (1-\mathcal{A}_{x,n})\lambda^{\mathcal{A}_{x,x}}_{\mathcal{A}_{x,n}} & +(1-\mathcal{B}_{x,n})\lambda^{\mathcal{A}_{x,x}}_{\mathcal{B}_{x,n}} + (1-\mathcal{C}_{x,n})\lambda^{\mathcal{A}_{x,x}}_{\mathcal{C}_{x,n}} & +(1-\mathcal{D}_{x,n})\lambda^{\mathcal{A}_{x,x}}_{\mathcal{D}_{x,n}} + (2-\mathcal{A}_{n,n})\lambda^{\mathcal{A}_{x,x}}_{\mathcal{A}_{n,n}} & & +(2-\mathcal{D}_{n,n})\lambda^{\mathcal{A}_{x,x}}_{\mathcal{D}_{n,n}} & = -\dfrac{2b}{a} - \mathcal{A}_{x,x}
\end{array}
$$

The first three equations in page 3 are directly derived from the method.

Due to the fact that types of $\mathcal{B}_{x,x}$ and $\mathcal{C}_{x,x}$ can be directly assigned to 0 and not influencing the result.

Without any assumption on the parameters, it seems the number of equations are not enough to remove all the nonlinear nomials in the equation of constant column.

However, observe the equation in the second line, we can add some assumptions according to the sum is 0:

$$+\lambda^{\mathcal{A}_{x,x}}_{\mathcal{A}_{x,n}} \qquad -\lambda^{\mathcal{B}_{x,x}}_{\mathcal{B}_{x,b}} + \lambda^{\mathcal{A}_{x,x}}_{\mathcal{C}_{x,n}} \qquad -\lambda^{\mathcal{A}_{x,x}}_{\mathcal{D}_{x,n}} + 2\lambda^{\mathcal{A}_{x,x}}_{\mathcal{A}_{n,n}} \qquad -2\lambda^{\mathcal{A}_{x,x}}_{\mathcal{D}_{n,n}} = 0$$

i.e. $\lambda^{\mathcal{A}_{x,x}}_{\mathcal{A}_{x,n}} = \lambda^{\mathcal{B}_{x,x}}_{\mathcal{B}_{x,b}}$, $\lambda^{\mathcal{A}_{x,x}}_{\mathcal{C}_{x,n}} = \lambda^{\mathcal{A}_{x,x}}_{\mathcal{D}_{x,n}}$ *and* $\lambda^{\mathcal{A}_{x,x}}_{\mathcal{A}_{n,n}} = \lambda^{\mathcal{A}_{x,x}}_{\mathcal{D}_{n,n}}$.

These assumptions can help to remove the nonlinear nomials.

But this situation only occurs when there is only one nomial with variable in the implied equation. If another equation is taking into consider, e.g. $\frac{1}{a}x + n - \frac{b}{a} \geq \mathcal{A}_{x,n}$. *The first two equation is similar to that of in page three except the RHS of the second equation in page 3 is not* 0. *Observing that we can make use of equation* 1 *and equation* 2 *to produce equation whose RHS is* 0, *thus resulting in a familiar analysis like before.*

Hence, we can remove the nonlinear constraint by simply applying some assumptions obtained from the equations which is only related to λ*s'.*

This is just a trivial case where two variables $\{x, n\}$ *are considered. Can we prove that this tactic can be applies to a system with* $k \in \mathbb{N}$ *variables? Seemingly yes, maybe by induction on the number of variables (A detailed proof needed.). Stopped here, deal with it later.*