EF-SMT

Ashish Tiwari
SRI International

EF-SMT

$$\exists \vec{a} : \forall \vec{x} : \phi(\vec{a}, \vec{x})$$

Important consideration: theory of ϕ

Many problems can be transformed into $EF\phi$ problems

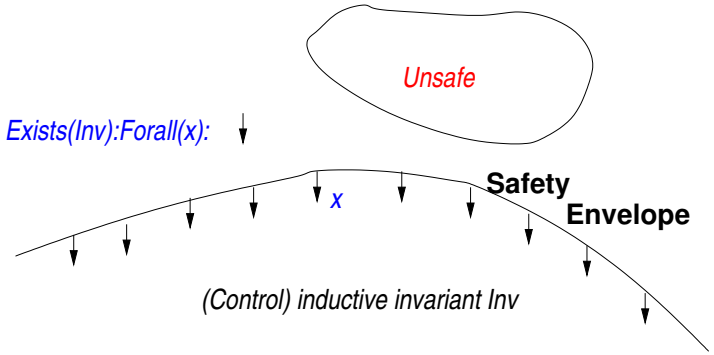
Why EFϕ?

$$M(\vec{c}, \vec{u}) \models \phi$$
$$\Uparrow$$
$$\exists \Phi, \vec{c}, \vec{u} : ((M(\vec{c}, \vec{u}) \models \Phi) \ \wedge \ (\Phi \Rightarrow \phi))$$
$$\Uparrow$$
$$\exists \Phi, \vec{c} : \forall \vec{x} : \exists \vec{u} : \ \text{quantifier-free FO formula}$$
$$\Uparrow$$
$$\exists \vec{d}, \vec{c} : \forall \vec{x} : \exists \vec{u} : \ \text{quantifier-free FO formula}$$

The last step is performed by choosing a template for Φ

Example 1: Controller Synthesis for Safety

Φ = (Global) Control inductive invariant



$\exists$(control inductive invariant that implies safety)

$\exists(set)$: (set is a control inductive invariant) and (set implies safety)

$\exists(V(\vec{x}) \geq 0) : \forall \vec{x} : \exists u : (V(\vec{x}) = 0 \;\Rightarrow\; V(F(\vec{x}, u, 0^+)) \geq 0)$

$\wedge(V(\vec{x}) \geq 0 \Rightarrow \text{Safe}(\vec{x}))$

Example 1: Controller Synthesis for Safety

Two things:

- Safe controller will be a fallback controller, so as well make it "bang-bang":

$$\exists \vec{a}, \vec{c} : \forall \vec{x} : \exists \vec{u} : \phi$$
$$\Uparrow$$
$$\exists \vec{a}, \vec{c} : \forall \vec{x} : (\phi(u_{min}) \vee \phi(u_{max}))$$

- Rewrite $V(F(\vec{x}, u, 0^+)) \geq 0$ to eliminate F

Example 2: Controller Synthesis for X

Can be similarly converted to an $\exists\forall\exists$ and if domain of u is finite, then to an $\exists\forall$ problem

Example 3: Verification for X

No inner $\exists u$, hence it is an $\exists\forall$ problem

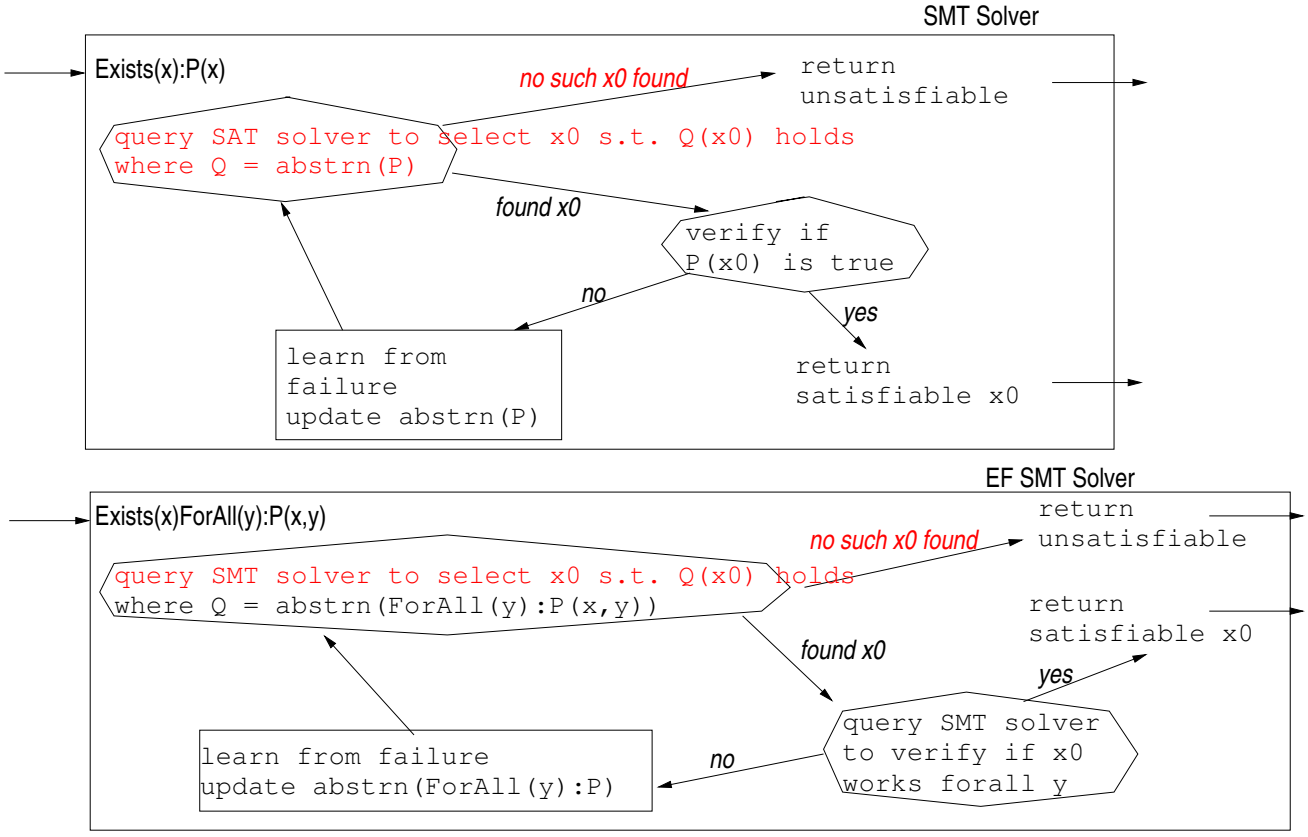
Simplex Architecture

Extract the Monitor and Safe Controller

- EF solver finds values for $\vec{a}$ s.t. $V(\vec{a})(\vec{x}) \geq 0$ is a controlled inductive invariant
- Monitor value of V
- If V is (close to) 0, switch to the safe controller
- Safe controller := at every $\vec{x}$, apply $\vec{u}$ that works

EF Solving

Constraint Solving Using Abstraction Refinement
Guided by Learning via Counter Examples



EF Solving: Simulation Guided Synthesis

$$\exists \vec{a} : \forall \vec{x} : (V \geq 0 \Rightarrow gap \geq 0) \ \wedge \ (V = 0 \Rightarrow dV/dt > 0)$$

1. S := Sample the reachable state space
   - simulation/trace data
   - constraint solving on guard for $\forall$
2. Solve $\exists \vec{a} : \bigwedge_{\vec{x} \in S} : V \geq 0$
   - Find a suitably large safety envelope that contains all points
   - Using the SMT solver Yices – note linear
3. Shrink the safety envelope minimalistically to guarantee safety
   - All but one $\exists$ variables is fixed
   - Constant term d is the only $\exists$ variable remaining
   - Use quantifier elimination

EF Solving: NRA Fragment

ϕ: Nonlinear real arithmetic

$$\exists \vec{a} : \forall \vec{x} : \phi(\vec{a}, \vec{x})$$
$$\Uparrow$$
$$\exists \vec{a}, \vec{p} : \forall \vec{x} : \bigwedge_i q_i(\vec{p}, \vec{a}, \vec{x}) = 0$$
$$\Uparrow$$
$$\exists \vec{a}, \vec{b} : \forall \vec{x} : \bigwedge_i r_i(\vec{a}, \vec{b}, \vec{x}) = 0$$
$$\Uparrow$$
$$\exists \vec{a}, \vec{b} : \bigwedge_i s_i(\vec{a}, \vec{b}) = 0$$

So problem reduces to $\exists \vec{a} : \phi(\vec{a})$

EF-NRA $\mapsto$ E-NRA: Example

$$\exists a, b, u, v, w \forall s, x_0, y_0, y, s_1, y_1 :$$

$$s = ax_0y_0 + bx_0y \wedge s_1 = s + x_0 \wedge y_1 = y - 1 \Rightarrow s_1 = ax_0y_0 + bx_0y_1$$

$$\Uparrow$$

$$s_1 - ax_0y_0 - bx_0y_1 = u(s - ax_0y_0 - bx_0y) + v(s_1 - s - x_0) + wx_0(y_1 - y + 1)$$

$$\Uparrow$$

$$1 = v \wedge -a = -ua \wedge -b = w \wedge 0 = u - v \wedge 0 = -ub - w \wedge 0 = -v + w$$

Nonlinear Fragment

(S = set of polynomial equations; M = model)

Split:
$$\frac{(S \cup \{A\vec{v} = x\vec{v}\}, M)}{(S \cup \{x = \lambda_1\}, M) \mid \ldots \mid (S \cup \{x = \lambda_k\}, M) \mid (S \cup \{\vec{v} = 0\}, M)}$$
where $\lambda_1, \ldots, \lambda_k$ are all the real eigenvalues of A.

Unit_Prop:
$$\frac{(S \cup \{x = c\}, M)}{(S[x \mapsto c], M \cup \{x \mapsto c\})}$$

Success:
$$\frac{(\emptyset, M)}{\text{output model M}}$$

Delete:
$$\frac{(S \cup \{0 = 0\}, M)}{(S, M)}$$

Fail:
$$\frac{(S \cup \{1 = 0\}, M)}{\bot}$$

Example

$$x_1x_2 - x_1x_3 = -2x_2 \qquad x_1x_2 = x_3 \qquad x_2x_3 = 1$$

Split:
$$\frac{(\{x_1x_2 - x_1x_3 + 2x_2 = 0, x_1x_2 - x_3 = 0, x_2x_3 - 1 = 0\}, \emptyset)}{(\{x_1x_2 - x_1x_3 + 2x_2 = 0, x_1x_2 - x_3 = 0, x_2x_3 - 1 = 0, x_1 = 2\}, \emptyset)}$$

Unit_Prop:
$$\frac{}{(\{2x_2 - x_3 = 0, x_2x_3 - 1 = 0\}, \{x_1 \mapsto 2\})}$$

Split:
$$\frac{}{(\{2x_2 - x_3 = 0, x_2x_3 - 1 = 0, x_2 = 1/\sqrt{2}\}, \{x_1 \mapsto 2\})}$$

Unit_Prop:
$$\frac{}{(\{2/\sqrt{2} - x_3 = 0\}, \{x_2 \mapsto 1/\sqrt{2}, x_1 \mapsto 2\})}$$

Unit_Prop:
$$\frac{}{(\emptyset, \{x_3 \mapsto \sqrt{2}, x_2 \mapsto 1/\sqrt{2}, x_1 \mapsto 2\})}$$

Success:
$$\frac{}{\text{output model } x_3 \mapsto \sqrt{2}, x_2 \mapsto 1/\sqrt{2}, x_1 \mapsto 2}$$

Example: Details

$$\begin{pmatrix} x_2 - x_3 \\ x_2 \end{pmatrix} x_1 = \begin{pmatrix} -2x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 & -2 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} x_2 - x_3 \\ x_2 \end{pmatrix}$$

$$\begin{pmatrix} 2 \\ x_3 \end{pmatrix} x_2 = \begin{pmatrix} x_3 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0.5 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ x_3 \end{pmatrix}$$

Features

- Works when Split rule can be applied
- Example class: Nonlinear encoding of SAT problems
- Can be made complete for polynomial equations whose variety is 0-dimensional over the complex
- Appears to work well for nonlinear equations generated from template-based synthesis
- Easily scales to 100s of variables

Future Work

- More competitive implementation of NRA fragment
- Unify
  - learning-based techniques
  - qualitative abstraction
  - time-oblivious and time-aware relational abstractions
  - aligned abstraction refinement
- Synthesis tools
  - EF SMT solver
  - bounded model synthesis
  - infinite bounded model synthesis

Tool

HybridSal (Time-aware) rel. abs.

Sal

sal-inf-bmc/k-ind

Yices

Hybrid system = Composition of (hybrid) automata

$\downarrow$

Infinite state transition system

$\downarrow$

Eϕ