Precise Arithmetic Reasoning using Approximate Solvers

Jaideep Ramachandran, Northeastern University
jaideep@ccs.neu.edu

Goal

To build SMT solvers for arithmetic theories like Real Arithmetic, Floating-Point Arithmetic using **approximate solvers**

Motivation

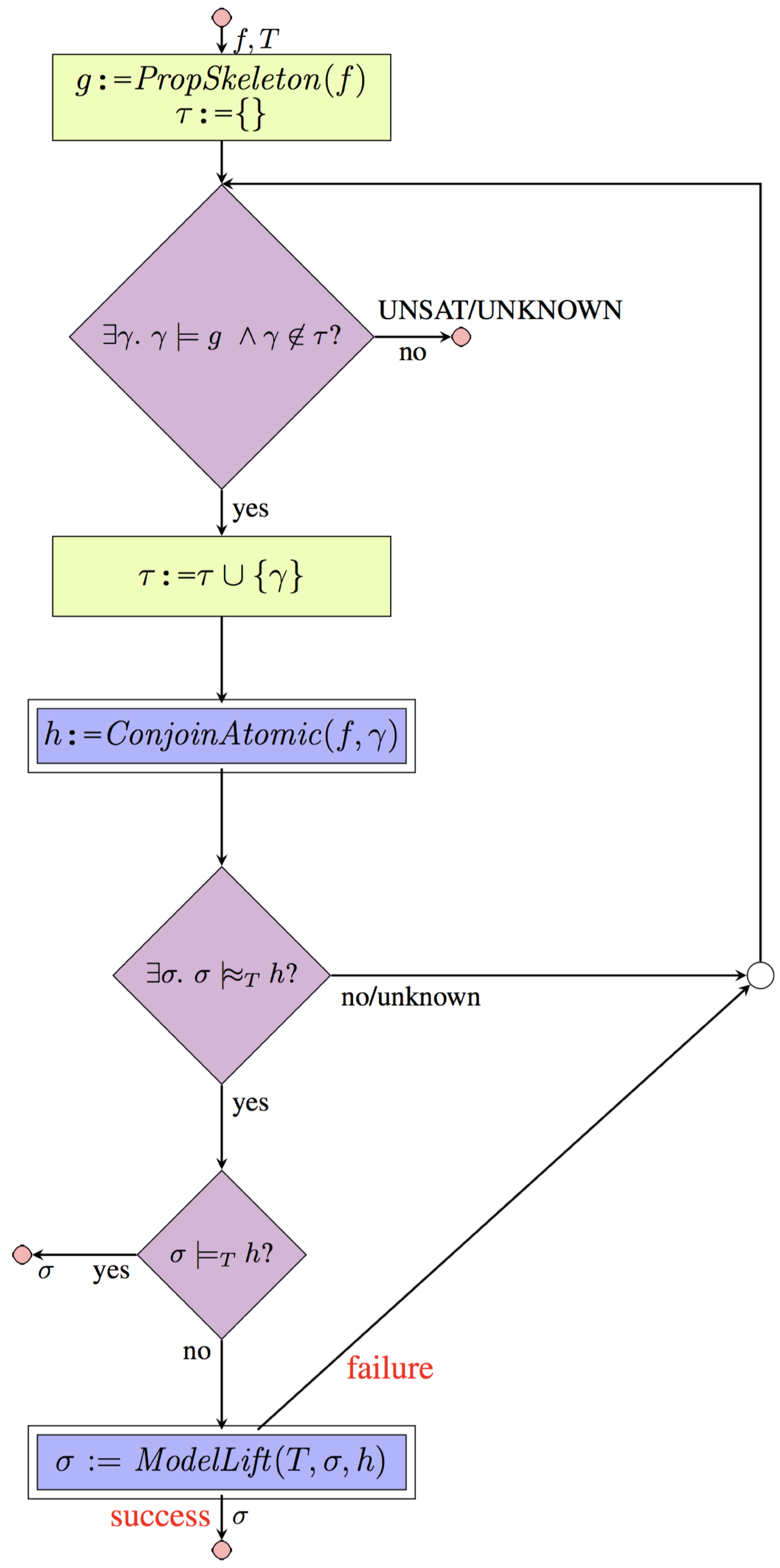
- Existing sound and precise symbolic arithmetic solvers
  - ⋆ efficiency deteriorates for higher-order polynomial constraints
  - ⋆ most can not handle complex arithmetic functions (trigonometric, exponential)
  - ⋆ hard to scale to large formulas
- Solvers based on **numerical techniques** can produce **approximate** solutions **efficiently**

Basic Ideas

- Use a numerical solver to get an approximate assignment efficiently
- *Lift* this approximate assignment to a *precise satisfying assignment* using either symbolic or numerical techniques
- Handle propositional structure of the formula using a SAT solver

Goal is to build sound, precise SMT solvers like Z3 (Microsoft Research), MathSAT (FBK, Italy) but that are more efficient, by using approximate solvers, like dReal (CMU), Matlab (Mathworks, Inc.).

Architecture



Illustration

Input formula f:

$$(x^2 + y < 10) \wedge ((x^4 + x > 84.25) \vee (x < -10)) \wedge (y^2 > 0.25)$$

Propositional skeleton g:

$$p_1 \wedge (p_2 \vee p_3) \wedge p_4$$

where $p_1 := (x^2 + y < 10),\ p_2 := (x^4 + x > 84.25),\ p_3 := (x < -10),\ p_4 := (y^2 > 0.25)$

Solve g using a SAT solver, assignment γ: $\{p_1 := \texttt{true},\ p_2 := \texttt{true},\ p_3 := \texttt{false},\ p_4 := \texttt{true}\}$

Conjoining atomic sub formulas consistent with γ to build h:

$$(x^2 + y < 10) \wedge (x^4 + x > 84.25) \wedge \neg(x < -10) \wedge (y^2 > 0.25)$$

Use a numerical solver to get approximate assignment σ: $\{\mathbf{x = 3.0},\ \mathbf{y = 0.75}\}$

With this σ, we have $\{p_1 := \texttt{true},\ p_2 := \texttt{false},\ p_3 := \texttt{false},\ p_4 := \texttt{true}\}$

g evaluates to `false`, and hence f evaluates to `false`. Inverting p_2, and preserving p_1, p_3, p_4 can make g evaluate to `true`

Perform *model lifting*, to obtain the nearby satisfying assignment σ: $\{x = 3.025,\ y = 0.75\}$, obtained by adding 0.025 to x

g evaluates to `true`, and hence f evaluates to `true`

Open Questions

- **Guaranteeing successful local search**
  - ⋆ under what conditions on the approximate assignment can we guarantee that there lies a precise solution within a small neighborhood?
- **Lifting `UNSAT` proofs**
  - ⋆ can `UNSAT` proofs from approximate solving be lifted to generate an `UNSAT` proof for the original problem?
  - ⋆ during the above process or independent of it, can theory lemmas be learned, like in DPLL(T), to guide the SAT solver in subsequent iterations?