Approximations for Deciding Quantified Floating-Point Constraints

Peter Backeman, Aleksandar Zeljić

Motivation
Efficient model construction of satisfiable constraints is an important challenge with applications in, for example, automatic test-case generation, synthesis of invariants, computation ranking functions, etc. Computing models for quantified constraints remains a very difficult problem. With continuous success of applying approximations in reasoning, we propose extending an approximation framework [1] with support for quantified reasoning.

Using Approximations
- Lift \( \phi \) to an approximated problem \( \hat{\phi} \)
- Solve it in the approximation theory
- Reconstruct the model

Approximation Refinement Framework

Lifting the constraints
- Introduce precision arguments for each quantifier, function and relation symbol.
- Allows control over approximations.
Lifting the formula:

\[ \exists x \forall a \exists y. \phi(x, a, y) \]

results in:

\[ \exists x \forall a \exists y. \phi(x, a, y) \]

Eliminating the Existential Quantifier
- By Skolemization
- Alternating quantifiers introduce functions

Applied to:

\[ \exists x \forall a \exists y. \phi(x, a, y) \]

becomes:

\[ \forall a. \phi(f_a, a, f_j(a)) \]

Approximating the Universal Quantifier
- Precision regulates domain size of the bound variable.
- Quantifier is replaced by a finite conjunction

Let \( D(a) = \{ a_0, a_1, \ldots, a_n \} \). The formula:

\[ \forall a. \phi(f_a, a, f_j(a)) \]

becomes the following:

\[ \phi(f_{a_0}, a_0, f_j(a_0)) \land \ldots \land \phi(f_{a_n}, a_n, f_j(a_n)) \]

Reducing the Domain
- Consider theory-specific solutions
- FPA is parametrized by design
- Domains of FPA are scaled based on precision
- The framework aims to exploit the domain reduction
- For universally bound variables, model reconstruction can depend on the choice of the reduced domain.

<table>
<thead>
<tr>
<th>Domain Reduction</th>
<th>Precise Model Reconstruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_x ) = 0</td>
<td>( f_y ) = 0</td>
</tr>
<tr>
<td>( f_y ) = 1 if ( a \neq 0 )</td>
<td>( f_y ) = 2 if ( a = 0 )</td>
</tr>
<tr>
<td>( f_y ) = 0 if ( a = 0 )</td>
<td>( f_y ) = 1 if ( a = 1 )</td>
</tr>
<tr>
<td>( f_y ) = 0 if ( a = 0 )</td>
<td>( f_y ) = 0 if ( a = 0 )</td>
</tr>
<tr>
<td>CEx = 0 ( a \neq 1 )</td>
<td>CEx = 2 ( a \neq 3 )</td>
</tr>
<tr>
<td>CEx = 1 ( a \neq 2 )</td>
<td>CEx = None</td>
</tr>
</tbody>
</table>

Reducing the range and precision of the floating-point sorts

Future Work and Challenges
- Compact Skolem-function representation
- Finding useful function templates
- Generalizing Skolem functions
- Informed reduction of the quantified domain
- Expanding the reduced domain in a meaningful way
- Balancing domain reduction with function generalization

[1]: Approximations For Model Construction — A. Zeljić, C. M. Wintersteiger, P. Rümmer, IJCAR 2014