Solving Higher-Order Quantified Boolean Satisfiability via Higher-Order Model Checking

Hiroshi Unno[1], Takeshi Tsukada[2], Jie-Hong Roland Jiang[3]
[1]Tohoku University, [2]Chiba University, [3]National Taiwan University

We present the first **HOQBF** solver **HOMCSat** based on **Higher-Order Model Checking**

Higher-Order QBF (HOQBF)

A generalization of **Quantified Boolean Formulas (QBF)** with **higher-order quantifiers**

$\varphi ::= \mathrm{true} \mid \mathrm{false} \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \leftrightarrow \varphi_2$
$\mid \forall x^A.\varphi \mid \exists x^A.\varphi \mid f(x_1, \ldots, x_n)$
$A ::= \mathrm{bool} \mid A_1 \to A_2$

function type

Apply $f^{A_1 \to \ldots \to A_n \to \mathrm{bool}}$ to arguments $x_1^{A_1}, \ldots, x_n^{A_n}$

- capable of succinct encoding of k-EXPTIME problems [Chistikov+ MFCS'22] $2^{2^{\cdot^{\cdot^{2^n}}}}\}k$
- potential applications: memory consistency verification, planning for multi-agent systems, secure system synthesis, and solving quantified bit-vector arithmetic problems
- **QBF** is a fragment where type A is fixed to bool

Ex.1 $\forall x^{\mathrm{bool}}.\exists y^{\mathrm{bool}}.\forall z^{\mathrm{bool}}.(x \vee y \vee z) \wedge (\neg x \vee \neg y \vee \neg z)$

- **Dependency QBF (DQBF)** and **Second-Order QBF (SOQBF)** [Jiang AAAI'23] are also subsumed

Ex.2 $\forall f^{\mathrm{bool}\to\mathrm{bool}}.\exists g^{\mathrm{bool}\to\mathrm{bool}}.\exists z^{\mathrm{bool}}.\big(f(g(z))\big) \leftrightarrow z$

Def.1 HOQBF Satisfiability (HOSAT)
Ask if a closed formula φ is equivalent to true.

Reduction of HOSAT to HOMC

- Thm.1 tells us that there exists a polynomial-time many-one reduction from order-n **HOSAT** to order-$(n+1)$ **HOMC**
- We present a more direct reduction that use **HOMC** for reasoning about H. O. quantifiers:

$P_{\forall x^A.\varphi} \triangleq \mathrm{let\ rec}\ f = \lambda x^A.\mathrm{and}\ P_\varphi$
$\big(\mathrm{if}\ \mathrm{ismax}_A\ x\ \mathrm{then\ true\ else}\ f(\mathrm{succ}_A\ x)\big)\ \mathrm{in}\ f\ \mathrm{zero}_A$

where enumeration structure $(\mathrm{zero}_A, \mathrm{succ}_A, \mathrm{ismax}_A)$ satisfies $A = \{\mathrm{zero}_A, \mathrm{succ}_A\ \mathrm{zero}_A, \ldots, \mathrm{succ}_A^k\ \mathrm{zero}_A\}$ and $\mathrm{ismax}_A(\mathrm{succ}_A^k\ \mathrm{zero}_A)$ for some k

Thm.2
There exists a linear-time **HOSAT** to **HOMC** reduction $\varphi \mapsto P_\varphi$. An order-$n$ **HOQBF** formula is reduced to order $(n+1)$ **HOMC**.

Synthesizing Skolem Functions

- Some applications require **Skolem functions**, i.e., satisfying assignments to existential quantifiers
- The paper discusses extraction from a witness (namely, a typing derivation) returned by HOMC

Ex.4 A Skolem function for **Ex.1**: $Y(x) = \neg x$

Ex.5 Skolem functions for **Ex.2**:
$G(f, x) = \mathrm{true}, \quad Z(f) = f(\mathrm{true})$

Higher-Order Model Checking (HOMC)

A generalization from model checking of **while-programs** to **higher-order functional programs**

$t ::= \mathrm{true} \mid \mathrm{false} \mid x^A \mid \mathrm{if}\ t_1\ \mathrm{then}\ t_2\ \mathrm{else}\ t_3 \mid t_1 t_2$
$\mid \lambda x^A.t \mid \mathrm{let\ rec}\ f^{A_1\to A_2} = t_1\ \mathrm{in}\ t_2$

apply t_1 to t_2

function that takes an argument x and returns t

let-binding (f can occur recursively in t_1)

Ex.3 A program that returns the truth value of **Ex.1**

$\mathrm{let\ rec\ not} = \lambda x^{\mathrm{bool}}.\mathrm{if}\ x\ \mathrm{then\ false\ else\ true\ in}$
$\mathrm{let\ rec\ and} = \lambda x^{\mathrm{bool}}.\lambda y^{\mathrm{bool}}.\mathrm{if}\ x\ \mathrm{then}\ y\ \mathrm{else\ false\ in}$
$\mathrm{let\ rec\ or} = \lambda x^{\mathrm{bool}}.\lambda y^{\mathrm{bool}}.\mathrm{if}\ x\ \mathrm{then\ true\ else}\ y\ \mathrm{in}$
$\mathrm{let\ rec}\ f = \lambda x^{\mathrm{bool}}.\lambda y^{\mathrm{bool}}.\lambda z^{\mathrm{bool}}.$
$\quad \mathrm{and}\ (\mathrm{or}\ (\mathrm{or}\ x\ y)\ z)\ \big(\mathrm{or}\ \big(\mathrm{or}\ (\mathrm{not}\ x)\ (\mathrm{not}\ y)\big)\ (\mathrm{not}\ z)\big)\ \mathrm{in}$
$\mathrm{let\ rec}\ f_z = \lambda x^{\mathrm{bool}}.\lambda y^{\mathrm{bool}}.\mathrm{and}\ (f\ x\ y\ \mathrm{true})\ (f\ x\ y\ \mathrm{false})$
$\mathrm{in\ let\ rec}\ f_{yz} = \lambda x^{\mathrm{bool}}.\mathrm{or}\ (f_z\ \mathrm{true})\ (f_z\ \mathrm{false})\ \mathrm{in}$
$\mathrm{and}\ \big(f_{yz}\ \mathrm{true}\big)\ \big(f_{yz}\ \mathrm{false}\big)$

Def.2 Higher-Order Model Checking (HOMC)
Ask if a well-typed closed term t of type bool is not evaluated to false (i.e. the evaluation diverges or results in true).

Thm.1 Decidability [Ong LICS'06]
The higher-order model checking of order-n term is $(n-1)$-EXPTIME complete.
$\mathrm{order}(\mathrm{bool}) = 1$ // 0 is common in the HOMC lit.
$\mathrm{order}(A_1 \to A_2) = \max(\mathrm{order}(A_1) + 1, \mathrm{order}(A_2))$

- practical model checkers (e.g., **HorSat2**) have been developed and applied to formal verification, despite the high computational complexity

Implementation and Evaluation

HOMCSat implemented using **HorSat2** as **HOMC**

problem	O	V	A	result	time (s)
example1	1	3	2	SAT	0.032
example2	2	3	1	SAT	0.024
aaai23_ex1	2	5	4	SAT	0.437
aaai23_ex2	2	7	4	UNSAT	12.274
sym-asym-b	2	5	0	UNSAT	0.190
sym-asym-bb	2	9	0	T/O	N/A
sym-asym-id	2	7	1	SAT	0.070
SB-theorem	2	16	3	SAT	0.152
left-total	2	3	1	SAT	0.035
right-total	2	3	1	UNSAT	0.035
left-uniq	2	4	0	UNSAT	0.035
right-uniq	2	4	0	SAT	0.032
refl-cl-exist	2	11	2	SAT	3.415
refl-cl-uniq	2	23	2	SAT	124.717
sym-cl-exist	2	13	2	SAT	7.717
sym-cl-uniq	2	27	2	M/O	N/A
tran-cl-exist	2	15	2	SAT	30.286
tran-cl-uniq	2	31	2	M/O	N/A
f_h-neq-g_h	3	3	2	SAT	0.111
cps-arity1	3	5	4	SAT	11.785
cps-arity2	4	6	4	M/O	N/A

O: order
V: number of variables
A: number of quantifier alternations

- Successfully solved HOQBFs that express typical properties of Boolean functions and binary relations
- Revealed limitations and possible future direction of **HOMC**

Conclusion and Future Work

- Bridged **HOMC** and **HOSAT** and will focus on exchanging techniques to advance both fields