Symbolic Automatic Relations and Their Applications to SMT and CHC Solving

Takumi Shimoda Naoki Kobayashi
Ken Sakayori Ryosuke Sato

The University of Tokyo

SMT/CHC on Recursive Data Structures

- Reasoning about recursive data structures is a challenge for many solvers
  (cf. Z3 [Moura+, 2008], CVC4 [Barrett+, 2011], Spacer [Komuravelli+, 2016], HoIce [Champion+, 2020], Eldarica [Hojjat+, 2018])

$$\forall i, x, y, X. sorted(x :: X) \land nth(i, y, X) \Rightarrow x \leq y$$

$$sorted(X) \stackrel{\text{def}}{=} X = [\,] \lor X = x :: [\,] \lor X = x :: y :: X' \land x \leq y \land sorted(y :: X')$$

$$// \text{ the list } X \text{ is sorted in ascending order}$$

$$nth(i, y, X) \stackrel{\text{def}}{=} i = 0 \land X = y :: X' \lor X = x :: X' \land nth(i-1, y, X)$$

$$// \text{ the } i\text{−th element of the list } X \text{ is the integer } x$$

Hard to check the validity

Our Work

- We propose symbolic automatic relations (SARs)
  - Combination of symbolic automata and automatic relations
  - Represent various relations on lists of integers
  - Closed under Boolean operations
- (Incomplete) decision procedure for the satisfiability problem for SAR-formulas
  - Reduction to CHC solving on integers
- Applications to SMT/CHC solving

Outline

- Symbolic Automatic Relations (SARs)
  - Combination of Symbolic Automata and Automatic Relations
  - Satisfiability Problem for SAR-Formulas
- (Incomplete) Decision Procedure for Satisfiability Problem for SAR-Formulas
- Applications
- Evaluation
- Related Work

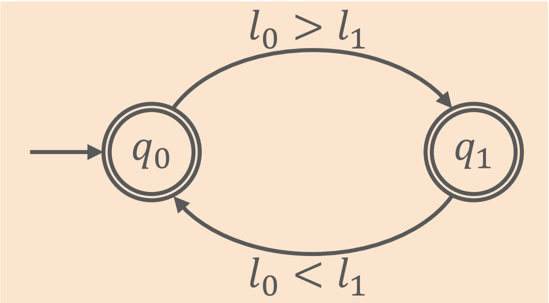
Symbolic Automatic Relations (SARs)

- Combination of symbolic automata [D'Antoni+, 2017] and automatic relations [Blumensath+, 2000]
  - Read inputs in a synchronous manner
  - Have predicates over integers as transition labels
  - Closed under Boolean operations

e.g.) Given the inputs $[10,0,10]$ and $[5,5,5]$,
R_0 reads $[(10,5),(0,5),(10,5)]$

$R_0(X, Y)$
$\Leftrightarrow X[i] > Y[i]$ if i is even number and
$X[i] < Y[i]$ if i is odd number

SAR $R_0(L_0, L_1)$

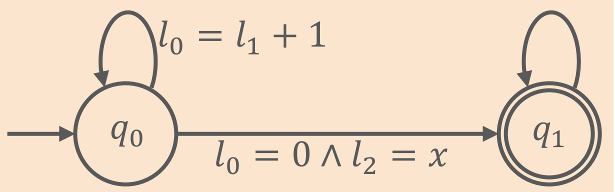


SAR-Formulas

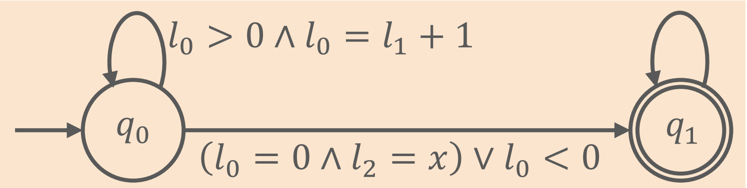
$$\text{SAR-Formula } \psi ::= \quad \varphi \text{ s.t. } \varphi \text{ and } \neg\varphi \text{ are of the form } \exists\tilde{X}\exists\tilde{x}.R(\tilde{T},\tilde{t})$$
$$\mid \text{primitive predicate on integers}$$
$$\mid \bot \mid \top \mid \psi \vee \psi \mid \psi \wedge \psi \mid \neg\psi$$

$nth(i, x, X)$, which means "the i-th element of X is x", can be defined as below and is an SAR-formula

$R_1(L_0, L_1, L_2, x)$



$R_2(L_0, L_1, L_2, x)$



$$nth(i, x, X) \Leftrightarrow \exists Y.R_1(i :: Y, Y, X, x)$$
$$\neg nth(i, x, X) \Leftrightarrow \exists Y.R_2(i :: Y, Y, X, x)$$

Satisfiability Problem for SAR-Formulas

Given an SAR-formula $\exists\tilde{X}\exists\tilde{x}.\,R(\tilde{T},\tilde{t})$, is $\exists\tilde{X}\exists\tilde{x}.\,R(\tilde{T},\tilde{t})$ satisfiable?

Undecidable in general

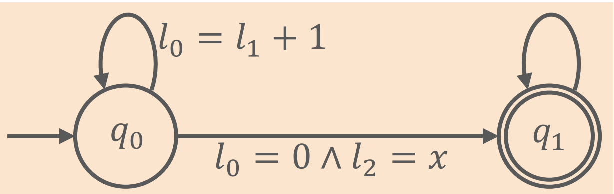
- Various properties on lists can be written as SAR-formulas

$$nth(2, 5, X)$$

The 2nd element of X is 5

$$\Leftrightarrow \exists Y. R_1(2 :: Y, Y, X, 5)$$

$R_1(L_0, L_1, L_2, x)$



Outline

- Symbolic Automatic Relations
- (Incomplete) Decision Procedure for Satisfiability Problem for SAR-Formulas
  - Reduction to CHC solving on Integers
- Applications
- Evaluation
- Related Work

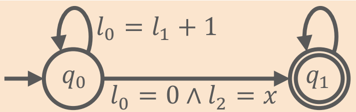
Reduction to CHCs on Integers

Satisfiability Problem for SAR-Formulas

Given an SAR-formula φ,
is φ satisfiable?

$$\varphi \overset{\text{def}}{=} \exists Y. R_1(2 :: Y, Y, X, 5)$$

$R_1(L_0, L_1, L_2, x)$



sound and complete reduction
s.t. φ is satisfiable $\Leftrightarrow$ Π_φ is unsatisfiable

CHC solving on Integers

Constrained Horn Clause

Given a set of CHCs Π_φ,
is Π_φ satisfiable?

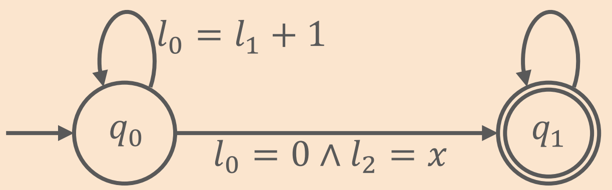
$$\underline{q_0}(i', j', k') \Leftarrow i' = 2$$
$$\underline{q_0}(i', j', k') \Leftarrow \underline{q_0}(i, j, k) \land i = j + 1 \land i' = j \land \neg\varphi_{end}$$
$$\vdots$$

Solvable in practice for many problems by existing CHC solvers

Intuition of Reduction

$$\varphi \stackrel{\text{def}}{=} \exists Y. R_1(2 :: Y, Y, X, 5)$$

$R_1(L_0, L_1, L_2, x)$



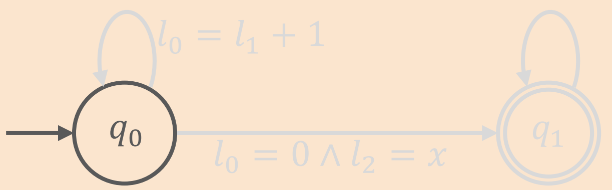
Intuition :
$\underline{q}(i, j, k) \Leftrightarrow$ "for some X and Y, given $(2 :: Y, Y, X)$ as inputs, the SAR R_1 visits q reading (i, j, k) as next elements"

CHCs Π_φ

$$\underline{q_0}(i', j', k') \Leftarrow i' = 2$$
$$\underline{q_0}(i', j', k') \Leftarrow \underline{q_0}(i, j, k) \wedge i = j + 1 \wedge i' = j \wedge \neg\varphi_{end}$$
$$\underline{q_1}(i', j', k') \Leftarrow \underline{q_0}(i, j, k) \wedge i = 0 \wedge k = 5 \wedge i' = j \wedge \neg\varphi_{end}$$
$$\underline{q_1}(i', j', k') \Leftarrow \underline{q_1}(i, j, k) \wedge i' = j \wedge \neg\varphi_{end}$$
$$\bot \Leftarrow \underline{q_1}(i, j, k) \wedge \varphi_{end}$$

Clauses from Initial States

$R_1(L_0, L_1, L_2, x)$



$$\varphi \stackrel{\text{def}}{=} \exists Y. R_1(2 :: Y, Y, X, 5)$$

R_1 is initially at state q_0 with the first element being 2

CHCs Π_φ

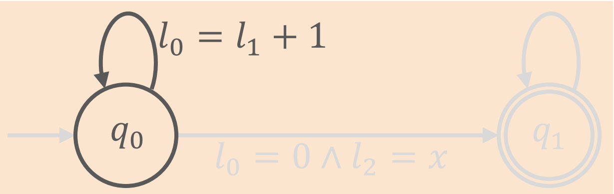
$$\underline{q_0}(i', j', k') \Leftarrow i' = 2$$
$$\underline{q_0}(i', j', k') \Leftarrow \underline{q_0}(i, j, k) \land i = j + 1 \land i' = j \land \neg\varphi_{end}$$
$$\underline{q_1}(i', j', k') \Leftarrow \underline{q_0}(i, j, k) \land i = 0 \land k = 5 \land i' = j \land \neg\varphi_{end}$$
$$\underline{q_1}(i', j', k') \Leftarrow \underline{q_1}(i, j, k) \land i' = j \land \neg\varphi_{end}$$
$$\bot \Leftarrow \underline{q_1}(i, j, k) \land \varphi_{end}$$

$\underline{q}(i, j, k) \Leftrightarrow$
"for some X and Y,
given $(2 :: Y, Y, X)$ as inputs,
the SAR R_1 visits q
reading (i, j, k) as next elements"

Clauses from Transitions

$$\varphi \stackrel{\text{def}}{=} \exists Y. R_1(2 :: Y, Y, X, 5)$$

$R_1(L_0, L_1, L_2, x)$



the transition label

the relation between inputs $2 :: Y$ and Y

There is still an input to read

CHCs Π_φ

$\underline{q_0}(i', j', k') \Leftarrow i' = 2$

$\underline{q_0}(i', j', k') \Leftarrow \underline{q_0}(i, j, k) \land i = j + 1 \land i' = j \land \neg\varphi_{end}$

$\underline{q_1}(i', j', k') \Leftarrow \underline{q_0}(i, j, k) \land i = 0 \land k = 5 \land i' = j \land \neg\varphi_{end}$

$\underline{q_1}(i', j', k') \Leftarrow \underline{q_1}(i, j, k) \land i' = j \land \neg\varphi_{end}$

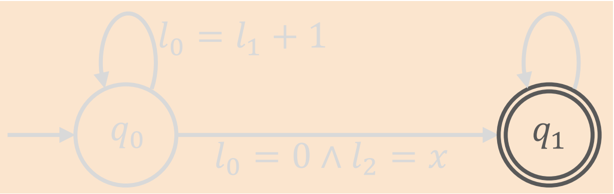
$\bot \Leftarrow \underline{q_1}(i, j, k) \land \varphi_{end}$

$\underline{q}(i, j, k) \Leftrightarrow$
"for some X and Y,
given $(2 :: Y, Y, X)$ as inputs,
the SAR R_1 visits q
reading (i, j, k) as next elements"

Clauses from Final States

$$\varphi \stackrel{\text{def}}{=} \exists Y. R_1(2 :: Y, Y, X, 5)$$

$R_1(L_0, L_1, L_2, x)$



$\bot$ is derived if R_1 finishes reading all the inputs at final state q_1

CHCs Π_φ

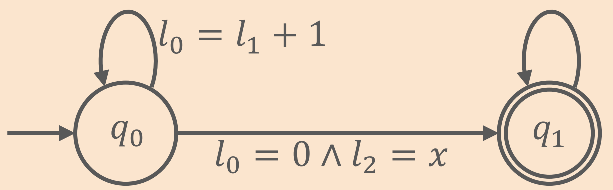
$$\underline{q_0}(i', j', k') \Leftarrow i' = 2$$
$$\underline{q_0}(i', j', k') \Leftarrow \underline{q_0}(i, j, k) \land i = j + 1 \land i' = j \land \neg\varphi_{end}$$
$$\underline{q_1}(i', j', k') \Leftarrow \underline{q_0}(i, j, k) \land i = 0 \land k = 5 \land i' = j \land \neg\varphi_{end}$$
$$\underline{q_1}(i', j', k') \Leftarrow \underline{q_1}(i, j, k) \land i' = j \land \neg\varphi_{end}$$
$$\bot \Leftarrow \underline{q_1}(i, j, k) \land \varphi_{end}$$

$\underline{q}(i, j, k) \Leftrightarrow$
"for some X and Y,
given $(2 :: Y, Y, X)$ as inputs,
the SAR R_1 visits q
reading (i, j, k) as next elements"

Correctness

$$\varphi \stackrel{\text{def}}{=} \exists Y. R_1(2 :: Y, Y, X, 5)$$

$R_1(L_0, L_1, L_2, x)$



accepting run of R_1 $\Leftrightarrow$ **the derivation of contradiction of Π_φ**

Given $X = [7,6,5], Y = [1,0]$,
R_1 accepts $(2 :: Y, Y, X)$ along the path

$$q_0 \xrightarrow{(2,1,7)} q_0 \xrightarrow{(1,0,6)} q_0 \xrightarrow{(0,\_,5)} q_1$$

$q_0(2,1,7) \Leftarrow$ clause(1)
$q_0(1,0,6) \Leftarrow q_0(2,1,7)$ and clause(2)
$q_0(0,\_,5) \Leftarrow q_0(1,0,6)$ and clause(2)
$q_1(\_,\_,\_) \Leftarrow q_0(0,\_,5)$ and clause(3)
$\bot \Leftarrow q_1(\_,\_,\_)$ and clause(5)

CHCs Π_φ

$$q_0(i', j', k') \Leftarrow i' = 2$$
$$q_0(i', j', k') \Leftarrow q_0(i, j, k) \wedge i = j + 1 \wedge i' = j \wedge \neg\varphi_{end}$$
$$q_1(i', j', k') \Leftarrow q_0(i, j, k) \wedge i = 0 \wedge k = 5 \wedge i' = j \wedge \neg\varphi_{end}$$
$$q_1(i', j', k') \Leftarrow q_1(i, j, k) \wedge i' = j \wedge \neg\varphi_{end}$$
$$\bot \Leftarrow q_1(i, j, k) \wedge \varphi_{end}$$

$q(i, j, k) \Leftrightarrow$
"for some X and Y,
given $(2 :: Y, Y, X)$ as inputs,
the SAR R_1 visits q
reading (i, j, k) as next elements"

Outline

- Symbolic Automatic Relations
- (Incomplete) Decision Procedure for Satisfiability Problem for SAR-Formulas
- Applications
- Evaluation
- Related Work

Applications

- SMT solving on recursive data structures
  - Applicable to checking the validity/satisfiability of quantifier-free formulas consisting of predicates belonging to SAR-formulas

Formulas s.t. itself and its negation are of the form $\exists \tilde{X} \exists \tilde{x}. R(\tilde{T}, \tilde{t})$

- CHC solving
  - Reduction from CHC solving on data structures to CHC solving on integers
  - Applicable to teacher part of ICE-based CHC solver

Outline

- Symbolic Automatic Relations
- (Incomplete) Decision Procedure for Satisfiability Problem for SAR-Formulas
- Application to ICE-based CHC solving
- Evaluation
- Related Work

Implementation

- Implemented a satisfiability solver for SAR-formulas
  - Input:
    - definitions of predicates as SAR-formulas
    - quantifier-free formulas

e.g.)

$$nth(i, x, X) \Rightarrow nth(i+1, x, y :: X) \quad \text{// formula whose validity to be checked}$$

$$nth(i, x, X) \stackrel{\text{def}}{=} \exists Y. R_1(i :: Y, Y, X, x) \quad \text{// definitions of predicate nth}$$

$$\neg nth(i, x, X) \stackrel{\text{def}}{=} \exists Y. R_2(i :: Y, Y, X, x) \quad \text{// and its negation}$$

$$R_1 = \cdots, R_2 = \cdots \quad \text{// definitions of SARs}$$

- Backend CHC solvers on integers
  - Spacer [Komuravelli+, 2016]
  - HoIce [Champion+, 2020]
  - Eldarica [Hojjat+, 2018]

Evaluation

Benchmark (#Instances)	**IsaPlanner (15)**	**SAR_SMT (60)**	**CHC (12)**	**All (87)**
Ours-Spacer	8	43	8	59
Ours-HoIce	14	55	11	80
Ours-Eldarica	14	59	12	85
Z3 [Moura+, 2008] (rec)	5	32	1	38
Z3 (assert)	7	20	3	30
CVC4 [Barrett+, 2011] (rec)	5	32	3	40
CVC4 (assert)	6	19	3	28

- Count the number of instances solved within 60 seconds
- Our tool solved more instances for three benchmarks

Outline

- Symbolic Automatic Relations
- (Incomplete) Decision Procedure for Satisfiability Problem for SAR-Formulas
- Applications
- Evaluation
- Related Work

Related Work

- Decidable theories on arrays [Bradley+,2006] or inductive data types [Barrett+,2007]
  - Decidable fragments of these theories are limited
- CHC Solver based on Tree Automatic Relations [Haudebourg,2020]
  - Does not deal with data structures of elements from an infinite set
- Fold/unfold transformation [Angelis+, 2020]
  - Removes algebraic data types from CHCs
  - Seems to be related to SARs in some way

Conclusion

- Symbolic automatic relations
  - Combination of symbolic automata and automatic relations
  - Useful for reasoning about recursive data structures
- (Incomplete) decision procedure for the satisfiability problem for SAR-Formulas
  - Reduction to CHC on integers
- Future work
  - Learner’s algorithm for ICE-based CHC solver