COMPSPEN: A Solver for Entailment of Separation Logic Formulae with Compositional Inductive Definitions

Chong Gao, Zhilin Wu

State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences

1 Solver name and team

Solver name: COMPSPEN

Team:

– Taolue Chen, Department of Computer Science and Information Systems, Birkbeck, University of London, UK,
– Chong Gao, State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, China,
– Zhilin Wu, State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, China.

2 SL-fragment and theoretical aspects

COMPSPEN is based on our IJCAR 2016 paper ([1]). In the following, we will briefly recall the fragment of separation logic and the theoretical aspects (decision procedures) in [2]. More details can be found in [2].

2.1 Fragment of separation logic

The fragment of separation logic that COMPSPEN can deal with is called the linearly compositional fragment of separation logic with inductive definitions (abbreviated as $SLID_{LC}[\mathcal{P}]$, where $\mathcal{P}$ is a set of inductive definitions), where both shape properties, e.g., singly and doubly linked lists, linked lists with tail pointers, and data constraints, e.g., sortedness property and size constraints, can be expressed. The basic idea of $SLID_{LC}[\mathcal{P}]$ is to focus on the compositional predicates introduced in [1], while restricting to linear shapes, and data constraints in the form of difference bound relations (which are sufficient to express sortedness properties and size constraints).

$SLID_{LC}[\mathcal{P}]$ formulae comprise three types of formulae: pure formulae Π, data formulae Δ, and spatial formulae Σ, which are defined by the following rules, where E, F, X are location variables, x, y are integer variables, c is an integer constant, f is a location field, d is a data field, and $\mathtt{o} \in \{=, \leq, \geq\}$,

$$\begin{array}{ll} \Pi ::= E = F \mid E \neq F \mid \Pi \wedge \Pi & \text{(pure formulae)} \\ \Delta ::= \mathtt{true} \mid x \ \mathtt{o} \ c \mid x \ \mathtt{o} \ y + c \mid \Delta \wedge \Delta & \text{(data formulae)} \\ \Sigma ::= \mathtt{emp} \mid E \mapsto \rho \mid P(E, \boldsymbol{\alpha}; F, \boldsymbol{\beta}; \boldsymbol{\xi}) \mid \Sigma * \Sigma & \text{(spatial formulae)} \\ \rho ::= (f, X) \mid (d, x) \mid \rho, \rho. & \end{array}$$

We require that for each inductive predicate P in $SLID_{LC}[\mathcal{P}]$, its inductive definition is given by the following two rules,

– base rule $R_0 : P(E, \boldsymbol{\alpha}; F, \boldsymbol{\beta}; \boldsymbol{\xi}) ::= E = F \wedge \boldsymbol{\alpha} = \boldsymbol{\beta} \wedge \mathtt{emp}$,
– inductive rule $R_1 : P(E, \boldsymbol{\alpha}; F, \boldsymbol{\beta}; \boldsymbol{\xi}) ::= \exists \boldsymbol{X} \exists \boldsymbol{x}.\ \Delta \wedge E \mapsto \rho * P(Y, \boldsymbol{\gamma}; F, \boldsymbol{\beta}; \boldsymbol{\xi})$,

satisfying some additional syntactical constraints, whose details are omitted here.

2.2 Theoretical aspects

In this section, we will briefly describe the decision procedures for the satisfiability and entailment problem of $SLID_{LC}[\mathcal{P}]$.

Satisfiability For the satisfiability problem, from each $SLID_{LC}$ formula φ, we define an abstraction of φ, i.e.,$Abs(\varphi)$, where Boolean variables are introduced to encode the spatial part of φ, together with quantifier-free Presburger

formulae to represent the transitive closure of the data constraint in the inductive definitions. The satisfiability of φ is then reduced to the satisfiability of $Abs(\varphi)$, which can be solved by the start-of-the-art SMT solvers (e.g., Z3 [3]), with an NP upper-bound.

Entailment For the entailment problem $\varphi \vDash \psi$, where φ, ψ are two $SLID_{LC}[\mathcal{P}]$ formulae, we assume that all the variables in ψ also occur in φ, both φ and ψ are satisfiable, and the set of (location and data) fields used in φ are the same as that of ψ. On a high level, we construct graph representations $\mathcal{G}_\varphi$ and $\mathcal{G}_\psi$ of φ and ψ respectively and reduce the entailment problem to (a variant of) the graph homomorphism problem from $\mathcal{G}_\psi$ to $\mathcal{G}_\varphi$. More specifically, we simplify the structure of $\mathcal{G}_\varphi$ by allocating plans, and the entailment holds iff for each allocating plan AP of $\mathcal{G}_\varphi$, there is a homomorphism from $\mathcal{G}_\psi$ to $\mathcal{G}_{\varphi,AP}$, the simplification of $\mathcal{G}_\varphi$ by AP.

We also extend our decision procedures to deal with acyclic singly linked lists, which are specified by pseudo-compositional inductive predicates [1], instead of compositional ones.

3 Solver architecture

The architecture of COMPSPEN is illustrated in Figure 1. COMPSPEN is implemented in C++, uses the libraries Z3 and boost, implements an input parser that accepts the text files in the SL-COMP 2014 format, and uses a pre-processor SL-COMPSEN to translate the benchmarks in SL-COMP 2018 format to the SL-COMP 2014 format.

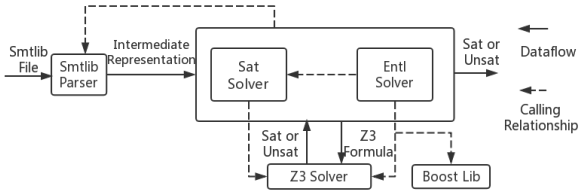

Fig. 1. Framework

4 Strengths and weaknesses

Strengths. COMPSPEN can solve the satisfiability and entailment problems for separation logic with inductive predicates on linear data structures, no matter whether there exist data constraints or not. Specifically, COMPSPEN can solve all the problems in the divisions qf_shls_sat and qf_shls_entl, and those for linear data structures in the divisions qf_shidlia_sat and qf_shidlia_entl.

Weaknesses. COMPSPEN is unable to deal with non-linear data structures at present. Moreover, COMPSPEN is unable to deal with the inductive predicates even for linear data structures, if they are neither compositional nor pseudo-compositional, e.g. the list segment unfolded from the right-hand side.

5 Contact information

COMPSPEN is open source, and available at https://github.com/gaochong1111/compspen. For any question, please contact Chong Gao (gaochong@ios.ac.cn) and Zhilin Wu (wuzl@ios.ac.cn), at State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences.

References

1. X. Gu, T. Chen, and Z. Wu. A complete decision procedure for linearly compositional separation logic with data constraints. In IJCAR, pages 532549,2016.
2. C. Enea, M. Sighireanu, and Z. Wu. On automated lemma generation for separation logic with inductive definitions. In ATVA, pages 8096, 2015.
3. Z3. http://rise4fun.com/z3.