Separation Logic Competition SL-COMP 2018

Presented by Mihaela Sighireanu joint work with Radu Iosif, Andrew J. Reynolds, Cristina Serban, Chong Gao, Jens Katelaan, Benedict Lee, Le Quang Loc, Adam Rogalewicz, Ta Quang Trung, and others

Workshop ADSL 2018, July 13th

Outline

Static Results

Dynamic Results

Conclusion and Future

SL-COMP

Started in 2014 as a satellite event of SMT-COMP 2014:

- Objectives:
  - promote the implementation effort on solvers for SL
  - share a benchmark of interesting problems
  - compare techniques
- Results:
  - 6 solvers
  - 678 problems, 25% sat and 75% entailment
  - common input format based on SMT-LIB 2.0
  - 5 divisions of (mainly) quantifier free formulas in the symbolic heap fragment with specific (e.g., lseg) or general inductive definitions

The second edition, SL-COMP 2018

Same objectives, new results:

- new cleaner input format, aligned with SMT-LIB 2.6
- +618 (~+100%) new benchmarks, fixes some old ones
- +6 divisions, better naming
- +4 (initially +6) solvers
- gain in visibility

Input Format

Work done by Adrew J. Reynolds, Cristina Serban and Radu Iosif
Start with the SMT-lib 2.6 (2017) including

- `datatypes` used to define types of heap cells
    - locations are abstract sorts
- `funs-rec` used for inductive heap predicates

```
(declare-sort RefCell 0)
(define-datatype Cell ((cons (data Int) (next RefCell)))
```

Input Format

Work done by Adrew J. Reynolds, Cristina Serban and Radu Iosif
Start with the SMT-lib 2.6 (2017) including

- `datatypes` used to define types of heap cells
  - locations are abstract sorts
- `funs-rec` used for inductive heap predicates

```
(declare-sort RefCell 0)
(define-datatype Cell ((cons (data Int) (next RefCell)))
```

Extend with a new command for heap typing

```
(declare-heap (RefCell Cell) (RefTree Tree))
```

Input Format (cont.)

Theory `SepLogTyped` has no predefined sorts but new operators:

```
:funs  ((emp Bool)
        (sep Bool Bool Bool :left-assoc)
        (wand Bool Bool Bool :right-assoc)
        (par (L D) (pto L D Bool))
        (par (L) (nil L)) )
```

Logics are defined as usual in SMT-lib.
Free variables are declared as constants (SMT-lib style)
Problems are either:

- `sat`, input is a set of assertions
- `entl`, input is two assertions, φ followed by $\neg\psi$, to check $\varphi \models \psi$

Division Naming

Division = a logic + a problem

- 8 divisions in SL-COMP'18 (+5 wrt 2014)

Naming follows rules of SMT-lib

- prefix *QF_* for quantifier free (SMT-lib)
- *LIA* for linear arithmetics (SMT-lib)
- *SH* for symbolic heaps
- *BSL* for boolean combination
- *ID* for general well-formed (SMT-lib) inductive definitions
- *LID* for linear ID (lists, nested lists, skip lists)

Example: `qf_shidlia_entl`

Collected Problems by Division

Division	#problems
qf_bsl_sat	46
qf_bsllia_sat	24
qf_shid_entl	311
qf_shid_sat	99
qf_shidlia_entl	75
qf_shidlia_sat	33
qf_shlid_entl	59
qf_shls_entl	296
qf_shls_sat	110
shid_entl	73
shidlia_entl	181

Participants

Old fellows (6):

- Asterix: A. Rybalchenko (MSR), J.A. Navarro Perez (Google)
- CYCLIST & SLSAT: N. Gorogiannis (Middlesex U.)
- SLEEK: B. Lee, C. Wei Ngan (NUS)
- SLIDE: R. Iosif (Verimag); A. Rogalewicz (TU Brno)
- SPEN: C. Enea, M.S. (UPD); T. Vojnar, O. Lengal (TU Brno)

New fellows (4 + *3*):

- *ComSPEN*: C. Gao, Z. Wu (Acad. China)
- CVC4: A. J. Reynolds (U. Iowa)
- Harrsh: J. Katelaan (TU Vienna)
- *Inductor*: R. Iosif, C. Serban (Verimag)
- S2S: L. Le Quang (Teesside U.)
- *Sloth*: J. Katelaan (TU Vienna)
- Songbird: T. Ta Quang, C. Wei Ngan (NUS)

Participants by Underlying Technique

- *SMT solving*: Asterix, CVC4, SLSAT
- *Small model and SMT*: ComSPEN, Sloth
- *Automata*: Harrsh, SLIDE, SPEN
- *Proofs*: SLEEK, SPEN
- *Cyclic proofs*: CYCLIST, S2S, Songbird

Collected Set of Benchmarks

Division	size	Solver
qf_bsl_sat	46	CVC4
qf_bsllia_sat	24	CVC4
qf_shid_entl	312	CYCLIST, S2S, SLEEK, SLIDE, Songbird, SP
qf_shid_sat	99	CYCLIST, Harrsh, S2S, SLEEK
qf_shidlia_entl	61	ComSPEN, S2S
qf_shidlia_sat	33	ComSPEN, S2S
qf_shlid_entl	60	ComSPEN, SPEN
qf_shls_entl	296	Asterix, S2S, SPEN
qf_shls_sat	110	Asterix
shid_entl	73	SLEEK, Songbird
shidlia_entl	181	Songbird

. . . and in a diagram

Execution on StarExec

NB: rules are not clearly stated, very flexible, on demand

Yet,

- solver binary running on StarExec
  - pull out 2 solvers!
- by default: 600 sec of timeout and 4 GB of memory
  - initially 120 sec and 1 GB, request to increase
  - timeout increased to 2400 then 3600 if ressourced out
- 3 or 4 rounds, depending on
  - availability of the final version of the solver
  - number of ressourced out problems

Division *qf_shls_entl*

- Origin: *sll0a_entl* of SL-COMP'14
- 7 solvers, 296 problems
- mainly run with 600 sec and 4GB
- too much wrong results
  - a problem in pre-processors?
  - inconsistency of solvers?

Entry division, includes problems that reveal solver's corner cases.

Division *qf_shls_sat*

- Origin: *sll0a_sat* of SL-COMP'14
- 7 solvers, 110 problems
- mainly run with 600 sec and 4GB
- PTIME algorithm, not for proof techniques

Asterix is still the best!

Division *qf_shid_entl*

- Origin: *UDB_entl* of SL-COMP'14
- 6 solvers, 312 problems
- interesting runs when timeout is >= 2400
  - yet, some problems are easy (see SPEN-TA)
  - a lot of wrong results!

Definitively a difficult division!

Division *qf_shlid_entl*

- Origin: *FDB_entl* of SL-COMP'14
  - ID with linear form, have a PTIME algorithm
- 6 solvers, 60 problems
- fragment not clearly defined, so many wrong results

Put on show S2S!

but

Work to do on the benchmark!

Division *shid_entl*

- Origin: *UDB_entl* of SL-COMP'14
    - incorrectly classified `QF`
    - mainly quantifiers in consequent
- 5 solvers, 73 problems
- Execution timeouts set to 2400 sec at least

Put on show Songbird!

Division qf_shid_sat

- Origin: UDB_sat of SL-COMP'14
- 7 solvers, 99 problems
- Impressive differences in execution times
- Some problems to be fixed with 9 problems or in the pre-processors

Put on show SLSAT!

Divisions qf_bsl_sat and qf_bsllia_sat

- New, problems mainly provided by CVC4
- 1(+/1/) solver
- Question: what to do with magic wand?

Need for solvers to challenge CVC4!

Division $qf\_shidlia\_entl$

- New, problems from proof based solvers
- 3 solvers, 61 problems
- Execution times differ very much

Put on show Songbird!

Division *qf_shidlia_sat*

- New, problems from proof based solvers
- 3 solvers, 33 problems

Put on show S2S!

Division *shidlia_entl*

- New, problems from proof based solvers
- 3 solvers, 181 problems
- Execution timeouts shall be >= 2400 sec

Put on show Songbird!

Conclusion and Future

Successfull edition:

- new benchmark for interesting logics
  - extension with arithmetics and boolean combination
- clean input and tools supporting it
  - C++ and Ocaml parser and checkers (typing, logic)
- new solvers, old ones are still competitive

Future:

- clean existing benchmark based on analysers
- fix problems of running on StarExec for some solvers
- fix inconsistency in solvers and pre-processors
- Toolympics at ETAPS 2019:
  - competition presentation: accepted
  - official publication in ETAPS proceedings?
  - re-run for April 2019??