Integration of SMT-LIB Support into Maple

Stephen Forrest
29 July 2017
SC² Workshop, Kaiserslautern

Maplesoft
Mathematics • Modeling • Simulation
A Cybernet Group Company

SMT-CAS Integration

- Some SMT solvers presently incorporate computer algebra techniques in their theory solvers.
  - Examples: veriT [3], SMT-RAT [4]

- Alternate avenue of co-operation:
  CAS dispatching instances of SMT problems which it encounters in the course of symbolic computation to a SAT/SMT solver
  - Examples: many, e.g. see [5]

© 2016 Maplesoft, a division of Waterloo Maple Inc.

Logic in Maple

- Maple's built-in logic is three-valued (`true`, `false`, `FAIL`)
- Also has a package (Logic) for two-valued propositional logic:
  - **BooleanSimplify** – return an equivalent but more compact formula if one exists
  - **Dual** – construct the logical dual
  - **Equivalent** – check equivalence of two formulae
  - **Normalize** – convert to a normal form (CNF or DNF)
  - **Satisfy / Satisfiable** – generate a satisfying truth assignment
  - **Tautology** – check if formula holds for all truth assignments
- Since Maple 2016, **Satisfy** and related commands use MiniSat.
- Package does not support existential or universal quantification

© 2016 Maplesoft, a division of Waterloo Maple Inc.

Maple Assume Facility (1)

- Maple's assume facility allows checking of Boolean propositions subject to declared assumptions.
- Assumptions consist of Boolean predicates or *properties* attached to specified symbols.
- Two main commands exist:
  - `is`: equivalent of ∀
  - `coulditbe`: equivalent of ∃

General form (fV stands for "free variables"):

- `is(p) assuming q` → ∀ fV(q,p) (q => p)
- `coulditbe(p) assuming q` → ∃ fV(q,p) (q => p)

© 2016 Maplesoft, a division of Waterloo Maple Inc.

Maple Assume Facility (2)

- Maple's assume facility allows checking of propositions subject to assumptions.

$$> \cos(n\pi) \text{ assuming } n :: integer;$$

$$(-1)^{n\sim}$$

$$> coulditbe\left(\frac{i}{3}, integer\right) \text{ assuming } \frac{i}{2} :: integer$$

$$true$$

$$> is\left(-\frac{(4a^2+1)^2}{8a^4} < 0 \textbf{ and } -\frac{(4a^2+1)^4}{32a^8} < 0 \textbf{ and } -\frac{(4a^2+1)^4}{64a^8} < 0\right) \text{ assuming } a > 0;$$

$$true$$

- Queries often include inequalities but in practice rarely have significant Boolean structure

© 2016 Maplesoft, a division of Waterloo Maple Inc.

Maple Assume Facility (3)

- Symbolic evaluation in Maple makes regular use of the assume facility and its associated queries
- *Example*: evaluation of symbolic product

```
product(f(n),n=a..b);
```

$$\prod_{n=a}^{b} f(n)$$

- Implementation includes a case which computes the roots of ***f(n)*** and checks whether any of them are integers in the range ***[a, b]*** using the `couldbe` command.

© 2016 Maplesoft, a division of Waterloo Maple Inc.

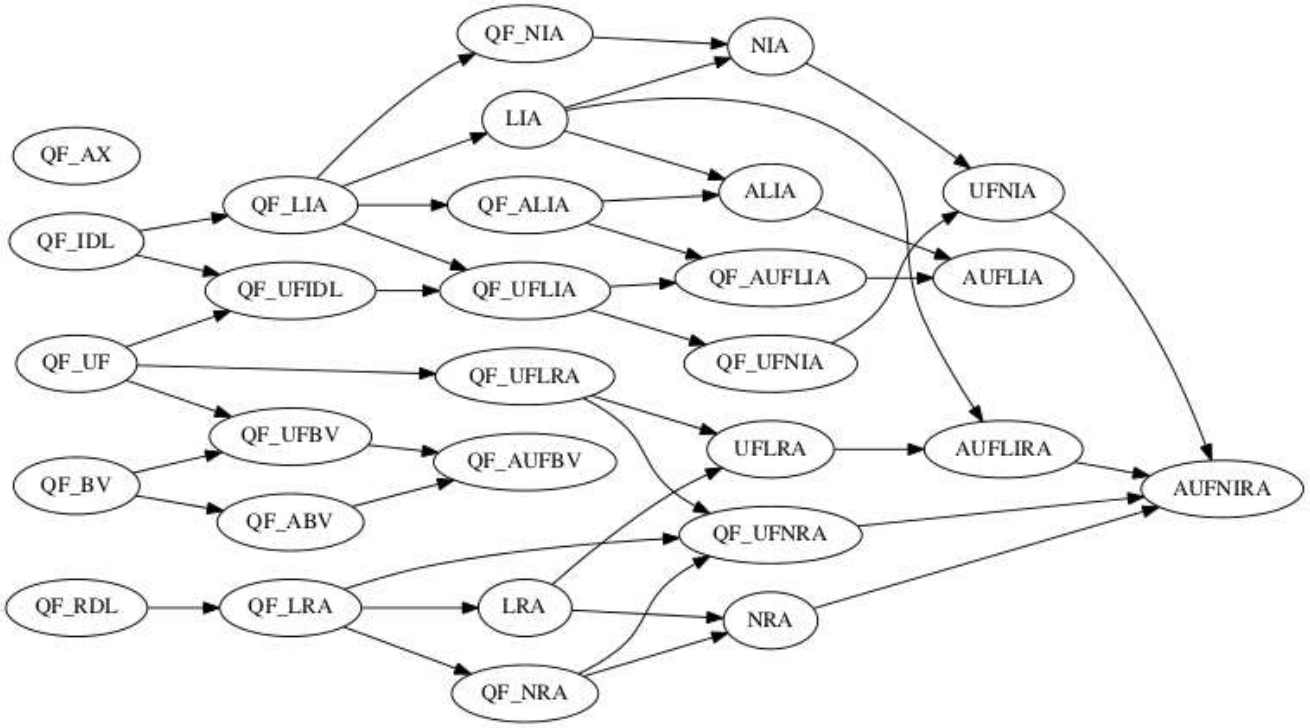
Invocations of is/coulditbe in tests

	`is`	`coulditbe`
Total distinct queries	24085	5771
Queries with result true	2335	3582
Queries with result false	19020	1519
Queries with result FAIL	2730	670
Queries including addition	9849	2022
Queries including multiplication	17097	3106
Queries including equations, inequations, or inequalities	11333	4439
Queries including integer exponents >= 2	7965	875
Queries including complex arithmetic	6215	389
Queries with Boolean structure (¬,∨, ∧)	564	168

© 2016 Maplesoft, a division of Waterloo Maple Inc.

SMT-LIB 2 Logics

Relations between logics induce a partial order.



[Source: http://smtlib.cs.uiowa.edu/logics.shtml]

© 2016 Maplesoft, a division of Waterloo Maple Inc.

SMTLIB Package in Maple

- **SMTLIB** is a Maple package implementing connection to an external SMT solver using the SMT-LIB2 language [1].
- New in Maple 2017, with plans to expand for 2018
- Upcoming version of the package will offer:
  - **ToString** – translate input to string encoding SMT-LIB 2 query (in Maple 2017)
  - **Satisfiable** – check for satisfiability (to be added)
  - **Satisfy** – generate a satisfying truth assignment (to be added)
- Supports some Maple primitives not supported by Logic package: (`forall`, `exists`, `abs`, `floor`, `iquo`, `irem`, `piecewise`)
- The default SMT solver used is Z3, but the interface permits an alternate SMT solver executable to be specified.

© 2016 Maplesoft, a division of Waterloo Maple Inc.

Type Inference

Traverse input expression with the goal of assigning one of the types `{Bool,Int,Real}` to each symbol in the input

1. If type was explicitly provided by the user, done.
   ```
   Satisfy( a + b < 3 ) assuming a::real, b::real;
   ```

2. If expression occurs as an operand in an expression where only one type could be valid, assign that type:
   ```
   Satisfy( a ^ 2 > 1 or abs(c) > 0 or d );
   ```

3. Propagate resolved types through expression:
   ```
   Satisfy( a^2 + 2 = b ) assuming a::integer;
   Satisfy( { a * b = 2.5, b = 1.5 } )
   ```

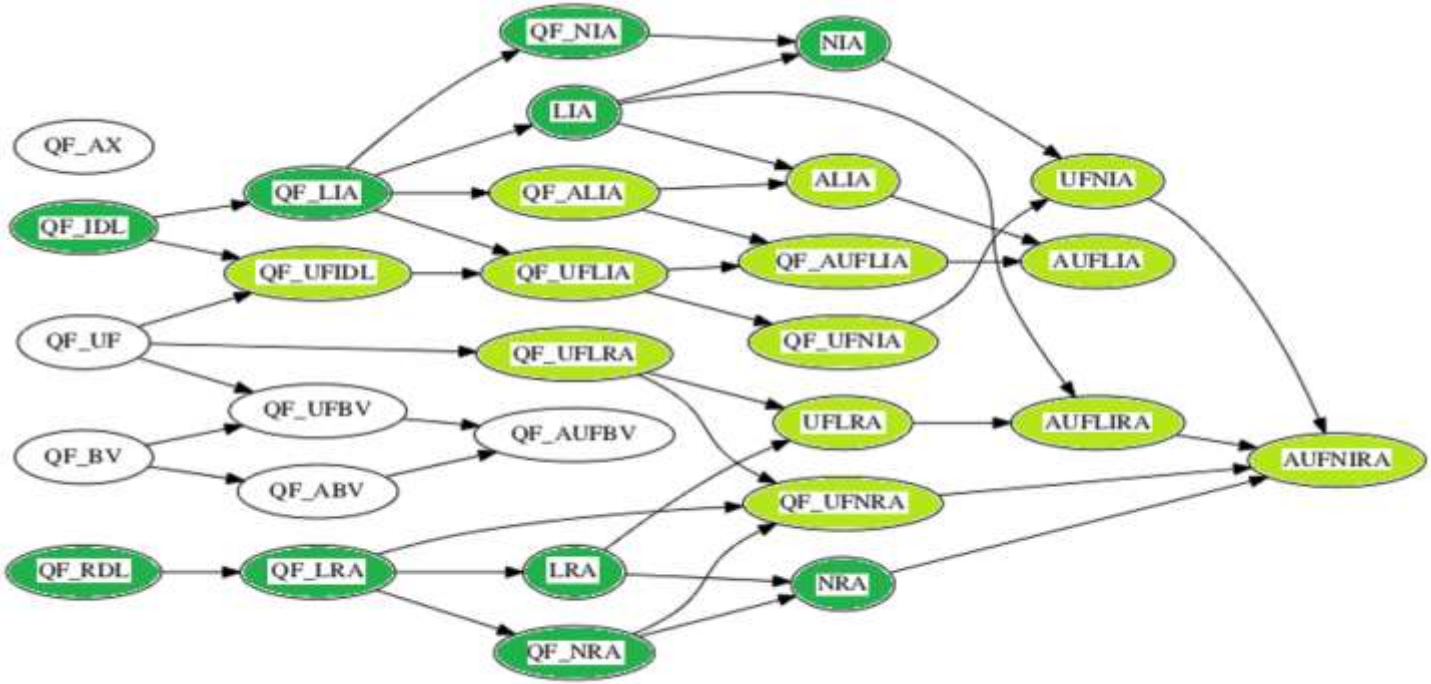
4. Remaining ambiguous expressions:
   - If symbol appears in an arithmetic context, assign to `Int`
   - Otherwise, assign to `Bool`

© 2016 Maplesoft, a division of Waterloo Maple Inc.

Supported SMT-LIB 2 Logics

Dark green indicates logics for which all formulas correspond to some possible input to the Maple `SMTLIB[Satisfy]` command.

Light green indicates logics for which a proper subset of formulas correspond to some possible input to the Maple `SMTLIB[Satisfy]` command.



© 2016 Maplesoft, a division of Waterloo Maple Inc.

Examples

```
> SMTLIBLink:-Satisfy( {w^3+x^3=y^3+z^3, w>0,x>0,y>0,z>0,w<>y,w<>z}, logic="QF_NIA");
```

$$\{w = 10, x = 9, y = 12, z = 1\}$$

```
> SMTLIBLink:-Satisfiable( {x^4+y^4+z^4=w^4, x>0,y>0,z>0,w>=1}, logic="QF_NIA");
```

FAIL

```
> expr := forall([x,y],exists(w,x+w=y));
```

$$expr := \forall([x, y], \exists(w, x + w = y))$$

```
> SMTLIBLink:-Satisfiable( expr, logic="LIA" );
```

true

```
> SMTLIBLink:-Satisfy({x^2+3*x*y+y^2 < 100, x<5, y>=100}, logic="QF_NRA", show_smtlib);
(set-option :produce-models true)
(set-logic QF_NRA)
(declare-fun x () Real)
(declare-fun y () Real)
(assert (and (<= 100 y) (< x 5) (< (+ (* x x) (* (* x y) 3) (* y y)) 100)))
(check-sat)
(get-value (x y))
(exit)
```

$$\{x = -264, y = 101\}$$

© 2016 Maplesoft, a division of Waterloo Maple Inc.

Future Work

- Examine incorporating solver link into Maple proper, i.e. using SMT as a general-purpose tool for solving SMT instances which arise during evaluation of symbolic expressions.
- Find ways of meaningfully incorporating other parts of the SMT-LIB 2 specification into Maple:
  - Support additional types
    - Uninterpreted functions (QF_UF and logics including it)
    - Arrays, bit vectors
    - Floating-point arithmetic?
    - Algebraic datatypes? (new in SMT-LIB 2.6)
  - Offer undecidable cores / proof of UNSAT to user

© 2016 Maplesoft, a division of Waterloo Maple Inc.

References

1. Clark Barrett, Pascal Fontaine, and Cesare Tinelli. *The Satisfiability Modulo Theories Library (SMT-LIB)*, http://www.smt-lib.org, 2016.

2. Logics in SMT-LIB, de Moura, L. M., and Björner, N. *Z3: an efficient SMT solver*. In TACAS (2008) vol. 4963 of Lecture Notes in Computer Science, Springer, pp. 337340. Z3 is available at https://github.com/Z3Prover/z3.

3. Thomas Bouton, Diego Caminha B. de Oliveira, David Deharbe, Pascal Fontaine. (2009). *veriT: An Open, Trustable and Efficient SMT-Solver. 151-156*. https://doi.org/10.1007/978-3-642-02959-2_12.

4. Florian Corzilius, Ulrich Loup, Sebastian Junges, and Erika Abraham. 2012. *SMT- RAT: an SMT-compliant nonlinear real arithmetic toolbox.* In Proceedings of the 15th international conference on Theory and Applications of Satisfiability Testing (SAT'12), Alessandro Cimatti and Roberto Sebastiani (Eds.). Springer-Verlag, Berlin, Heidelberg, 442-448. http://dx.doi.org/10.1007/978-3-642-31612-8_35.

5. Konstantin Korovin, Marek Kosta, Thomas Sturm. *Towards Confict-Driven Learning for Virtual Substitution.* Vladimir P. Gerdt and Wolfram Koepf and Werner M. Seiler and Evgenii V. Vorozhtsov. Computer Algebra in Scientific Computing - 16th International Workshop, CASC 2014, 2014,Warsaw, Poland. Springer, 8660, pp.256-270, 2014, Lecture Notes in Computer Science. http://dx.doi.org/10.1007/978-3-319-10515-4_19.

© 2016 Maplesoft, a division of Waterloo Maple Inc.

Questions

- Thank you!

© 2016 Maplesoft, a division of Waterloo Maple Inc.