Quantifier-Free Linear Arithmetic – Lab Notes

Bow-Yaw Wang

Institute of Information Science
Academia Sinica, Taiwan

July 12, 2015

Useful Z3 Commands

- `from z3 import *` imports Z3 python library.
- `solver = Solver()` creates a Z3 solver called `solver`
- `solver.add(`*formula*`)` adds *formula* to `solver`.
- `solver.check()` checks satisfiability of added formulae.
- `solver.model()` returns a model.
- `Reals(`*names*`)` creates Z3 real variables with *names*.

Complete Example I

```
solver = Solver()
c1, c2, c3, c4 = 1, 1, -1, -1
b1, b2, b3, b4, b5, b6, b7 = 0, 0, 0, 0, 3, 2, 2
v11, v12, v13, v14 = 0, 0, 0, 0
u11, u12, u13, u14 = Reals('u11 u12 u13 u14')
solver.add(-1 * u11 == c1, -1 * u12 == c2)
solver.add(-1 * u13 == c3, -1 * u14 == c4)
""" get values of u11, u12, u13, u14 """

y11, y12, y13, y14 = Reals('y11 y12 y13 y14')
solver.add(-1 * y11 == -1, -1 * y12 == 0)
solver.add(-1 * y13 ==  0, -1 * y14 == 0)
""" get values of y11, y12, y13, y14 """

lambda1, lambda15, lambda16 = Reals('lambda1 lambda15 lambda16')
solver.add(1 * lambda15 == 3)
solver.add(1 * lambda16 == 2)
solver.add(lambda1 == If(lambda15 > lambda16, lambda16, lambda15))
solver.check()
solver.model()
""" find lambda1 """
```

Complete Example II

```
v21, v22, v23, v24 = 2, 0, 0, 0
u21, u22, u23, u24 = Reals('u21 u22 u23 u24')
solver.add( 1 * u21 == c1, -1 * u22 == c2)
solver.add(-1 * u21 -1 * u23 == c3, -1 * u24 == c4)
""" get values of u21, u22, u23, u24 """

y21, y22, y23, y24 = Reals('y21 y22 y23 y24')
solver.add( 1 * y21 - 1 * y23 == 0, -1 * y22 == -1)
solver.add(-1 * y23 ==  0,          -1 * y24 == 0)
""" get values of y21, y22, y23, y24 """

lambda2, lambda25, lambda27 = Reals('lambda2 lambda25 lambda27')
solver.add(2 + 1 * lambda25 == 3)
solver.add(1 * lambda27 == 2)
solver.add(lambda2 == If(lambda25 > lambda27, lambda27, lambda25))
solver.check()
solver.model()
""" find lambda2 """
```

Complete Example III

```
v31, v32, v33, v34 = 2, 1, 0, 0
u31, u32, u33, u34 = Reals('u31 u32 u33 u34')
solver.add( 1 * u31 + 1 * u32 == c1,  1 * u32 == c2)
solver.add(-1 * u31 - 1 * u33 == c3, -1 * u34 == c4)
""" get values of u31, u32, u33, u34 """
solver.check()
solver.model()
```

Exercises

- Is the following $T_{\mathbb{Q}}$-formula satisfiable?

$$\begin{aligned} x + y + z &\geq 1 \\ x - y + z &\geq 2 \\ 2 * x + y - 2 * z &\leq 5 \end{aligned}$$

- Use the simplex method to find a solution with Z3.
- Verify your solution in Z3.