Technische Universität Wien
Fakultät für Informatik
Florian Lonsing

WS 2017
FMI UE Exercise Sheet (Block 2)

185.A93: Exercises on Formal Methods in Computer Science

Document Updates:

- 15 November 2017: Exercise (2a)—added full parentheses.

The topic of these exercises related to block 2 (satisfiability) is modelling and solving satisfiability modulo theories (SMT) problems using the theorem prover Z3:

`https://github.com/Z3Prover/z3`

The schedule of the exercises is as follows:

- November 7: presentation of this exercise sheet and introduction.
- November 27: **submission deadline** (upload in TUWEL).
- December 12: presentation of solutions.

A brief introduction to SMT and Z3 will be given in the first meeting on November 7 (slides available in TUWEL). The following resources may be helpful:

- Tutorial on Z3 (*highly recommended*):
  `https://rise4fun.com/Z3/tutorial/guide`
- Appendix chapter "SMT-LIB: A Brief Tutorial" (available as PDF from TUWEL) of the book *Daniel Kroening and Ofer Strichman. Decision Procedures - An Algorithmic Point of View, Second Edition.*
- SMT-LIB tutorial by David R. Cok:
  `http://smtlib.github.io/jSMTLIB/SMTLIBTutorial.pdf`

A total number of 15 points can be achieved by solving the exercises on this sheet. The *bonus exercise (4d)* is worth up to three additional points. However, the total number of points that can be achieved for this sheet is still 15. That is, the additional points allow you to make up for points lost in other exercises on *this* sheet.
Please consider the following *important guidelines* when preparing and submitting your solutions to the exercises:

- Your submission of the solution to a (sub)exercise must contain the following files:
  - a text file containing a list of commands that can be interpreted by Z3 as *input*. Examples of input files and Z3 commands will be provided in the meeting on November 7 when this exercise sheet is presented.
  - a text file containing the *output* of Z3 that is produced for the respective input file.

  Please note that your solution will *not* be considered for grading if either of the two files is missing.

- Please *explain and comment* each of your solutions *in detail*. Explanations and comments are considered an *essential part* of your submission and are *necessary to receive full points* for an exercise.

  Explanations and comments should be included directly in the submitted input files of Z3 by prefixing the comment lines with a semicolon ";". Any line prefixed with ";" is interpreted as a comment and hence ignored by Z3.

- Please submit the solutions to the exercises as *separate text files* by uploading them in TUWEL, and adhere to the following file naming conventions:

  `<surname>-<matrnr>-<exnr>-input.txt`

  `<surname>-<matrnr>-<exnr>-output.txt`

  where `<surname>` is your surname, `<matrnr>` is your matriculation number, `<exnr>` is the number of the subexercise on this sheet, and `input` or `output` indicate whether the file contains the input or output of Z3, respectively.

- Please submit your solution *on time* before the deadline. Late submissions will not be considered for grading.

- Please make sure that the input text file `<surname>-<matrnr>-<exnr>-input.txt` is *free of syntactic errors*. Note that your solution will *not* be considered for grading if Z3 reports any syntactic errors.

  For example, consider the following erroneous command:

  ```
  (echo "hello world!)
  ```

  Z3 reports the following syntactic error on the above command:

  ```
  (error "line 2 column 1: unexpected end of string")
  ```

  The correct command is as follows:

  ```
  (echo "hello world!")
  ```

General information on the organization of this course was presented in the kick-off meeting (slides available in TUWEL).

Exercise 1 Installation of Z3 and Basic Use **2 Points**

Download and install version 4.5.0 of Z3:
`https://github.com/Z3Prover/z3`
Read the Z3 tutorial (not all parts are relevant for this exercise sheet):
`https://rise4fun.com/Z3/tutorial/guide`
Given the propositional formulas $\phi_1 := p \rightarrow q$ and $\phi_2 := \neg q \rightarrow \neg p$, use Z3 to prove that the formula $\phi_1 \leftrightarrow \phi_2$ is valid.

Exercise 2 Propositional Logic: Counterexamples to Validity **1+2 Points**

(a) Use Z3 to show that the propositional formula $\phi := ((x \otimes y) \wedge (y \otimes z)) \rightarrow (x \otimes z)$ is not valid, where $\otimes$ denotes the XOR operator, and compute a concrete counterexample (i.e., concrete truth assignments to x, y, and z).

(b) Use Z3 and its `get-model` command to iteratively and manually enumerate all counterexamples to the validity of ϕ.

Exercise 3 Integer vs. Bitvector Arithmetic **1+1+1 Points**

(a) Consider line 6 of the `binsearch` program (lecture slide set 3): `m = (l + h) / 2`. Use Z3 to check whether the assertion $l \leq m \wedge m \leq h$ holds in the theory of bitvectors of size 3, assuming that $l \leq h$ as in the condition of the `while`-loop in `binsearch`. Justify your answer in detail, e.g., by providing a concrete counterexample.

(b) Like (a) but in the theory of integers.

(c) Like (a), but instead of the statement
```
m = (l + h) / 2
```
consider the statement
```
m = l + ((h - l) / 2)
```
and check whether the assertion holds.

Exercise 4 Equality Logic and Uninterpreted Functions **1+2+1(+3) Points**

(a) Consider the *EUF*-formula

$$\varphi^{EUF} := x = y \wedge F(x) = G(y) \wedge z = G(F(y)) \wedge z \neq G(F(x))$$

where F and G denote uninterpreted functions. Use Z3 to check whether φ^{EUF} is *E-satisfiable.*

(b) Apply Ackermann's reduction to φ^{EUF} to obtain the E-formula φ^E and use Z3 to check whether φ^E is E-satisfiable.

(**Hints:** recall that Ackermann's reduction of φ^{EUF} results in an E-formula φ^E that is *E-valid* if and only if φ^{EUF} is *E-valid.* Further, recall that $\neg\varphi^{EUF}$ is E-valid if and only if φ^{EUF} is E-unsatisfiable.)

(c) Consider the formula φ^E from (b) that results from φ^{EUF} by Ackermann's reduction. Construct a formula φ^E_M from φ^E by removing as many functional consistency constraints as possible such that φ^E_M and φ^E have the same logical status. Use Z3 to check whether φ^E_M is E-satisfiable.

(d) **Bonus exercise:** Like (c) but use the unsatisfiable core extraction feature of Z3 (command `get-unsat-core`) to find a *not necessarily minimal subset* of the functional consistency constraints in φ^E.

Exercise 5 Equality Logic and Uninterpreted Predicates **1+2 Points**

Given the *EUF*-formula

$$\varphi := (y = z \wedge Q(z, x) \wedge P(x, z)) \rightarrow (P(x, y) \wedge Q(x, y))$$

where P and Q denote uninterpreted predicates.

(a) Use Z3 to check whether φ is E-valid.

(b) Like (a) but assume that, in addition to the usual axioms of equality logic, the following axiom is part of the theory: $\forall x, y.\ Q(x, y) \leftrightarrow Q(y, x)$.