

Lecture 7

First-Order Theories II

Zvonimir Rakamarić
University of Utah

slides acknowledgements: Zohar Manna

Last Time

- ▶ First-order theories
- ▶ Theory of equality
- ▶ Arithmetic over integers and natural numbers
 - ▶ Peano arithmetic
 - ▶ Undecidable
 - ▶ Presburger arithmetic
 - ▶ No multiplication between two variables
 - ▶ Decidable
 - ▶ Theory of integers
 - ▶ Same expressiveness as Presburger arithmetic

This Time

- ▶ Theory of reals
- ▶ Theory of rationals
- ▶ Theory of arrays
- ▶ Exercises with SMT solver Z3

Theory of Reals $T_{\mathbb{R}}$ and Rationals $T_{\mathbb{Q}}$

$\Sigma_{\mathbb{R}} : \{0, 1, +, -, *, =, \geq\}$ with multiplication

$\Sigma_{\mathbb{Q}} : \{0, 1, +, -, =, \geq\}$ without multiplication

Decidability of $T_{\mathbb{R}}$ and $T_{\mathbb{Q}}$

- ▶ Both are decidable
 - ▶ High time complexity
- ▶ Quantifier-free fragment of $T_{\mathbb{Q}}$ is efficiently decidable

Theory of Arrays T_A

$\Sigma_A : \{select, store, =\}$

where

- ▶ $select(a, i)$ is a binary function:
 - ▶ read array a at index i
- ▶ $store(a, i, v)$ is a ternary function:
 - ▶ write value v to index i of array a

Axioms of T_A

1. $\forall a, i, j. i = j \rightarrow \text{select}(a, i) = \text{select}(a, j)$
(array congruence)
2. $\forall a, v, i, j. i = j \rightarrow \text{select}(\text{store}(a, i, v), j) = v$
(select-store 1)
3. $\forall a, v, i, j. i \neq j \rightarrow \text{select}(\text{store}(a, i, v), j) = \text{select}(a, j)$
(select-store 2)

Note about T_A

- ▶ Equality (=) is only defined for array elements...

- ▶ Example:

$select(a, i) = e \rightarrow \forall j. select(store(a, i, e), j) = select(a, j)$

is T_A -valid

- ▶ ...and not for whole arrays

- ▶ Example:

$select(a, i) = e \rightarrow store(a, i, e) = a$

is not T_A -valid

Decidability of T_A

- ▶ T_A is undecidable
- ▶ Quantifier-free fragment of T_A is decidable

Theory of Arrays with Extensionality $T_A^=$

- ▶ Signature and axioms of $T_A^=$ are the same as T_A , with one additional axiom:

$$\forall a, b. (\forall i. \text{select}(a, i) = \text{select}(b, i)) \leftrightarrow a = b$$

(extensionality)

- ▶ $T_A^=$ -valid example

$$\text{select}(a, i) = e \rightarrow \text{store}(a, i, e) = a$$

Decidability of $T_A^=$

- ▶ $T_A^=$ is undecidable
- ▶ Quantifier-free fragment of $T_A^=$ is decidable

Summary of Decidability Results

	Theory	Quantifiers Decidable	QFF Decidable
T_E	Equality	NO	YES
T_{PA}	Peano Arithmetic	NO	NO
$T_{\mathbb{N}}$	Presburger Arithmetic	YES	YES
$T_{\mathbb{Z}}$	Linear Integer Arithmetic	YES	YES
$T_{\mathbb{R}}$	Real Arithmetic	YES	YES
$T_{\mathbb{Q}}$	Linear Rationals	YES	YES
T_A	Arrays	NO	YES

Summary of Complexity Results

	Theory	Quantifiers	QF Conjunctive
PL	Propositional Logic	NP-complete	$O(n)$
T_E	Equality	–	$O(n \log n)$
$T_{\mathbb{N}}$	Presburger Arithmetic	$O(2^{2^{2^{kn}}})$	NP-complete
$T_{\mathbb{Z}}$	Linear Integer Arithmetic	$O(2^{2^{2^{kn}}})$	NP-complete
$T_{\mathbb{R}}$	Real Arithmetic	$O(2^{2^{kn}})$	$O(2^{2^{kn}})$
$T_{\mathbb{Q}}$	Linear Rationals	$O(2^{2^{kn}})$	PTIME
T_A	Arrays	–	NP-complete

Z3 SMT Solver

- ▶ <http://rise4fun.com/z3/>
- ▶ Input format is an extension of SMT-LIB standard
- ▶ Commands
 - ▶ `declare-const` – declare a constant of a given type
 - ▶ `declare-fun` – declare a function of a given type
 - ▶ `assert` – add a formula to Z3's internal stack
 - ▶ `check-sat` – determine if formulas currently on stack are satisfiable
 - ▶ `get-model` – retrieve an interpretation
 - ▶ `exit`

Linear Integer Arith. Example 1

$$x \leq y \wedge z = x + 1 \rightarrow z \leq y$$

Linear Integer Arith. Example 2

$$x \leq y \wedge z = x - 1 \rightarrow z \leq y$$

Linear Integer Arith. Example 3

$$1 \leq x \wedge x + y \leq 3 \wedge 1 \leq y \rightarrow x = 1 \vee x = 2$$

Dog, Cat, and Mouse Puzzle (from Z3 page)

- ▶ Puzzle
 - ▶ Spend exactly \$100 and buy exactly 100 animals.
 - ▶ Dogs cost \$15, cats cost \$1, and mice cost 25 cents each.
 - ▶ You have to buy at least one of each.
 - ▶ How many of each should you buy?
- ▶ Use linear integer arithmetic

Scheduling Example

	Machine 1	Machine 2
Job 1	2	1
Job 2	3	1
Job 3	2	3

- ▶ Table gives time units required to process Job x on Machine y
- ▶ For a job, complete a phase on Machine 1 before starting the next on Machine 2
- ▶ Find using Z3 whether jobs can be scheduled in T time units
 - ▶ Try $T=6$, $T=7$, $T=8$