Dr. Matthias Heizmann
Dominik Klumpp

Hand in until 10:00 on June 03, 2020
Discussion: June 03, 2020

Tutorial for Program Verification
Exercise Sheet 5

In this exercise sheet we work with First-Order Logic and First-Order Theories. The last exercise serves as preparation for the next part of the lecture, in which we will define the syntax (and later the semantics) of a programming language.

Submit your solution by uploading it as PDF in ILIAS.

Exercise 1: Sudoku in First-Order Logic 3 Points

Formalize the rules of Sudoku in First-Order Logic.
An n-Sudoku (for $n \in \mathbb{N}$) is given as $(n^2 \times n^2)$-grid of numbers from 1 to n^2. The grid is further divided into n^2 squares of size $(n \times n)$. Use the following predicate and function symbols:

- the binary equality predicate $\cdot = \cdot$,
- the binary function $num(\cdot, \cdot)$ such that $num(x, y)$ returns the number in column x and row y,
- the binary function $square(\cdot, \cdot)$ such that $square(x, y)$ returns the the number of the square containing cell (x, y).

Assume the underlying domain contains only the elements $\{1, \ldots, n^2\}$ and the meaning of the equality predicate is already defined (e.g., because we consider the theory of equality).

(a) In every row, each number occurs at least once.

(b) In every row, each number occurs at most once.

(c) In every column, each number occurs at least once.

(d) In every column, each number occurs at most once.

(e) In every square, each number occurs at least once.

(f) In every square, each number occurs at most once.

Exercise 2: First-Order Theories 4 Points

In the lecture we have seen that the signature of Peano arithmetic does not contain a symbol for $\geq$, the "greater-or-equal" relation on natural numbers. However, we can define this relation by a formula in Peano arithmetic.

$$x \geq y :\equiv \exists z \,.\, x = y + z$$

In a similar manner, formalize the following statements. **Use only the function, constant and predicate symbols of Peano arithmetic, and possibly statements you have already defined.**

> *Note:* The purpose of this exercise is not only that we get more familiar with first-order logic. We will later in the course learn program analyses that can only reason about sets of program states that can be defined in a certain theory. Users and developers of program analysis tools that have a good intuition which properties can be expressed in certain theories can more easily estimate the power of these program analyses.

(a) $x \mid y$, i.e., x is a divisor of y

(b) x is an odd number

(c) x is not divisible by 4

(d) x is a prime number

(e) z is the greatest common divisor of x and y

Let us now consider the combination of the theory of equality and the theory of Peano arithmetic. Let us then consider an 1-ary function symbol that we use to represent an array. We consider the domain of the function as the indices of the (infinite) array and the range of the function as the values of the (infinite) array. Formalize the following statements:

(f) The array is sorted. Positions with higher indices have (not necessarily strictly) higher values.

(g) At all positions between five and 42 (inclusive) the value is even.

(h) Every value occurs at least at two different positions.

Exercise 3: Satisfiability of FOL Formulas 2 Points

Are the following formulas φ_i satisfiable with respect to the theory of integers $T_{\mathbb{Z}}$? Write an SMT script to determine the satisfiability of each formula and, if it is satisfiable, a satisfying model. You may use an SMT solver like Z3[1] to execute this script. If the formula is satisfiable, give a satisfying model.

[1] https://rise4fun.com/Z3

(a) φ_1 : $\forall x, y.\ a \neq 21 \cdot x + 112 \cdot y$

(b) φ_2 : $\exists x.\ (x = 10 \cdot a + b \wedge a + b = 9 \wedge \neg \exists y.\ x = 3 \cdot y)$

Exercise 4: Context-Free Grammars 2 Points

Consider the context-free grammar $\mathcal{G} = (\Sigma, N, P, S)$ over the alphabet $\Sigma = \{a, b, c\}$ that has the nontermination symbols $N = \{S, A, B\}$ and the following derivation rules.

$$\begin{aligned} P = \{ \quad & S \to AabBBc, \\ & A \to \varepsilon \mid aB, \\ & B \to ab \mid bc \mid c \mid A \quad \} \end{aligned}$$

(a) Find a word that is not in the language of $\mathcal{G}$.

(b) Find a derivation tree for the word $w = abcabc$.