Dr. Matthias Heizmann
Tanja Schindler
Dominik Klumpp

Hand in until 10:00 on May 13, 2019
Discussion: May 13, 2019

Tutorial for Program Verification
Exercise Sheet 4

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.

(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 funtion 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.

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 anlysis tools that have a good intuition which properties can be expressed in certain theories can more easily estimate the power of these program analyses.

Exercise 3: Natural Deduction Proofs for First-Order Logic 2 Points

Pick one of the following implications and use the Natural Deduction proof system for First-Order Logic $\mathcal{N}_{\mathsf{FOL}}$ to show that the implication holds.

- $\{(\forall x.\ p(x)) \lor (\forall x.\ q(x))\} \models \forall x.\ p(x) \lor q(x)$
- $\{\exists x.\ \forall y.\ p(x, y)\} \models \forall y.\ \exists x.\ p(x, y)$

If you solve this task for both implications you get additionally two bonus points.

Exercise 4: Proof rules of $\mathcal{N}_{\mathsf{FOL}}$ 2 Points

In the last lecture, you have seen the rules for the Natural Deduction proof system for First-Order Logic $\mathcal{N}_{\mathsf{FOL}}$. The proof system $\mathcal{N}_{\mathsf{FOL}}$ extends the proof system $\mathcal{N}_{\mathsf{PL}}$ by four rules.

Show that the side conditions of the rules $(I_\forall)$ and $(E_\exists)$ are necessary for the correctness of the rules. That is, for each of the rules, ignore the side condition and give an example such that the implication above the horizontal line holds, but the implication below the horizontal line does not hold.