Test 1, Theory of Computations, Spring 2026

Problem 1. Translate, step-by-step, the following for-loop into a primitive recursive expression:

```
int x = a + b;
for (int i = 1; i <= c; i++)
  {x = x * x - x;}
```

You can use $sum(.,.)$, $mult(.,\ .)$ (product) and $sub(,,.)$ (subtraction) in this expression.

Problem 2. Translate, step-by-step, the following primitive recursive function into a for-loop:

$$F = \sigma(PR(mult(\pi^2_1, \pi^2_2), sum(\pi^4_4, \pi^4_3))).$$

For this function F, what is the value $F(2, 1, 2)$?

Problem 3-4. Prove, from scratch, that the function $n!\,\%\,(n+1)$ is primitive recursive, where $n! = 1 \cdot \ldots \cdot n$ is factorial and $a\,\%\,b$ is remainder. Start with the definitions of a primitive recursive function, and use only this definition in your proof – do not simply mention results that we proved in class, prove them.

Problem 5. Prove that the following function $f(a, n)$ is μ-recursive: $f(a, n) = n!\,\%\,(n+1)$ when $n!\,\%\,(n+1) = a$, and $f(a, n)$ is undefined for all other n. You can use the fact that remainder and factorial are primitive recursive.

Problem 6. Translate the following μ-recursive expression into a while-loop:

$$f(a) = \mu n.(n!\,\%\,(n+1) = a).$$

Take into account that $0! = 1$ and that $n\,\%\,1 = 0$ for all n. For this function f, what is the value of $f(0)$? $f(1)$? $f(2)$?

Problem 7-8. What if, in addition to 0, π^k_i, and σ, we also allow the physics-related function $Ph(n)$ – the result of n-th experiment – in our constructions? Let us call functions that can be obtained from 0, π^k_i, σ, and $Ph(n)$ by using composition and primitive recursion *Ph-primitive recursive* functions. Will then every computable function be Ph-primitive recursive? Prove that your answer is correct.

Problem 9. Design a Turing machine for computing the constant function $f(n) = 2$ in binary code. In other words, no matter what combination of 0s and 1s was originally placed on the tape, this Turning machine should return the number $2 = 10_2$. Trace your Turing machine for $n = 3$.