FUN WITH FORMAL METHODS IS MORE THAN FUN

N. V. Shilov[1,2*]

1) School of Science and Technology, Nazarbayev University, Astana, Kazakhstan; 2) A.P.Ershov Institute of Informatics Systems Russian Academy of Science, Novosibirsk, Russia; *nikolay.shilov@nu.edu.kz

Introduction. Almost half of a century has passed since Robert W. Floyd published the first research that explicitly discussed formally how to assign meaning to programs. But recently David L. Parnas have called Really Rethinking "Formal Methods", to question the assumptions underlying the well-known current formal software development methods to see why they have not been widely adopted and what should be changed. So, things are right where they started decades ago?

Not at all, since industrial applications of Formal Methods are not the unique measure of success. Another dimension where we can discuss utility of Formal Methods could be better education. A very popular (in Russia) aphorism of Mikhail Lomonosov (the first Russian academician) says: "Mathematics should be learned just because it disciplines and bring up the mind". Similarly, Formal Methods discipline and bring up minds in Computer Science. and help to bridge a "cultural gap" (E.W.Dijkstra) between Mathematics and Computer Science.

Method. A part of the reason of student's and engineer's poor attitude to Formal Methods (FM), is very simple: FM-experts do not care about primary education in this field at the early stage of higher education. In particular, many courses on Formal Semantics start with fearful terms like state machine, logic inference, denotational semantics, etc., without elementary explanations of the basic notions, without challenging and paradoxical but simple examples of how Formal Methods work.

In the framework of project "Research of Formal Models for analysis of programs with Dynamic Memory" a number of challenging and paradoxical but simple examples of use of Formal Methods were developed:

- Solving finite position games with MS Excel;
- Solving Rubik's Cube puzzles with model checking tool Spin;
- Debugging (using axiomatic semantics) applications that compute e and n incorrectly.

In particular, C-source file MonteCarlo.c that may compute wrong value for n follows below:

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
int main(void){
srand(time(NULL));
int i, j, r, n = 10;
float pi_val, x, y;
int n_hits, n_trials=1000000;
for(j = 0; j < n; j++){n_hits=0;
for(i = 0; i<n_trials; i++){
r = rand()% 10000000;
x = r/10000000.0;
r = rand()% 10000000;
y = r/10000000.0;
if(x*x+y*y<1.0)n_hits++;}
pi_val=
4.0*n_hits/(float)n_trials;
printf(«%f\n», pi_val); }
return 0;}
```

Acknowledgement. Project "Research of Formal Models for analysis of programs with Dynamic Memory" has been supported by Nazarbayev University Seed Grant KF-14/16.