Model-Checking Linear Dynamical Systems under Floating Point Rounding

1 Context

Loops are a fundamental staple of any programming language, and the study of loops plays a pivotal role in many subfields of computer science, including automated verification, abstract interpretation, program analysis, semantics, etc. The focus of the present internship is on the algorithmic analysis of simple (i.e., non-nested) linear (or affine) while loops, such as the following:

```
x = 3, y = 4, z = 2
while x+3y+z > 4:
    x = 3x +2z
    y = 3x + y
    z = y + z
```

We are interested in analysing how the loop evolves. A simple reachability query is to decide whether the loop variables ever satisfy a (Boolean combination of) polynomial inequalities, for example modelling a loop guard. More generally, model checking can express significantly more complex temporal properties, such as those expressible in linear temporal logic or monadic second order logic.

Modelling the evolution of such a loop may require unbounded memory. That is, the number of bits required to represent the numbers x, y, and z may grow larger and larger. However, most computer systems do not represent rational numbers to arbitrary precision, but rather use *floating-point rounding*, in which a number y is stored using two components: the mantissa $m \in \mathbb{Q}$ and the exponent $\alpha \in \mathbb{Z}$, such that $y = m \cdot 2^{\alpha}$.

Formally, programs can be modelled using linear dynamical systems (LDS), which are composed of a starting vector representing the initial state of each variable of the system and of a matrix describing the evolution of the program. An LDS generates an infinite sequence of vectors (the *orbit* of the system) by multiplying the matrix with the current vector and then applying floating-point rounding to the result.

2 Subject

LDS have been widely studied over the years, in particular to show terminations of loops, either directly [4] or through approximate schemes such as invariant generation [2]. However, the LDS in most of those works do not take into account the limited memory of the actual system.

There are two notable exceptions.

- In [1], the author study the model-checking of an LDS, where every coefficient is rounded to the closest integer. This line of work unfortunately led to extremely complex and sometimes longstanding open problems.
- In [3], the authors consider a form of floating point rounding where the mantissa is bounded while the exponent is not (note that if both are bounded, the system is finite which entails decidability of most problems immediately). Their work shows that model-checking can be achieved under some conditions on the initial system.

The goal of this internship is to continue the research started in [3]. In particular, we wish to refine the existing algorithm to establish its precise complexity and to consider invariant synthesis when the algorithm fails.

3 Expected skills

The intern should have some knowledge of formal verification and be interested in working with the mathematical aspects of computer science.

References

[1] Christel Baier, Florian Funke, Simon Jantsch, Toghrul Karimov, Engel Lefaucheux, Joël Ouaknine, Amaury Pouly, David Purser, and Markus A. Whiteland. Reachability in dynamical systems with rounding. In *FSTTCS 2020*, volume 182 of *LIPIcs*, pages 36:1–36:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[2] Ehud Hrushovski, Joël Ouaknine, Amaury Pouly, and James Worrell. Polynomial invariants for affine programs. In *LICS 2018*, pages 530–539. ACM, 2018.

[3] Engel Lefaucheux, Joël Ouaknine, David Purser, and Mohammadamin Sharifi. Model checking linear dynamical systems under floating-point rounding. *CoRR*, abs/2211.04301, 2022.

[4] Joël Ouaknine and James Worrell. Positivity problems for low-order linear recurrence sequences. In *SODA 2014*, pages 366–379. SIAM, 2014.