Verifying Computations with Streaming Interactive Proofs  
by G. Cormode, J. Thaler, and K. Yi

1 Summary

In this paper authors presented a set of protocols for verifying some computations over streams of data. The verifier (e.g., that outsourced computations) needs to pass over the input once and uses only log space. After getting the result of computations from the prover (e.g., untrusted cloud provider) it initiates a protocol that verifies results with logarithmic communication spread over a logarithmic number of rounds. After concluding the protocol run, the verifier is highly confident if computations have been performed. For correct computations protocol always accept, while for any disturbance in computation (intentional or unintentional), protocol accepts with negligible probability \( \log \frac{u}{p} \), where \( u \) is the domain size of stream items, and \( p \) is a prime number such that \( u < p \leq 2u \). Authors use polynomial interpolation to verify computations, but they do not describe it clearly.

All protocols presented in the paper are efficient for \( NP \) problems and \( NC \) problems. Note a problem belongs to the class \( NC \), if there exist constants \( c \) and \( k \) such that it can be solved in time \( O(\log^c(n)) \) using \( O(n^k) \) parallel processors. Informally, \( NC \) is a class of problems, which can be easily solved on parallel processors, and \( NC \subset P \), but we do not know it \( NC = P \).

2 Strong Points

- All proofs and conclusions are correct and complete.
- Detailed proofs and analysis have been moved to appendix to not reduce clarity of the paper.
- Authors presented verification protocols just for a few algorithms, and reuse them to verify correctness of computations for other algorithms, e.g., adapting \( Sub-vector \) verification protocol they define verification protocols for \( Index \), \( Dictionary \), \( Predecessor \), and \( Range-Query \).

3 Weak Points

- The paper is theoretical, and difficult to read – no baby examples that would help to understand definitions and notations. The latter ones are often not necessary, e.g., \( (i)_d^l \), which denotes the \( k \)-th least significant digit of \( i \) in base-\( l \) representation.
- An unclear explanation why \( u \) in one context means domain of values (Section 1), in other it is a size of input vector (Section 2).
- Some theoretical background knowledge of readers is assumed, but not mentioned and defined clearly.
- Authors assume that the domain of stream values is limited to \( u \) items, and their random value \( r \), which is used in verification is greater than \( u \). In addition, the base of the algebraic group for the computed polynomials is a prime number \( p \) taken from the range \( u < p \leq 2u \). Finding \( r \) and \( p \) for an infinite domain of the stream values is not discussed.

4 Questions and Discussion Points

- This approach of verifying if a protocol has been computed correctly, can be used in our research to check if any provider is malicious. The identity of such provider may not be disclosed.
- Authors silently limited their results to functions on integer values. They do not consider floating point number, which are used in majority of computations. It would be interesting to extend their research to functions of such numbers.

Preferences for Paper Presentations.

1. Summarization and Matching of Density-Based Clusters in Streaming Environments by D. Yang, E. A. Rundensteiner, and Matthew O. Ward
2. Mining Recent Temporal Patterns for Event Detection in Multivariate Time Series Data by I. Batal, D. Fradkin, J. Harrison, F. Moerchen, M. Hauskrecht