Benchmarking SNARKs

Bachelor Thesis / FP
Jens Ernstberger, May 8, 2023

Title: “Benchmarking SNARKs”
Supervisor: Jens Ernstberger
Period: 9 weeks

Context
The landscape of SNARKs is vast and constantly changing. From a theoretical perspective, a novel SNARK is constructed by combining an interactive protocol, called a polynomial IOP, with a polynomial commitment scheme. The resulting construction is rendered non-interactive by applying the Fiat-Shamir transform, which replaces random challenges chosen by the verifier through a locally computed hash. In practice, additional tooling allows developers to specify the circuit in either a domain specific programming language, or through a library in a commonly known language. The process of defining the circuit, compiling the circuit to an intermediate representation, and generating a witness given the assignment of public and private inputs is commonly referred to as the frontend of a SNARK implementation. The backend, on the other hand, handles the generation of the setup (if required), the generation of the proof as well as the verification of the proof. To facilitate a polynomial IOP in practice, the backend relies on arithmetic operations in finite fields and, depending on the proof system and polynomial commitment applied, elliptic curve cryptography.

The goal of this work is to extend an already existing benchmarking framework to systematically evaluate memory consumption of common SNARKs [1]. As such, you should extend the support to other libraries that are not yet supported.

Requirements
- High: Independent work ethic and mathematical background.
- High: Programming in Rust.

Tasks
1. Familiarization with SNARKs and the benchmarking framework.
2. Developing custom benchmarking utilities in Rust.
3. Extending support for the existing framework.
4. Writing the report.

Preliminary Schedule

<table>
<thead>
<tr>
<th>Task / Week</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Familiarization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Developing custom benchmarking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tools in Rust</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extend the framework support</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

References