What the Compiler Prints: A Verified Round Trip

Advisor: **Roderick Bloem**

Motivation

We have a compiler that protects circuits against power side-channel attacks, with a mechanized proof in Lean 4 that its output is secure. The proof is about data structures, but the compiler emits text. That the text denotes the circuit the proof is about is the one link in the chain that is trusted rather than proved — and it is where the bugs have been found so far.

Printer and parser both exist in Lean. Proving them correct is harder and more interesting than it looks. If you print and parse a circuit, you do NOT get the same circuit: there are small differences like the names of wires. Thus, the theorem to aim at — equality up to renaming, equality of meaning, or exact equality — is itself the first question of the thesis.

Goals and Tasks

- Study the text format, its printer and parser, and a circuit that refutes the naive claim
- Settle the statement worth proving, and say why it's better than the alternatives
- Supply the missing groundwork: separator lemmas over lists, injectivity of decimal numerals
- Prove the round trip through a token list: lexing inverts rendering, parsing inverts printing
- Stretch: the grammar invariant that rules out the parser's error branches

Literature

- X. Leroy
  Formal Verification of a Realistic Compiler
  Communications of the ACM 2009
  doi:10.1145/1538788.1538814
- L. de Moura and S. Ullrich
  The Lean 4 Theorem Prover and Programming Language
  CADE 2021
  doi:10.1007/978-3-030-79876-5_37

Courses & Deliverables

☑ **Master Project**
Project code
Report
Presentation

– OR –

☑ **Master's Thesis**
Initial presentation
Project code
Thesis (60+ pages)
Final presentation

Recommended if you're studying

☑ CS ☑ ICE ☑ SEM

Prerequisites

- Enjoys proofs and precise definitions
- Some Lean, or the patience to learn it

Advisor Contact

roderick.bloem@tugraz.at