SONNX

Formal methods meeting #2

Objective and outputs of the meeting

- Objectives
    - Do we need formal techniques in the context of SONNX?
        - NO: *thank you and bye-bye!*
        - YES: **For what purpose? Using which language, tools?**
- Outputs
    - A clear statement of the purpose of FMs in SONNX
    - A formalisation strategy
    - A formalism
    - A short term workplan

Why would we need formal methods?

To express the semantics of operations and graph execution unambiguously?

- 😈 Will the spec be something different from the algorithm/code?
- 😈 Wouldn't it be simpler to provide a straightforward, traceable and verified implementation, i.e., "Operation X is what is realized by the following reference code...".
  - How would we account for difference in low-level implementation?
  - Example: Arm's TOSA
- 😈 Are FMs really necessary considering the (simple) operators at stake?
- 😈 Will the usage of a cryptic formalism simplify the task of developers?

Why would we need formal methods?

To verify the specification (completeness, absence of inconsistencies...)?

To describe the algorithm and verify it against the specification?

- Example with Why3 (Clochard et al, The Matrix Reproved)

Why would we need formal methods?

To verify the reference implementation of the algorithm?

- 😈 Is it really necessary considering that this is a one-shot effort?

To generate a the reference implementation?

- 😈 Is it really necessary considering that this is a one-shot effort?

Applied on what?

- Operations (esp. Tensor operations)
  - Specification of the ONNX operators (possibly on the vasis of a library of basic operators (e.g., linear algebra))
- Graph of operations
  - Specification of what is a graph (what it is composed of?), how are graph executed?
- ONNX format?
  - Give a formal semantic to the ONNX "language"?

Using which language, tools?

- Using "Pen and paper" specification (Loïc)
- Using some specification languages (ACSL, Why3, Coq, other)
  - Which one?
- What are the expected properties of a formal specification langage in our contex?
  - Clarity? Understandability (by non experts)
  - Expressivity?
  - Support of tools?
- What do we expect from the formal language?
- What expertize do we have at hand
  - Why3?, ACSL, other?

What we have done yet?

- First exercize on the `CONV2D` operator using Why3 and ACSL, see here

Issues

- Formal specification of floating point operations
    - What do we want to specify, precisely?

Refs

General

- Krichen *et al*, Are Formal Methods Applicable To Machine Learning and Artificial Intelligence
- Urban and Miné, "A Review of Formal methods Applieds to Machine Learning

Refs

Operators semantics

- Coq, see Kellison *et al, "LAProof: A Library of Formal Proofs of Accuracy and COrrectness for Linear Algebra Programs]

Refs

Graph semantics

- [Arm's TOSA] (https://www.mlplatform.org/tosa/tosa_spec.html#operator-graphs)
- Gauffriau et al. Formal description of ML models for unambiguous implementation, use Petri nets (example on LeNet)

Refs?

Verification of the low-level implementation of code

- Formal verification on Deep Learning Instruction of GPU