

SONNX V&V

SONNX Outputs

- Operators of an ML-Model
 - Informal specification
 - Reference input values (for testing the implementation of an operator)
 - Formal specification (*Is the Formal specification an output of SONNX?*)
 - Trusted Reference implementation (from formal specification)
- Graph of an ML-Model
 - Informal specification
 - Formal specification (*Is the Formal specification an output of SONNX?*)
 - Reference implementation
- Graph exchange format and data storage
 - To be filled

SONNX V&V: Operator informal specification

- Informal specification validation
 - *Review*
 - *Inputs and applicable documents*
 - ONNX operators [ONNX Ops]: <https://onnx.ai/onnx/operators/index.html>
 - IEEE Standard for Floating-Point Arithmetic - 2008 [IEEE 754]
 - International Standard of the C programming language [ISO C 99]
 - <operator>.md file in *safety-related-profile/sonnx/ops/spec/informal/<operator>*
 - *Outputs*
 - <reviewer name>.md in *safety-related-profile/sonnx/ops/spec/informal/<operator>/reviews*
 - **Objective:** verification of the **compliance** of the informal specification with
 - The informal specification guidelines
 - [ONNX Ops]
 - [IEEE 754] for floating-point calculus
 - [ISO C 99] for integer calculus

SONNX V&V: Operator informal specification (cont'd)

- Informal specification validation (cont'd)
 - **Tests**
 - *Inputs and applicable documents*
 - Informal specification in *safety-related-profile/sonnx/ops/spec/informal/<operator>/<operator>.md*
 - Guidelines for Hypothesis based testing
 - *Outputs*
 - Hypothesis and pytest based test programs in *safety-related-profile/sonnx/ops/spec/informal/<operator>/tests/hypothesis*
 - **Objective** verification that the *informal specification* of an operator and *the ONNX runtime implementation* of that operator **are compliant**.
 - **Activity:** development of hypothesis / pytest based test cases from:
 - (Input test patterns) The operator input constraints defined in the informal specification
 - (Test oracle) The operator output values computation also defined in the informal specification

SONNX V&V: Reference test input values

- Reference test input values (for testing the implementation of an operator)
 - *These values are those produced during the informal specification test activity*
- Review of the reference test input values generation
 - **Inputs and applicable documents**
 - Hypothesis and pytest based test programs in `safety-related-profile/sonnx/ops/spec/informal/<operator>/tests/hypothesis`
 - Informal specification in `safety-related-profile/sonnx/ops/spec/informal/<operator>/<operator>.md`
 - Guidelines for Hypothesis based testing
 - **Outputs**
 - `test_<reviewer name>.md` in `safety-related-profile/sonnx/ops/spec/informal/<operator>/reviews`
 - **Objectives:** verification of the **compliance** reference test input values with:
 - The informal specification
 - Guidelines for Hypothesis based testing

SONNX: Operator Why3 formal specifications

- Formal specification development
 - ***Inputs and applicable documents***
 - Informal specification in `safety-related-profile/sonnx/ops/spec/informal/<operator>`
 - Common why3 modules in `safety-related-profile/sonnx/ops/spec/formal/common`
 - Formal specification guidelines in `safety-related-profile/sonnx/ops/docs/guidelines`
 - ***Outputs***
 - ***Abstract*** formal specification
 - “***C-aware***” ***concrete*** formal specification
 - ***Objective:*** enable the generation of trusted reference C code
 - ***Activities***
 - Abstract specification: formalize the informal specification in why3 as directly as possible
 - Develop the concrete specification in such a way that:
 - Its proof against the abstract specification is achievable
 - The extraction of trusted reference C code is feasible

SONNX V&V: Verification of the formal specifications

- Review of the formal specifications
 - ***Inputs and applicable documents***
 - Informal specification in `safety-related-profile/sonnx/ops/spec/informal/<operator>`
 - Abstract formal specification in `safety-related-profile/sonnx/ops/spec/formal/<operator>`
 - “C-aware” concrete formal specification in `safety-related-profile/sonnx/ops/spec/formal/<operator>`
 - Common why3 modules in `safety-related-profile/sonnx/ops/spec/formal/common`
 - Formal specification guidelines in `safety-related-profile/sonnx/ops/docs/guidelines`
 - ***Outputs***
 - `<reviewer name>.md` in `safety-related-profile/sonnx/ops/spec/formal/<operator>/reviews`
 - ***Objective:*** verification of the ***compliance*** of the formal specifications with
 - The Informal specification
 - The formal specification guidelines

SONNX V&V: Verification of the formal specifications

- Proof of the formal specifications
 - **Inputs**
 - Abstract formal specification in `safety-related-profile/sonnx/ops/spec/formal<operator>`
 - Concrete formal specification in `safety-related-profile/sonnx/ops/spec/formal<operator>`
 - Common why3 modules in `safety-related-profile/sonnx/ops/spec/formal/common`
 - **Outputs**
 - Proof report
 - **Objective:** achieve 100% proof rate
 - Note: this might lead iterate on the formal specification, in order to:
 - Express some aspects differently
 - To introduce / update “proof-oriented” construct like assert, invariants; etc

SONNX: Trusted Reference C implementation

- Extraction the trusted reference implementation
 - **Inputs**
 - Concrete formal specification in *safety-related-profile/sonnx/ops/spec/formal/<operator>*
 - Concrete Common why3 modules in *safety-related-profile/sonnx/ops/spec/formal/common*
 - **Outputs**
 - Set of .c files
 - **Objective:** extract the reference implementations C files from the concrete formal specification
 - **Activity:** apply the why3 extract command

SONNX V&V: Graph

- TO DO.

SONNX V&V: Graph exchange format and data storage

- TO DO