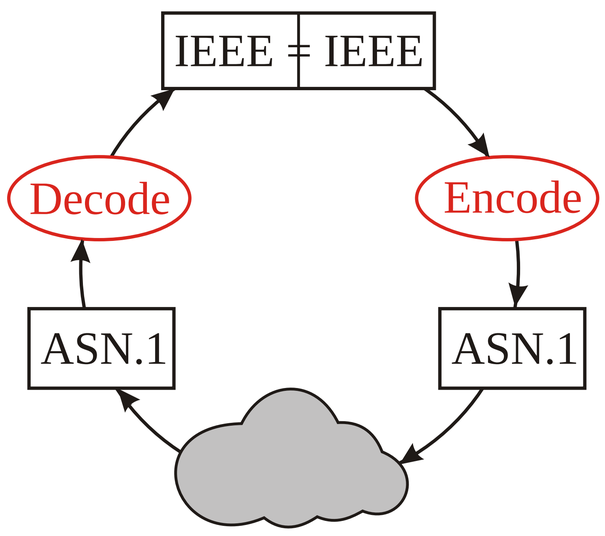
Formal verification of floating-point number conversion between ASN.1 BER and IEEE 754 binary encodings

Ilia Zaichuk
Taras Shevchenko National University of Kyiv, Digamma.ai
zoickx@knu.ua

Problem

ASN.1 encoding is widely used for data transfer between various computing systems. Such data may contain floating-point numbers. The most common representation of floating-point numbers is the IEEE 754 standard. ASN.1, however, does not rely on IEEE for that task. Conversion between the two floating-point representations is error-prone in most ASN.1 protocol implementations. In this project, we formalize such conversion and prove its correctness using the Coq proof assistant.



Scope

Both standards define more than one encoding scheme, but for the purposes of this project we focused on most commonly used practical variants of both.

- Only binary IEEE/ASN formats are considered
- Only short-form ASN is allowed
- NaN payload is lost (could not be represented in ASN.1)
- C wrappers for unit testing and integration were provided only for “binary32” and “binary64” IEEE formats.

Abstract Representation

IEEE 754

SIGN	SIGNIFICAND	EXPONENT

- Radix is 2
- Practical lengths: 32, 64, 128

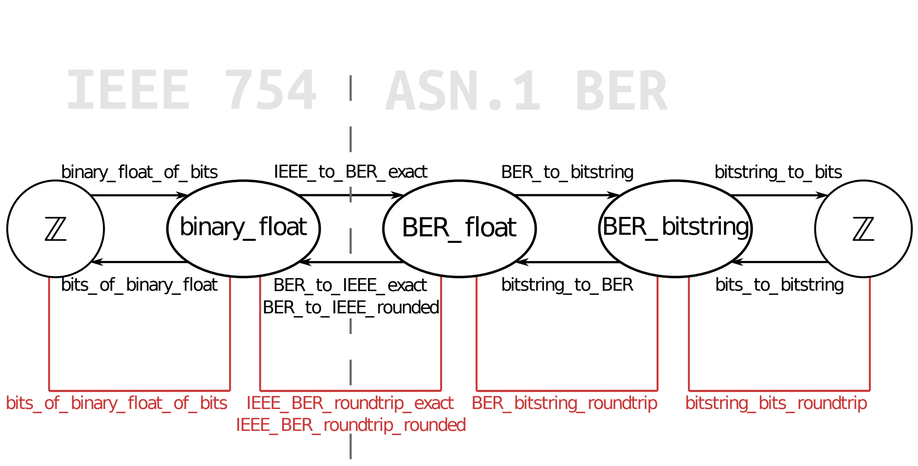
ASN.1

SIGN	SIGNIFICAND	EXPONENT	RADIX

- Radix can be 2, 4, 8, 16
- Octet length < 128

$$\texttt{SIGN*SIGNIFICAND*2}^{\texttt{EXPONENT}} = \texttt{SIGN*SIGNIFICAND*RADIX}^{\texttt{EXPONENT}}$$

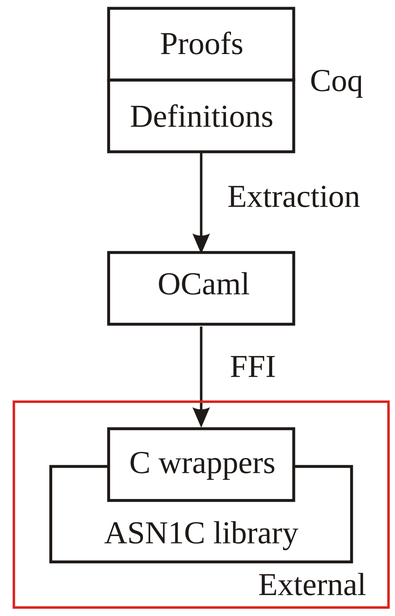
The Proof



- Reference C implementation: ~200 LOC
- Coq spec: 1,488 LOC
- Coq proofs*: 822 LOC
- OCaml test suite: 141 LOC

* Some trivial arithmetic lemmas were admitted

Architecture



We defined and prove in Coq an “executable specification” of encoder and decoder, which is extracted to OCaml. Finally, C wrappers using FFI will be used to link it as a drop-in replacement for FP encoding/decoding functionality in the popular open-source ASN.1 library.

Bit Encoding

IEEE 754

A serizalized float in IEEE 754 format consists of three parts of given predetermined bit-lengths:

Sign	Exponent	Significand
1	8	23

ASN.1

A float in ASN.1 from is noticeably more complicated:

Identifier	Length	Aux	Exponent length	Exponent	Significand
8	8	8	8	...	...

The "Aux" octet carries information about these 5 parameters:
- Encoding type (binary/decimal)
- Sign
- Radix
- Scaling factor
- Exponent length identifier

Main Theorem in Coq

Top-level theorem:

```
Theorem main_roundtrip (scaled : bool) (f : IEEE_float):
  is_Some_b (BER_of_IEEE_exact f) = true ->
  bool_het_inverse
    (option IEEE_float) BER_float IEEE_float
    BER_of_IEEE_exact
    IEEE_of_BER_exact
    float_eqb_nan_t
    f
  = Some true.
```

Heterogeneous inverse identity:

```
Definition bool_het_inverse
           (A1 B A2 : Type)
           (f: A1 -> B)
           (b: B -> A2)
           (e: A1 -> A2 -> bool)
           (x: A1)
: Prop :=
  e x (b (f x)) = true.
```

Work in progress, to be extended to full ASN.1.
Sources: https://github.com/digamma-ai/asn1fpcoq