A Formalization of the C99 Standard in HOL, Isabelle and Coq

Robbert Krebbers Freek Wiedijk

Institute for Computing and Information Sciences, Radboud University Nijmegen, The Netherlands

The C99 standard

The official description issued by ANSI and ISO:

- Written in English
- No mathematically precise formalism
- Incomplete and ambiguous

The Formalin project

- May 2011 to May 2015
- `http://ch2o.cs.ru.nl/`
- Create a formalization of the **complete** C99 standard
- In the theorem provers HOL4, Isabelle/HOL and Coq
- Which follow the standard closely
- All derived from a common master formalization (e.g. in Ott)

Features

- C preprocessor
- C standard library
- Floating point arithmetic
- Casts
- Non-determinism
- Sequence points
- Alignment requirements
- Non-local control flow (`goto`, `setjmp`/`longjmp`, signal handling)
- `volatile`, `restrict` and `const` variables
- Programs in a 'freestanding environment'

Purposes

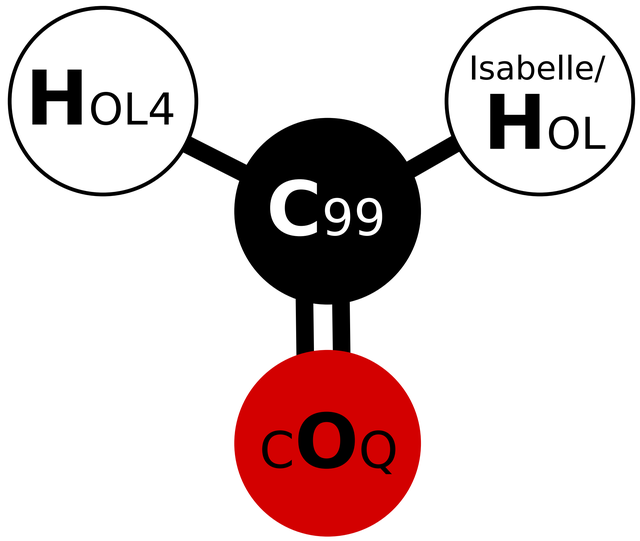
- Utterly precise version of the standard. Useful for compiler writers and programmers
- Validate correctness of formal versions of subsets of C (e.g. Compcert) with respect to the whole standard
- Verify correctness of verification conditions generated by tools (e.g. VCC or Frama-C)

Related projects

- Michael Norrish. C and C++ semantics (L4.verified)
- Xavier Leroy *et al.* Verified C compiler in Coq (Compcert)
- Chucky Ellison and Grigore Rosu. Executable C semantics in Maude

The formalizations

- Describe a space `C_semantics` of all possible C semantics with relations between these semantics
- And, a small step semantics, `C99 : C_semantics`



Dissemination

- Open source, under a BSD-style license
- Using MKM tools like those being developed in the MathWiki project

Some subtleties of C

- Undefined behavior due to unknown evaluation order:

```
int i = 0;
i = ++i; // undefined
```

- Overflow of signed integers is undefined:

```
int i = INT_MAX;
return i < i + 1;
// undefined: hence, a compiler is allowed to
// optimize this to return 1
```

  On the other hand, unsigned integer arithmetic is modular
- Undefined behavior due to jumping into a block with a variable length array declaration:

```
goto foo; // undefined
int a[n];
label foo; printf("bar\n");
```

- Freeing memory makes pointers to it indeterminate

```
int *x = malloc(sizeof(int));
free(x);
printf("%p\n", x); // undefined
```

- Contiguously allocated objects

```
int x = 30, y = 31;
int *p = &x + 1, *q = &y;
if (memcmp(&p, &q, sizeof(p)) == 0) {
  printf("%d\n", *p);
  // the standard is unclear whether this is
  // defined (see Defect report #260).
}
```

References

- International Organization for Standardization. *ISO/IEC 9899:1999: Programming languages – C*. ISO Working Group 14, 1999.
- Freek Wiedijk. *Formalizing the C99 standard in HOL, Isabelle and Coq*. `http://www.cs.ru.nl/~freek/notes/ch2o.pdf`, 2010.

Research team

Robbert Krebbers

PhD student
RU, The Netherlands

Freek Wiedijk

Project leader
RU, The Netherlands

Herman Geuvers

Promotor
RU, The Netherlands

James McKinna

Advisor
RU, The Netherlands

Erik Poll

Advisor
RU, The Netherlands

Michael Norrish

HOL advisor
NICTA, Australia

Andreas Lochbihler

Isabelle advisor
KIT, Germany

Jean-Christophe Filliâtre

Coq advisor
CNRS, France