DE LA RECHERCHE À L'INDUSTRIE

Real Behavior of Floating Point

SOPRANO | F.Bobot, Bruno Marre, Guillaume Melquiond?

08 Janvier

www.cea.fr

Floating Points Have Some Properties!

Rounding $o(\cdot)$ is increasing !!

User_Rule_4

```
procedure User_Rule_4 (X, Y : Float;
                       Res  : out Boolean)
is
begin
   pragma Assume (X <= Y);
   pragma Assume (Y > 0.0);
   Res := X / Y <= 1.0;
   pragma Assert (Res);
end User_Rule_4;
```

$$\forall x, y, z \in \mathcal{R},\ 0 < y \wedge \mathrm{o}\left(\frac{x}{y}\right) < \mathrm{o}(z) \implies \mathrm{o}(x) \leq \mathrm{o}(z \times y)$$

Let $D, E \subset \mathcal{R}$, $f : D \mapsto E$ and $f^{-1} : E \mapsto D$ such that:

- $\forall x : D,\ f^{-1}(f(x)) = x$
- f increasing

we have:

- $\forall x \in D,\ \mathrm{o}(y) \in E,\ \mathrm{o}(f(x)) < \mathrm{o}(y) \implies \mathrm{o}(x) \leq \mathrm{o}(f^{-1}(\mathrm{o}(y)))$
- $\forall x \in D,\ y \in E,\ \mathrm{o}(f(x)) < \mathrm{o}(f(y)) \implies x < y$

```
procedure User_Rule_7 (X, Y, Z, A : Float;
                       Res         : out Boolean)
begin
   pragma Assume (Z >= 0.0);
   pragma Assume (X >= Y);
   pragma Assume (Y >= Z);
   pragma Assume (X > Z);
   pragma Assume (A >= 1.0);
   Res := (X - Y) / (X - Z) <= A;
   pragma Assert (Res);      -- valid
end User_Rule_7;
```

Let $x, y, z \in \mathcal{R}$, if $0 < y$, and $1 < o\left(\frac{o(x)}{o(y)}\right)$ we have $o(y) < o(x)$

For Rule7:

$$A < o\left(\frac{o(x-y)}{o(x-z)}\right)$$

$$1 < o\left(\frac{o(x-y)}{o(x-z)}\right)$$

$$o(x-z) < o(x-y)$$

$$y < z$$

contradict $(y \geq z)$

Let $x, y, z \in \mathcal{R}$, if $0 < y$, and $2^n < \mathrm{o}\left(\frac{\mathrm{o}(x)}{\mathrm{o}(y)}\right)$ we have $\mathrm{o}(2^n * y) < \mathrm{o}(x)$

```
procedure User_Rule_6 (X, Y, Z, A : Float;
                       Res        : out Boolean)
is
begin
   pragma Assume (Z >= 0.0);
   pragma Assume (X >= Y);
   pragma Assume (Y >= Z);
   pragma Assume (X > Z);
   pragma Assume (A <= 0.0);
   Res := (X - Y) / (X - Z) >= A;
   pragma Assert (Res);     -- valid
end User_Rule_6;
```

By domain and delta propagation (sign)

```
procedure User_Rule_16 (X, Y : Float;
                        Res   : out Boolean)
is
begin
   pragma Assume (X in -7800.0 .. 7800.0);
   pragma Assume (Y in -7800.0 .. 7800.0);
   pragma Assume (X > abs Y);
   Res := Sqrt (X * X - Y * Y) <= X;
   pragma Assert (Res);     --@ASSERT:FAIL
end User_Rule_16;
```

$$
\begin{aligned}
\mathrm{o}\left(\sqrt{\mathrm{o}(x^2) - \mathrm{o}(y^2)}\right) &> x \\
\mathrm{o}(x^2) - \mathrm{o}(y^2) &\geq \mathrm{o}(x^2) \\
\mathrm{o}(x^2) - \mathrm{o}(y^2) &= \mathrm{o}(x^2) \text{ because of } x > \mathrm{abs}(y) \\
\mathrm{o}\left(\sqrt{\mathrm{o}(x^2)}\right) &> x \\
x &> x \text{ if } \mathrm{o}(x^2) \text{ is normalized} \\
\mathrm{o}(x^2) & \text{ is denormalized} \\
x & \text{ the minimum of the remaining values is a solution}
\end{aligned}
$$

```
procedure Incr_By_Const
         (State : in out Float_32;
          X     : range 0 .. 1_000_000)
is
begin
   pragma Assume (X < 1_000_000 and
       State in 0.0 | 10.0 .. Float_32 (X) * 10.0);
   State := State + 10.0;
   pragma Assert
      (State in 10.0 .. Float_32 (X + 1) * 10.0);
end Incr_By_Const;
```

$$
\begin{aligned}
State &\leq \mathrm{o}(Float\_32(X) * 10.0) \\
State &\leq Float\_32(X) * 10.0 \\
State &\leq X * 10.0
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{o}(Float\_32(X+1) * 10.0) &= Float\_32(X+1) * 10.0 \\
&= (X+1) * 10.0 \\
&= X * 10.0 + 10.0
\end{aligned}
$$

$$
\begin{aligned}
State + 10.0 &\leq \mathrm{o}(Float\_32(X+1) * 10.0) < \mathrm{o}(State + 10.0) \\
\mathrm{o}(State + 10.0) &\leq \mathrm{o}(Float\_32(X+1) * 10.0) < \mathrm{o}(State + 10.0)
\end{aligned}
$$

Comment from the file: *In this simple calculation, it can be easily seen by interval reasoning that there is no overflow. But current SMT solvers with our encoding can't solve this.*

```
procedure Polynomial (X : Float)
is
begin
   pragma Assume (X in 0.0 .. 60.0);
   pragma Assert ((((X + 2.0) * X + 3.0) + 4.0)
                  * X + 5.0 in Float'Range);
end Polynomial;
```

Propagation.

Not solved: Angle_Between, Approximate_Inverse_Square_Root