Modern Language Tools and 754R

E. Jason Riedy
UC Berkeley
Current State:

HW Designers
- Can implement anything!
- No feedback on trade-offs.

SW Designers
- Few implementation options.
- No access to features.

Languages
- Not bridging the gap.

What can 754R do?
• Current 754 support is ad-hoc.
  • C99, Fortran 2003 provide different descriptions for same model.
  • Many languages provide **no** semantics.
  • Compilers provide less...

• 754R: Provide a language for languages.
  • **Not LIA:** Describe 754 arithmetic, not all possible arithmetics.
  • **Use modern language tools:** Type theory and inferencing.
Typing from 20k Feet

- Precisions $\approx$ data types
  \[ \text{Prec} = \{ \text{single, double, } \ldots \} \]

- Operations $\approx$ function types
  \[ + \in \{ \tau \times \tau \rightarrow \tau \mid \tau \in \text{Prec} \} \]

- *Expression evaluation rules* $\subset$ *typing rules*

  \[ A, S \vdash x : \alpha, y : \alpha \]

  \[ \frac{}{A, S \vdash x + y : \alpha} \]
Impact on 754R

- Provides a sound base for definitions.
- Could define typing hierarchy and “literal” type for wide expression evaluation.
  - Assists interval arithmetic!
- Models modes and flags:
  $$+ : \mathcal{T} \times \mathcal{T} \times \text{modes} \rightarrow \mathcal{T} \times \text{flags}$$
- Possible problems:
  - Comparison operators and wide eval?
  - Compiler support?