A computational science agenda for programming language research

Dominic Orchard and Andrew Rice

http://dorchard.co.uk/science

International Conference on Computational Science, 2014
Better Software
Better Research

Software Sustainability Institute (www.software.ac.uk)
Better Languages
Better Software
Better Research
The two complexities

Inherent complexity
Inadequately supported

Accidental complexity
Too easy to introduce
Current programming approaches

Abstract model  Solution strategy  Prediction calculation
Case study - heat equation

Abstract model

\[
\frac{\partial \phi}{\partial t} = \alpha \frac{\partial^2 \phi}{\partial x^2}
\]

Solution strategy

\[
\phi_x^t = \phi_x^{t-1} + \frac{\alpha \Delta t}{\Delta x^2} (\phi_{x+1}^{t-1} - 2\phi_x^{t-1} + \phi_{x-1}^{t-1})
\]
Case study - heat equation

Prediction calculation

```fortran
9  real :: h(1:nx), h_old(1:nx)  ! heat function (discretised in space)
10  h   = 0  ! initialise as cold
11  h(1) = 1  ! with one hot end
12  do t = 1, nt
13     h_old = h
14     forall (x = 2:(nx - 1))
15        h(x) = h_old(x) + r*(h_old(x-1) - 2*h_old(x) + h_old(x+1))
16  end do
```

Accidental complexity in prediction

• Invalid predictions: who’s to blame?
  Invalid/incomplete model?
  Buggy implementation?
  Both?

• Verification (program correctness) confused with validation (correct model)
Hypothesis/model

Prediction

Experiment

Analysis

Reproduction

complicated relationship

Computer programs
Accidental complexity in reproduction

• Replication/repetition is good
  Some progress: publishing code, Open Science

• but not the final solution…
  ‣ shares bugs and over approximations
  ‣ abstract model hidden/lost
  ‣ code hard to understand
Abstract model  Prediction strategy  Prediction calculation
Problems…

• Reproduction
• Understandability
• Verification
• Utilise new hardware; scalability
Roadmap

1. Computer science engagement with scientists
   • Understand needs
   • Study applications/programming patterns
Results from programming language research

- Advanced type systems for
  - specification/verification
  - abstraction

- Architecture-independent programs

- Automated test generation suites

- Better control of side effects
Roadmap

1. Computer science engagement with scientists
2. New systems for abstraction and specification
Future programming approaches

Abstract model  Solution strategy  Prediction calculation

Automated Verification  Validation
Case study - heat equation

-- Specification of heat equation PDE
spec h = (d h T) === (constant ?alpha * d2 h X) 'withDomain' (X :. T :. Nil)

-- Implementation using a recurrence relation
approx h’ (x, t)
| x == 0    = 1
| x == ?nx  = 0
| t == 0    = 0
| otherwise = h’ (x, t-1) +
             r * (h’ (x+1, t-1) - 2 * h’ (x, t-1) + h’ (x-1, t-1))
                where r = ?alpha * (?dt / (?dx * ?dx))

experiment = let ?dx = 0.05
     ?dt = 0.05
     ?nx = 100
     ?nt = 100
     ?alpha = 0.006
     in verifyModel Euler spec approx
Roadmap

1. Computer science engagement with scientists
2. New systems for abstraction and specification
3. Evolutionary approach
Evolutionary approach

CamFort

Past
- Refactoring tools
- Language extensions
- Test generation tools

PDESpec

Present
- Leverage advanced features
- Libraries/DSLs

Future
- New languages
Roadmap

1. Computer science engagement with scientists
2. New systems for abstraction and specification
3. Evolutionary approach
Better Languages
Better Software
Better Research

Thanks!

http://dorchard.co.uk/science