Separation Logic

Introduction

Arthur Charguéraud

February 2014
Motivation for Separation Logic

Separation Logic: a technique for modular specification and verification of arbitrarily-complex imperative programs.

Covered in this course:

- theory of Separation Logic,
- writing of specifications in practice,
- verification proofs in Separation Logic.

Not covered:

- Automated analyses based on Separation Logic.
Essence of Separation Logic

Linearity:
- from theorists: linear logic, linear types
- from practitioners: region, ownership, capabilities

Local reasoning:
- to isolate the relevant part of the heap,
- to allow for modular verification.
Origins of Separation Logic

- John Reynolds (2000)
  - Intuitionistic Reasoning about Shared Mutable Data Structure
  - —building on ideas from Burstall (1972).

- John Reynolds, Peter O’Hearn, Hongseok Yang (2001)
  - Local reasoning about programs that alter data structures

  - Separation Logic: A logic for shared mutable data structure.
## Applications of Separation Logic

<table>
<thead>
<tr>
<th>Language</th>
<th>Authors</th>
<th>Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micro-controller</td>
<td>Klein et al</td>
<td>NICTA</td>
</tr>
<tr>
<td>Assembly language</td>
<td>Chlipala et al</td>
<td>MIT</td>
</tr>
<tr>
<td>Operating system</td>
<td>Shao et al</td>
<td>Yale</td>
</tr>
<tr>
<td>C (drivers)</td>
<td>Yang et al</td>
<td>Oxford</td>
</tr>
<tr>
<td>C-light</td>
<td>Appel et al</td>
<td>Princeton</td>
</tr>
<tr>
<td>C11 (concurrent)</td>
<td>Vafeiadiis, Parkinson et al</td>
<td>MPI and MSR</td>
</tr>
<tr>
<td>ML</td>
<td>Morisset et al</td>
<td>Harvard</td>
</tr>
<tr>
<td>Java</td>
<td>Parkinson et al</td>
<td>MSR and Cambridge</td>
</tr>
<tr>
<td>Java</td>
<td>Jacobs et al</td>
<td>Leuven</td>
</tr>
<tr>
<td>Javascript</td>
<td>Gardner et al</td>
<td>Imperial College</td>
</tr>
<tr>
<td>Caml</td>
<td>Charguéraud</td>
<td>Inria</td>
</tr>
</tbody>
</table>

→ For more projects, see Peter O’Hearn’s webpage on Separation Logic:

http://www0.cs.ucl.ac.uk/staff/p.ohearn/SeparationLogic/Separation.Logic/SL.Home.html
This course on Separation Logic

Compared with other Separation Logic courses:

- targets a clean ML language (not C)
- presents definitions used in practice (not $\rightarrow$, not $\wedge$)
- targets higher-order logic (not first-order)
- supports higher-order functions
Main difference with the Why approach

Compared with the previous course based on Why:

- applies to un-annotated programs
- proof automation is not a concern
- many similar rules, but different interpretations
- addresses the aliasing problem
- allows for local reasoning.