Separation Logic
Introduction

Arthur Charguéraud

January 28th, 2016

Motivation for Separation Logic

Specification and verification of imperative programs in a **modular** way, to scale up to large and complex programs.

$$\text{Separation Logic} + \text{Interactive proofs} = \text{No limits}$$

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
- John Reynolds (2002)
  - Separation Logic: A logic for shared mutable data structure.

Adopters of Separation Logic

Micro-controller	Klein et al	NICTA
Assembly language	Chlipala et al	MIT
Operating system	Shao et al	Yale
C (drivers)	Yang et al	Oxford
C-light	Appel et al	Princeton
C11 (concurrent)	Vafeiadis, Parkinson et al	MPI and MSR
Java	Parkinson et al	MSR and Cambridge
Java	Jacobs et al	Leuven
Javascript	Gardner et al	Imperial College
ML	Morisset et al	Harvard
OCaml	Charguéraud	Inria
...		

More on Peter O'Hearn's webpage on Separation Logic:

`http://www0.cs.ucl.ac.uk/staff/p.ohearn/SeparationLogic/Separation_Logic/SL_Home.html`

Overview

Program verification using Separation Logic supports:

- Tree-shaped structures
- Structures with sharing
- First-class functions
- Polymorphism
- Modularity
- Abstraction
- Proof assistants