Scopes Describe Frames
A Uniform Model for Memory Layout in Dynamic Semantics
Casper Bach Poulsen¹, Pierre Néron², Andrew Tolmach³, Eelco Visser¹

Problem with Previous Approaches

Static Name Binding
Handed in a number of ways in semantic specifications:
- Lexical scope: type substitution, type environments
- Stateful references: reference types, store typings
- Structured memory: class tables
Lack of uniform model

Dynamic Memory
Handed in a number of ways in semantic specifications:
- Lexical scope: substitution, environments, de Bruijn, HOAS
- Stateful references: Mutable stores, heaps
- Structured memory: Mutable values for records, objects
Lack of uniform model

Our Solution

Scope Graphs [ESOP’15]
Nodes of scope graphs represent three basic notions derived from the program abstract syntax tree:
- Scopes (●) and edges (→) between scopes
- Declarations (        )
- References (         )
- Static resolution paths (             )
Uniform model

Frames and Heaps
We propose frames as a language-independent model for dynamic memory. The model is based on these notions:
- Frames (        ) and links (          ) between frames
- Heap: a frame graph
- Dynamic lookup (         ): static resolution path interpreted relative to the “current” frame
Uniform model

Static Binding matches Dynamic Behavior

Well-Bound Frame
Frame slots and links correspond to scope declarations and edges

Well-Typed Frame
Types of values in frame slots match types of corresponding scope declarations

Good Heap
All frames are well-bound and well-typed

Memory Invariants

Type Soundness Principle
Evaluation preserves good heap property

Verification

Language-Independent

Language-Specific

Specification Architecture

Garbage Collection

Language-Independent Lemmas

Language-Independent

Dynamic Static

Scopes

Well-Boundness

Well-Typedness

Frames

Dynamic Semantics

Type Soundness

Unreferenced

Lemma (Safe Removal).
For a good heap X = A ∪ B, if nothing in B is referenced from A, then A is a good heap (B can be safely garbage collected).