Section 2.6 Shape Analysis

Reporter: Xie Li

August 30, 2021

Syntax of Pointer Language

Selectors: a set of selector names are given (What can be selectors?)

$$sel \in \mathbf{Sel}$$

A set of pointer expressions.

$$p \in \mathbf{PExp}$$

where $p ::= x \mid x.sel$ The extended WHILE-language:

Arithmetic a is extended to pointer expressions, but not pointer arithmetic.
op_r allows equality test for pointers.

Example

Example

Example

Structural Operational Semantics Basic Definitions

- An infinite set of locations $\mathbf{Loc}$: $\xi \in \mathbf{Loc}$.
- A set of states $\mathbf{State}$: $\sigma \in \mathbf{State} = \mathbf{Var}_* \to (\mathbf{Z} + \mathbf{Loc} + \{\diamond\})$
- A set of heaps $\mathbf{Heap}$: $\mathscr{H} \in \mathbf{Heap} = (\mathbf{Loc} \times \mathbf{Sel}) \to_{fin} (\mathbf{Z} + \mathbf{Loc} + \{\diamond\})$

where the partial means not all selector fields need to be defined.
The semantic of pointer arithmetic is given by:

Example

- oval nodes: heap cells
- ξ_is': locations
- labelled edges: heap
- unlabelled edges: state

Semantics of Expressions

Extend the old semantic to store and heap:

Semantics of Statements

- $\langle [x := a]^{\ell}, \sigma, \mathcal{H} \rangle$
- $\langle [x.sel := a]^{\ell}, \sigma, \mathcal{H} \rangle$
- `malloc`: allocation for pointers.

Semantic of Statements

For `malloc`: a limited reused strategy is used.

- x can be reused: $[\texttt{malloc}x]^1; [x := \texttt{nil}]^2; [\texttt{malloc}x]^3$
- x cannot be reused although unreachable: $[\texttt{malloc}x]^1; [x.cdr := \texttt{nil}]^2; [x := \texttt{nil}]^3; [\texttt{malloc}x]^4$

Shape Graphs: Abstraction of the Memory Configurations

Definition

Shape Graph A shape graph is a triplet $(\mathsf{S}, \mathsf{H}, \mathtt{is})$, where

- Abstract state S is a map from variables to *abstract locations*:

$$\mathsf{S} \in \textbf{AState} = \mathscr{P}(\textbf{Var}_* \times \textbf{ALoc})$$

- Abstract heap H is set specifies the links between abstract locations :

$$\mathsf{H} \in \textbf{AHeap} = \mathscr{P}(\textbf{ALoc} \times \textbf{Sel} \times \textbf{ALoc})$$

- A set of abstract locations that are shared:

$$\mathtt{is} \in \textbf{IsShared} = \mathscr{P}(\textbf{ALoc})$$

Abstract Locations

The abstract locations have the form n_X where X is a subset of the variables of $\mathbf{Var}_*$:

$$\mathbf{ALoc} = \{n_X \mid X \subseteq \mathbf{Var}_*\}$$

Finiteness: the set $\mathbf{Var}_*$ is finite.

- Intuitively, n_X is the abstraction of the location $\sigma(x)$ of pointer variables $x \in X$
- We shall enforce:
  Invariant 1: If n_X and n_Y occur in the same shape graph, then either $X = Y$ or $X \cap Y = \emptyset$.
- We use *abstract summary location* $n_\emptyset$ to represent all locations that cannot be reached directly from the state without using the heap.

Example of Abstract Locations

Abstract States and Heaps

- Abstract state: Abstract state S is a map from variables to *abstract locations*:

$$\mathsf{S} \in \mathbf{AState} = \mathscr{P}(\mathbf{Var}_* \times \mathbf{ALoc})$$

Invariant 2. If x is mapped to n_X by the abstract state then $x \in X$.

Define the set $ALoc(\mathsf{S}) = \{n_X \mid \exists x : (x, n_X) \in \mathsf{S}\}$ to be the abstract locations occuring in S.

- Abstract heap: heap H is set specifies the links between abstract locations :

$$\mathsf{H} \in \mathbf{AHeap} = \mathscr{P}(\mathbf{ALoc} \times \mathbf{Sel} \times \mathbf{ALoc})$$

Intuitively, if $\mathscr{H}(\xi_1, sel) = \xi_2$ and ξ_1, ξ_2 are represented by n_V, n_V resp., then $(n_V, sel, n_W) \in \mathsf{H}$.

Invariant 3. Whenever (n_V, sel, n_W) and $(n_V, sel, n_{W'})$ are in the abstract heap, then either $V = \emptyset$ or $W = W'$.

Both abstract state and abstract heap are changing along the execution.

Example of Abstract States and Heaps

Sharing Information

A set of abstract locations that are shared due to pointers in the *heap*:

$$\mathtt{is} \in \mathbf{IsShared} = \mathscr{P}(\mathbf{ALoc})$$

Sharing Information

With above observations we have following invariant:

And the connection of `is` to the abstract heap H:

Sharing information clearly gives extra information: $n_\emptyset$

Use of Shape Graph

The Complete Lattice of Shape Graph

$$\begin{aligned} \mathsf{S} &\in \mathbf{AState} = \mathscr{P}(\mathbf{Var}_* \times \mathbf{ALoc}) \\ \mathsf{H} &\in \mathbf{AHeap} = \mathscr{P}(\mathbf{ALoc} \times \mathbf{Sel} \times \mathbf{ALoc}) \\ \mathtt{is} &\in \mathbf{IsShared} = \mathscr{P}(\mathbf{ALoc}) \end{aligned}$$

$\mathbf{ALoc} = \{n_Z \mid Z \subseteq \mathbf{Var}_*\}$.
A shape graph $(\mathsf{S}, \mathsf{H}, \mathtt{is})$ is compatible if it satisfies **Invariant 1.- 5.**, the set of compatible shape graphs is denoted:

$$\mathbf{SG} = \{(\mathsf{S}, \mathsf{H}, \mathtt{is}) \mid \mathsf{S}, \mathsf{H}, \mathtt{is} \textit{ iscompatible}\}$$

The analysis will operate over on $\mathscr{P}(\mathbf{SG})$. Since it is a power set, it is trivially a complete lattice over union and subset relation. Due to the finiteness of $\mathbf{Var}_*$ is finite.

The Analysis: Basic Framework

An instance of monotone framework: let $\mathscr{P}(\mathbf{SG})$ be the complete lattice of properties.
For label consistent program S_*, we obtain a set of equations by

In the following, the transfer functions over different statements will be developed.
The transfer function associate with label l: $f_\ell^{\mathsf{SA}} : \mathscr{P}(\mathbf{SG}) \to \mathscr{P}(\mathbf{SG})$.

$$f_\ell^{\mathsf{SA}}(SG) = \bigcup \{\phi_\ell^{\mathsf{SA}}((\mathtt{S},\mathtt{H},\mathtt{is})) \mid (\mathtt{S},\mathtt{H},\mathtt{is}) \in SG\}$$

$\phi_\ell^{\mathsf{SA}} : \mathbf{SG} \to \mathscr{P}(\mathbf{SG})$.

Transfer Functions

- For $[b]^\ell$ and $[\mathsf{skip}]^\ell$: $\phi_\ell^{\mathsf{SA}}((\mathtt{S},\mathtt{H},\mathtt{is})) = \{(\mathtt{S},\mathtt{H},\mathtt{is})\}$
- For $[x := a]^\ell$ where a is of the form $n, a_1\, op_a\, a_2$ or $\mathtt{nil}$:
  - Renaming of abstract locations to exclude x: $k_x(n_Z) = n_{Z\setminus\{x\}}$
  - Function ϕ_ℓ^{SA} is given by

Example

Transfer Functions

- For $[x := y]^\ell$: - $x = y$. - $x \neq y$
- Based on previous $(\mathtt{S}', \mathtt{H}', \mathtt{is}')$:

Here (y', n_Y) means n_Y must can be visit by a pointer variable by state.

Example

Transfer Functions

- For $[x := y.sel]^{\ell}$, we can regard it as an equivalent sequence:

$$[t := y.sel]^{l_1}; [x := t]^{l_2}; [t := nil]^{l_3}$$

and

$$f_{\ell}^{\mathsf{SA}} = f_{\ell_3}^{\mathsf{SA} \cdot f_{\ell_2}^{\mathsf{SA} \cdot f_{\ell_3}^{\mathsf{SA}}}}$$