Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D.
Fabian Huch

WS 2022/23
Jan 31

Semantics of Programming Languages

Exercise Sheet 13

Exercise 13.1 Inverse Analysis

Consider a simple sign analysis based on this abstract domain:

datatype *sign* = *None* | *Neg* | *Pos0* | *Any*

fun γ :: *"sign ⇒ val set"* **where**
"γ None = {}" |
"γ Neg = {i. i < 0}" |
"γ Pos0 = {i. i ≥ 0}" |
"γ Any = UNIV"

Define inverse analyses for "+" and "<" and prove the required correctness properties:

fun *inv_plus'* :: *"sign ⇒ sign ⇒ sign ⇒ sign * sign"*
lemma
"⟦ inv_plus' a a1 a2 = (a1',a2'); i1 ∈ γ a1; i2 ∈ γ a2; i1+i2 ∈ γ a ⟧
⟹ i1 ∈ γ a1' ∧ i2 ∈ γ a2' "
fun *inv_less'* :: *"bool ⇒ sign ⇒ sign ⇒ sign * sign"*
lemma
"⟦ inv_less' bv a1 a2 = (a1',a2'); i1 ∈ γ a1; i2 ∈ γ a2; (i1<i2) = bv ⟧
⟹ i1 ∈ γ a1' ∧ i2 ∈ γ a2'"

Homework 13.1 Inverse Analysis for the Extended Reals

Submission until Monday, Feb 6, 23:59pm.

Extend the abstract interpreter from the last sheet with an inverse analysis. We skip the technical part, so you only need to define the inverse operations:

fun *inv_plus'* :: *"bounds ⇒ bounds ⇒ bounds ⇒ (bounds * bounds)"*
fun *inv_less'* :: *"bool option ⇒ bounds ⇒ bounds ⇒ (bounds * bounds)"*

They should be as precise as possible, i.e., the result minimal. For example:

value *"inv_plus' (B {NaN}) (B {∞⁺}) (B {∞⁻, Real}) = (B {∞⁺}, B {∞⁻})"*
value *"inv_plus' (B {∞⁺,∞⁻}) (B {∞⁺, NaN}) (B {∞⁻, Real, ∞⁺}) = (B {∞⁺}, B {∞⁺, Real})"*
value *"inv_less' (Some True) (B {∞⁺, ∞⁻}) (B {∞⁺}) = (B {∞⁻}, B {∞⁺})"*

Prove them sound!

lemma *inv_plus′_sound*:
  assumes "*inv_plus′ (B A) (B A1) (B A2) = (B A1′, B A2′)*"
    and "*i1 ∈ γ_bounds (B A1)*"
    and "*i2 ∈ γ_bounds (B A2)*"
    and "(*case (i1,i2) of*
        (*None*, _) ⇒ *None ∈ γ_bounds (B A)*
      | (_, *None*) ⇒ *None ∈ γ_bounds (B A)*
      | (*Some PInfty*, *Some MInfty*) ⇒ *None ∈ γ_bounds (B A)*
      | (*Some MInfty*, *Some PInfty*) ⇒ *None ∈ γ_bounds (B A)*
      | (*Some i1*, *Some i2*) ⇒ *Some (i1+i2) ∈ γ_bounds (B A)*)"
  shows "*i1 ∈ γ_bounds (B A1′) ∧ i2 ∈ γ_bounds (B A2′)*"

lemma *inv_less′_sound*:
  assumes "*inv_less′ bv (B A1) (B A2) = (B A1′,B A2′)*"
    and "*i1 ∈ γ_bounds (B A1)*"
    and "*i2 ∈ γ_bounds (B A2)*"
    and "(*case (i1,i2) of*
        (*None*, _) ⇒ *None = bv*
      | (_, *None*) ⇒ *None = bv*
      | (*Some i1*, *Some i2*) ⇒ *Some (i1<i2) = bv*)"
  shows "*i1 ∈ γ_bounds (B A1′) ∧ i2 ∈ γ_bounds (B A2′)*"