Program Analysis and Synthesis
HW 2

Igor Zarivach 306831835
Jalil Moraney 302872833

February 10, 2015

1 Domain

The abstract domain is: $(L, \leq_i, \vee_i, \wedge_i, \perp_i, [-\infty, \infty])$m , where we define the following as well:

- $Z_\infty = Z \cup \{-\infty, \infty\}$
- $L = \{[x, y] \mid x, y \in Z_\infty, y \geq_\infty x\} \cup \perp_i$.
- The relation $\leq_\infty$ for Z_∞: $x \leq_\infty y \iff (x, y \in Z,\, x \leq y) \vee (x = -\infty) \vee (y = \infty)$.
- $\forall z \in Z_\infty \setminus \{-\infty\} :\ \infty + z = \infty, \infty - z = \infty, z + \infty = \infty, z - \infty = \infty.$
- $\forall z \in Z_\infty \setminus \{\infty\} :\ (-\infty) + z = -\infty, (-\infty) - z = -\infty, z + (-\infty) = -\infty, z - (-\infty) = \infty.$
- $\forall z \in Z_\infty :\ z * 0 = 0 * z = 0.$
- $\forall z \in Z_\infty \setminus \{-\infty\} \cup \{z \in Z \,|\, z < 0\} :\ \infty * z = \infty, z * \infty = \infty.$
- $\forall z \in Z_\infty \setminus \{\infty\} \cup \{z \in Z \,|\, z > 0\} :\ \infty * z = -\infty, z * \infty = -\infty.$
- $\forall z \in Z_\infty \setminus \{-\infty\} \cup \{z \in Z \,|\, z < 0\} :\ (-\infty) * z = -\infty, z * (-\infty) = -\infty.$
- $\forall z \in Z_\infty \setminus \{\infty\} \cup \{z \in Z \,|\, z > 0\} :\ (-\infty) * z = -\infty, z * (-\infty) = \infty.$
- $\forall z \in \{-\infty, \infty\}, \forall x \in \{z \in Z \,|\, z > 0\} :\ \frac{z}{x} = z.$
- $\forall z \in \{-\infty, \infty\}, \forall x \in \{z \in Z \,|\, z < 0\} :\ \frac{z}{x} = -z.$
- $\forall x \in Z, \forall z \in \{-\infty, \infty\} :\ \frac{x}{z} = 0.$
- We also note that the only thing we can say about $\frac{\infty}{\infty}$ and $\frac{-\infty}{-\infty}$ is that they are positive. i.e., greater or equal to 1. Thus to simple the definition of the transformer we define the $\frac{\infty}{\infty} = \frac{-\infty}{-\infty} = 1$.

- We also note that the only thing we can say about $\frac{\infty}{-\infty}$ and $\frac{-\infty}{\infty}$ is that they are negative. i.e., less or equal to -1. Thus to simple the definition of the transformer we define the $\frac{-\infty}{\infty} = \frac{\infty}{-\infty} = -1$.
- For a set $S \subseteq Z_\infty$, $min_\infty(S)$ is the minimal number in S according to $\leq_\infty$.
- For a set $S \subseteq Z_\infty$,$max_\infty(S)$ is the maximal number in S according to $\leq_\infty$.
- $[a,b] \leq_i [c,d] \Longleftrightarrow (c \leq_\infty a) \wedge (b \leq_\infty d)$.
- $[a,b] \vee_i [c,d] = [min_\infty(\{a,c\}), max_\infty(\{b,d\})]$.
- $[a,b] \wedge_i [c,d] = [meet(max_\infty(\{a,c\}), min_\infty(\{b,d\}))]$ where $meet(a,b)$ returns $[a,b]$ if $a \leq_\infty b$ and $\perp_i$ otherwise.

1.1 Interval abstraction

α_i : (Label $\rightarrow$ (Var $\rightarrow$ $\wp$(Z))) $\longrightarrow$ (Label $\rightarrow$ (Var $\rightarrow$ L))

$$\alpha_i(C)(Label)x = \begin{cases} [min(C(Label)x), max(C(Label)x)] & C(Label)x \neq \emptyset \\ \perp_i & C(Label)x = \emptyset \end{cases}$$

α_i maps for each program Label and local variable and a set of integers A to interval that contains every $a \in A$.

low and high

Define for local x and abstract state σ

$\sigma(x).low$: $\sigma(x)$=[a,b]$\Rightarrow$ $\sigma(x).low$ = a
$\sigma(x).high$: $\sigma(x)$=[a,b]$\Rightarrow$ $\sigma(x).high$ = b

1.2 Logical Transformations - If

1.2.1 [x>y]

True

$$[\text{if (e1} > \text{e2)}]_{true}(\sigma) = \begin{cases} below & \text{e1 and e2 are constants} \\ below & \text{e1 is local, e2 is constant} \\ below & \text{e1 is constant, e2 is local} \\ below & \text{e1,e2 are locals} \end{cases}$$

x is local, a is constant

$$[\text{if (x} > \text{a)}]_{true}(\sigma) = \begin{cases} \sigma \wedge_i \{x \rightarrow [a+1, \infty] & \sigma(x).high > \text{a} \\ \perp & else \end{cases}$$

x is local, a is constant

$$[\text{if (a} > \text{x)}]_{true}(\sigma) = \begin{cases} \sigma \wedge_i \{x \rightarrow [-\infty, a-1] & \text{a} > \sigma(x).low \\ \perp & else \end{cases}$$

a,b are constants

$$[\text{if (a > b)}]_{true}(\sigma) = \begin{cases} \sigma & \text{a > b} \\ \perp & else \end{cases}$$

x,y are locals

$$[\text{if (x > y)}]_{true}(\sigma) = \begin{cases} \perp & \text{x and y are the same local} \\ \perp & \sigma(x).high \leq \ \sigma(y).low \\ \sigma \wedge_i \{x \rightarrow [\sigma(y).low + 1, \infty], y \rightarrow [-\infty, \sigma(x).high - 1]\} & else \end{cases}$$

Explanation: Integer $n \in$ [if (x > y)]$_{true}(\sigma)(x)$ iff $n \in \sigma(x)$ and $\exists m \in \sigma(y)$, $n > m$.
We have $\sigma(y).low \leq m \leq \sigma(y).low$, so $n \geq \sigma(y).low + 1$, which implies that $n \in [\sigma(y).low + 1, \infty]$.
So n is in the interval $\sigma(x) \wedge_i [\sigma(y).low + 1, \infty]$. The same logic works for y.
We can also check if σ(x).high $\leq \sigma$(y).low by

$$\sigma(\text{x}).\text{high} \leq \sigma(\text{y}).\text{low} \iff \sigma(x) \wedge_i [\sigma(y).low + 1, \infty] = \perp_i$$

So the rule to implement is

$$[\text{if(x>y)}]_{true}(\sigma) = \begin{cases} \perp & \text{x and y are the same local} \\ \perp & \sigma(x) \wedge_i [\sigma(y).low + 1, \infty] = \perp_i \\ \sigma \wedge_i \{x \rightarrow [\sigma(y).low + 1, \infty], & else \\ y \rightarrow [-\infty, \sigma(x).high - 1]\} & \end{cases}$$

False
[if (e1 > e2)]$_{false}(\sigma)$ = [if (e2 $\geq$ e1)]$_{true}(\sigma)$

1.2.2 [x≥y]

True

$$[\text{if (e1} \geq \text{e2)}]_{true}(\sigma) = \begin{cases} below & \text{e1 and e2 are constants} \\ below & \text{e1 is local, e2 is constant} \\ below & \text{e1 is constant, e2 is local} \\ below & \text{e1,e2 are locals} \end{cases}$$

a,b are constants

$$[\text{if (a} \geq \text{b)}]_{true}(\sigma) = \begin{cases} \sigma & \text{a} \geq \text{b} \\ \perp & \text{a < b} \end{cases}$$

x is local, a is constant

$$[\text{if (x} \geq \text{a)}]_{true}(\sigma) = \begin{cases} \sigma \wedge_i \{x \rightarrow [a, \infty] & \sigma(x).high \geq \text{a} \\ \perp & else \end{cases}$$

x is local, a is constant

$$[\text{if (a} \geq \text{x)}]_{true}(\sigma) = \begin{cases} \sigma \wedge_i \{x \to [-\infty, a] & \text{a} \geq \sigma(x).low \\ \bot & else \end{cases}$$

x,y are locals

$$[\text{if (x} \geq \text{y)}]_{true}(\sigma) = \begin{cases} \sigma & \text{x and y are the same local} \\ \bot & \sigma(x).high < \sigma(y).low \\ \sigma \wedge_i \{x \to [\sigma(y).low, \infty], y \to [-\infty, \sigma(x).high]\} & else \end{cases}$$

Agan, using only meet operation

$$[\text{if (x} \geq \text{y)}]_{true}(\sigma) = \begin{cases} \sigma & \text{x and y are the same local} \\ \bot & \sigma(x) \wedge_i [\sigma(y).low, \infty] = \bot_i \\ \sigma \wedge_i \{x \to [\sigma(y).low, \infty], y \to [-\infty, \sigma(x).high]\} & else \end{cases}$$

False

$[\text{if (e1} \geq \text{e2)}]_{false}(\sigma) = [\text{if (e2} > \text{e1)}]_{true}(\sigma)$

1.2.3 [x<y]

True

$[\text{if (e1} < \text{e2)}]_{true}(\sigma) = [\text{if (e2} > \text{e1)}]_{true}(\sigma)$

False

$[\text{if (e1} < \text{e2)}]_{false}(\sigma) = [\text{if (e1} \geq \text{e2)}]_{true}(\sigma)$

1.2.4 [x≤y]

True

$[\text{if (e1} \leq \text{e2)}]_{true}(\sigma) = [\text{if (e2} \geq \text{e1)}]_{true}(\sigma)$

False

$[\text{if (e1} \leq \text{e2)}]_{false}(\sigma) = [\text{if (e1} > \text{e2)}]_{true}(\sigma)$

1.2.5 [x=y]

True

$[\text{if (e1} = \text{e2)}]_{true}(\sigma) = ([\text{if (e1} \geq \text{e2)}]_{true}(\sigma)) \wedge_i ([\text{if (e2} \geq \text{e1)}]_{true}(\sigma))$

False

$[\text{if (e1} = \text{e2)}]_{false}(\sigma) = ([\text{if (e1} \geq \text{e2)}]_{false}(\sigma)) \vee_i ([\text{if (e2} \geq \text{e1)}]_{false}(\sigma)) =$
$= ([\text{if (e2} > \text{e1)}]_{true}(\sigma)) \vee_i ([\text{if (e1} > \text{e2)}]_{true}(\sigma))$

1.2.6 [x≠y]

True
[if (e1 $\neq$ e2)]$_{true}(\sigma)$ =[if (e1 = e2)]$_{false}(\sigma)$=
= ([if (e2 > e1)]$_{true}(\sigma)$)$\vee_i$([if (e1 > e2)]$_{true}(\sigma)$)

False
[if (e1 $\neq$ e2)]$_{false}(\sigma)$ =[if (e1 = e2)]$_{true}(\sigma)$ =
= ([if (e1 $\geq$ e2)]$_{true}(\sigma)$)$\wedge_i$([if (e2 $\geq$ e1)]$_{true}(\sigma)$)

1.3 Logical Transformations - Switch

1.3.1 [lookupswitch(i) { case 2: goto label0; case 7: goto label1; default: goto label2; };]

[lookupswitch(i)
{ case a: goto label0; }] (σ) = [if $(i = a)$]$_{true}$
[lookupswitch(i)
{ default: }] $(\sigma) = \sigma$

TableSwitch The same semantics works for the tableswitch.

1.4 Arithmetic Operations

For simplicity of dontation, we assume that the operation is at label l, and all of the replacment in the state $\sigma$of the form $\sigma [w \to a]$ are actually $\sigma [l \to \{w \to a\}]$. i.e, changing only the value w is mapped to in the map of the label l.

1.4.1 [w=z]

$$[w = z](\sigma) = \begin{cases} \sigma [w \to [z, z]] & z \in Z \\ \sigma [w \to \sigma (z)] & z \in L \end{cases}$$

1.4.2 [w=x+y]

$$[w = x + y](\sigma) = \begin{cases} \sigma [w \to [x + y, x + y]] & x, y \in Z \\ \sigma [w \to [\sigma (x) .low + y, \sigma (x) .high + y]] & x \in L, y \in Z \\ \sigma [w \to [x + \sigma (y) .low, x + \sigma (y) .high]] & x \in Z, y \in L \\ \sigma [w \to [\sigma (x) .low + \sigma (y) .low, \sigma (x) .high + \sigma (y) .high]] & x, y \in L \end{cases}$$

1.4.3 [w=x-y]

$$[w = x - y](\sigma) = \begin{cases} \sigma[w \to [x-y, x-y]] & x, y \in Z \\ \sigma[w \to [\sigma(x).low - y, \sigma(x).high - y]] & x \in L, y \in Z \\ \sigma[w \to [x - \sigma(y).low, x - \sigma(y).high]] & x \in Z, y \in L \\ \sigma[w \to [\sigma(x).low - \sigma(y).low, \sigma(x).high - \sigma(y).high]] & x, y \in L \end{cases}$$

1.4.4 [w=x*y]

- if $x, y \in Z$ then:

$$[w = x * y](\sigma) = \sigma[w \to [x * y, x * y]]$$

- if $x \in L, z \in Z$ then:

$$\begin{aligned} [w = x * z](\sigma) &= \sigma[w \to [min_\infty(\{\sigma(x).low * z, \sigma(x).high * z\}), \\ &\quad max_\infty(\{\sigma(x).low * z, \sigma(x).high * z\})]] \end{aligned}$$

.

- if $z \in Z, x \in L$ then $[w = z * x](\sigma) = [w = x * z](\sigma)$.
- if $x, y \in L$,we define:

$$\begin{aligned} lowest &= min_\infty(\{\sigma(x).low * \sigma(y).low, \\ &\quad \sigma(x).high * \sigma(y).low, \\ &\quad \sigma(x).low * \sigma(y).high, \\ &\quad \sigma(x).high * \sigma(y).high\}) \end{aligned}$$

$$\begin{aligned} highest &= max_\infty(\{\sigma(x).low * \sigma(y).low, \\ &\quad \sigma(x).high * \sigma(y).low, \\ &\quad \sigma(x).low * \sigma(y).high, \\ &\quad \sigma(x).high * \sigma(y).high\}) \end{aligned}$$

then $[w = x * y](\sigma) = \sigma[w \to [lowest, highest]]$.

1.4.5 [w=x/y]

- if $x, y \in Z$ then: $[w = x/y](\sigma) = [w = [x, x] / [y, y]](\sigma)$.
- if $x \in Z, y \in L$ then: $[w = x/y](\sigma) = [w = [x, x] / y](\sigma)$.
- if $y \in Z, x \in L$ then: $[w = x/y](\sigma) = [w = x / [y, y]](\sigma)$.

- if $x, y \in L$, $\sigma(y).low \leq 0 \leq \sigma(y).high$ then:

$$[w = x/y](\sigma) = \sigma[w \rightarrow [-\infty, \infty]]$$

- if $x, y \in L$,$\sigma(y).low > 0 \,||\, 0 > \sigma(y).high$, we define:

$$\begin{aligned} lowest &= min_{\infty}(\{\sigma(x).low/\sigma(y).low, \\ &\quad \sigma(x).low/\sigma(y).high, \\ &\quad \sigma(x).high/\sigma(y).low, \\ &\quad \sigma(x).high/\sigma(y).high\}) \end{aligned}$$

$$\begin{aligned} highest &= max_{\infty}(\{\sigma(x).low/\sigma(y).low, \\ &\quad \sigma(x).low/\sigma(y).high, \\ &\quad \sigma(x).high/\sigma(y).low, \\ &\quad \sigma(x).high/\sigma(y).high\}) \end{aligned}$$

  then $[w = x/y](\sigma) = \sigma[w \rightarrow [lowest, highest]]$.

1.4.6 [w=x%y]

- if $y = [-\infty, \infty]$ then $[w = x\%y](\sigma) = \sigma[w \rightarrow [-\infty, \infty]]$.
- if$y = \perp$ then $[w = x\%y](\sigma) = \sigma[w \rightarrow [\perp]]$.
- if $\sigma(y).low \leq 0 \,\&\&\, \sigma(y).high \geq 0$ then $[w = x\%y](\sigma) = \sigma[w \rightarrow [-\infty, \infty]]$.
- if $\sigma(y).low > 0$ then $[w = x\%y](\sigma) = \sigma[w \rightarrow [0, \sigma(y).high - 1]]$.
- if $\sigma(y).high < 0$ then $[w = x\%y](\sigma) = \sigma[w \rightarrow [\sigma(y).high + 1, 0]]$.

1.4.7 $[w = -x]$

same as $[w = [-1, -1] * x]$.