A Live Variable Analysis for Non-higher order Languages based on 0-CFA

Flow Analysis

- Prediction of the possible values of any expression
- Prediction of the possible values of a variable
- Data flow
- Control flow

Higher-order Languages and flow analysis

- proc(x) … (x y) …
- must build control flow and data flow at the same time
- Solution: track closures and their flow through the program

First Approach

- mark each expression with a label

$$\ell \in LAB$$

- modify standard semantics
- find every procedure call
- record the call in a table as a call cache

$$CCache = (LAB \times ENV) \rightarrow \mathrm{Proc}$$

- return the table

- unrealistic!

First Approach

- mark each expression with a label

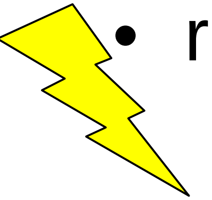
$$\ell \in LAB$$

- modify standard semantics
- find every procedure call
- record the call in a table as a call cache

$$CCache = (LAB \times ENV) \to Proc$$

- return the table

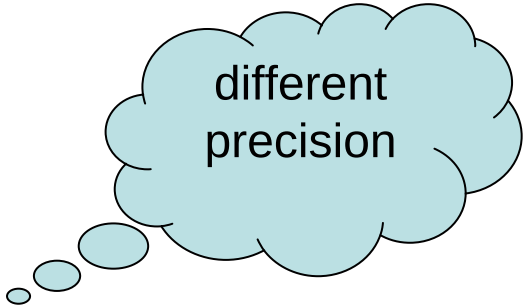
infinite domain



- unrealistic!

Another Approach

- collapse context environments
- collapse call caches
- abstract semantics



- different abstraction, different analysis

0-CFA

- no context environment
- all bindings of a given variable are merged together
- calls with distinct environments from the same call are merged together

0-CFA

consider the CBV lambda-calculus with scalars

$$x \in VAR$$

$$l \in LAB$$

$$e \equiv x^l \mid (e_1 \quad e_2)^l \mid \left(\lambda^l x.e\right) \mid b^l$$

$$v \equiv \left(\lambda^l x.e, \rho\right) \mid b^l$$

$$\rho \equiv [\ ] \mid [x = v]\rho$$

0-CFA

$$(x^l, \rho) \Downarrow \rho(x)$$

$$((\lambda^l x.e), \rho) \Downarrow ((\lambda^l x.e), \rho)$$

$$(b^l, \rho) \Downarrow (b^l, \rho)$$

$$(e_1, \rho) \Downarrow ((\lambda^{l'} x.e), \rho')$$

$$\frac{(e_2, \rho) \Downarrow v \qquad (e, [x = v]\rho') \Downarrow w}{((e_1 \quad e_2)^l, \rho) \Downarrow w}$$

0-CFA

dropping the environments results in the creations of abstract values

$$\hat{v} \equiv \left(\lambda^{l} x.e\right) \mid b^{l}$$

0-CFA

$$\hat{v} \equiv (\lambda^{l} x.e) \mid b^{l}$$

dropping the environments results in the creations of abstract values

and since labels are unique

0-CFA

$$\hat{v} \equiv \ell$$

we can use labels as abstract values

Defining a correct analysis

- So an analysis can be defined as:

$$\Phi : (LAB + VAR) \rightarrow P(\hat{V})$$

- An analysis describes an environment

$$\Phi \quad \succ \quad \rho$$

$$\Phi \succ [\;]$$

$$\Phi \succ [x = b^l]\rho \qquad , iff (l \in \Phi(x)) \wedge (\Phi \succ \rho)$$

$$\Phi \succ [x = ((\lambda^l z.e), \rho')]\rho \qquad , iff (l \in \Phi(x)) \wedge (\Phi \succ \rho') \wedge (\Phi \succ \rho)$$

Defining a correct analysis

$$\Phi \succ e^l \text{ iff for all } \rho :$$

$$\text{if } \Phi \succ \rho \text{ and } \left(e^l, \rho\right) \Downarrow v^{l'}, \text{ then}$$

$$1. \ell' \in \Phi\left(\ell\right)$$

$$2. \text{if } \mathrm{v}^{l'} = \left(\left(\lambda^{l'} z.e\right), \rho'\right),$$

$$\text{then } \left(\Phi \succ \rho'\right)$$

Generating a correct analysis based on a set of constraints

$$\frac{\left(\lambda^{l} x.e\right) \in U}{\left(l \in \Phi(l)\right) \in C[U]}$$

Generating a correct analysis based on a set of constraints

$$\frac{\left(\lambda^{l} x.e\right) \in U}{\left(l \in \Phi(l)\right) \in C[U]}$$

$$\frac{b^{l} \in U}{\left(l \in \Phi(l)\right) \in C[U]}$$

Generating a correct analysis based on a set of constraints

$$\frac{(\lambda^l x.e) \in U}{(l \in \Phi(l)) \in C[U]}$$

$$\frac{b^l \in U}{(l \in \Phi(l)) \in C[U]}$$

$$\frac{x^l \in U}{(\Phi(x) \in \Phi(l)) \in C[U]}$$

Generating a correct analysis based on a set of constraints

$$\frac{\left(\lambda^{l} x.e\right) \in U}{\left(l \in \Phi(l)\right) \in C[U]}$$

$$\frac{b^{l} \in U}{\left(l \in \Phi(l)\right) \in C[U]}$$

$$\frac{x^{l} \in U}{\left(\Phi(x) \in \Phi(l)\right) \in C[U]}$$

$$\frac{\left(e_1^{l_1} \quad e_2^{l_2}\right)^{l} \in U \quad \left(\lambda^{l'} x.e\right) \in U}{\left(\left(l' \in \Phi(l_1)\right) \Rightarrow \Phi(l_2) \subseteq \Phi(x)\right) \in C[U]}$$

Generating a correct analysis based on a set of constraints

$$\frac{(\lambda^{l} x.e) \in U}{(l \in \Phi(l)) \in C[U]}$$

$$\frac{b^{l} \in U}{(l \in \Phi(l)) \in C[U]}$$

$$\frac{x^{l} \in U}{(\Phi(x) \in \Phi(l)) \in C[U]}$$

$$\frac{\left(e_1^{l_1} \quad e_2^{l_2}\right)^{l} \in U \quad (\lambda^{l'} x.e) \in U}{((l' \in \Phi(l_1)) \Rightarrow \Phi(l_2) \subseteq \Phi(x)) \in C[U]}$$

$$\frac{\left(e_1^{l_1} \quad e_2^{l_2}\right)^{l} \in U \quad (\lambda^{l'} x.e^{l''}) \in U}{((l' \in \Phi(l_1)) \Rightarrow \Phi(l'') \subseteq \Phi(l)) \in C[U]}$$

Generating a correct analysis based on a set of constraints

- $O(n^2)$ constraints
- Relaxation on the rules in $C[U]$
- We can find the smallest correct Φ in $O(n^3)$ time

A Non-higher order Language

- $e \equiv x^l \mid b^l \mid \varphi^l(e_1 \ldots . e_n) \mid if^{\,l}\ e_0 \text{ then } e_1 \text{ else } e_2$

  $\varphi \equiv op \mid New \mid Upd \mid \mathrm{Re}\,f \mid f_i$

only scalars
in the arrays

A Non-higher order Language

- $e \equiv x^l \mid b^l \mid \varphi^l(e_1 \ldots . e_n) \mid if^l\ e_0 \text{ then } e_1 \text{ else } e_2$

  $\varphi \equiv op \mid New \mid Upd \mid \mathrm{Re}\ f \mid f_i$

$$f_1(x_{11},\ldots,x_{1n}) = e_1^{\ell_1}$$

$$\vdots$$

$$f_n(x_{n1},\ldots,x_{nn}) = e_n^{\ell n}$$

$$in\ e_0^{\ell_0}$$

A Non-higher order Language

- $e \equiv x^l \mid b^l \mid \varphi^l(e_1 \ldots. e_n) \mid if^l\ e_0 \text{ then } e_1 \text{ else } e_2$

  $\varphi \equiv op \mid New \mid Upd \mid \mathrm{Re}\ f \mid f_i$

- $Conf \equiv \langle halted, v \rangle \mid \langle a, \rho, G, K, \Sigma \rangle$

A Non-higher order Language

- $e \equiv x^l \mid b^l \mid \varphi^l(e_1 \ldots . e_n) \mid if^l\ e_0 \text{ then } e_1 \text{ else } e_2$

  $\varphi \equiv op \mid New \mid Upd \mid \mathrm{Re}\ f \mid f_i$

- $Conf \equiv \langle halted, v \rangle \mid \langle a, \rho, G, K, \Sigma \rangle$

computation address

A Non-higher order Language

- $e \equiv x^l \mid b^l \mid \varphi^l(e_1 \ldots . e_n) \mid if^l\ e_0 \text{ then } e_1 \text{ else } e_2$

  $\varphi \equiv op \mid New \mid Upd \mid \mathrm{Re}\ f \mid f_i$

- $Conf \equiv \langle halted, v \rangle \mid \langle a, \rho, G, K, \Sigma \rangle$

contexts

A Non-higher order Language

- $e \equiv x^l \mid b^l \mid \varphi^l(e_1 \ldots . e_n) \mid if^l\ e_0 \text{ then } e_1 \text{ else } e_2$
  $\varphi \equiv op \mid New \mid Upd \mid \mathrm{Re}\ f \mid f_i$

- $Conf \equiv \langle halted, v \rangle \mid \langle a, \rho, G, K, \Sigma \rangle$

partially reduced expressions

A Non-higher order Language

- $e \equiv x^l \mid b^l \mid \varphi^l(e_1 \ldots e_n) \mid if^l\ e_0 \text{ then } e_1 \text{ else } e_2$

  $\varphi \equiv op \mid New \mid Upd \mid \mathrm{Re}\,f \mid f_i$

- $Conf \equiv \langle halted, v \rangle \mid \langle a, \rho, G, K, \Sigma \rangle$

continuation

A Non-higher order Language

- $e \equiv x^l \mid b^l \mid \varphi^l(e_1 \ldots . e_n) \mid if^{\,l}\ e_0 \text{ then } e_1 \text{ else } e_2$

  $\varphi \equiv op \mid New \mid Upd \mid \text{Re}\ f \mid f_i$

- $Conf \equiv \langle halted, v \rangle \mid \langle a, \rho, G, K, \Sigma \rangle$

store

A Non-higher order Language

- $e \equiv x^l \mid b^l \mid \varphi^l(e_1....e_n) \mid if^{\,l}\ e_0 \text{ then } e_1 \text{ else } e_2$

  $\varphi \equiv op \mid New \mid Upd \mid \text{Re}\ f \mid f_i$

- $Conf \equiv \langle halted, v \rangle \mid \langle a, \rho, G, K, \Sigma \rangle$

- the initial configuration:

$$\langle a_0, \rho_0, e_0, halt, \varnothing \rangle$$

Generating a correct analysis based on a set of constraints

$$\frac{b^{l} \in U}{\left(l \in \Phi\left(l\right)\right) \in C\left[U\right]}$$

Generating a correct analysis based on a set of constraints

$$\frac{b^l \in U}{\left(l \in \Phi\left(l\right)\right) \in C\left[U\right]}$$

$$\boxed{\frac{x^l \in U}{\left(\Phi\left(x\right) \in \Phi\left(l\right)\right) \in C\left[U\right]}}$$

Generating a correct analysis based on a set of constraints

$$\frac{b^{l} \in U}{\left(l \in \Phi\left(l\right)\right) \in C\left[U\right]}$$

$$\frac{x^{l} \in U}{\left(\Phi\left(x\right) \in \Phi\left(l\right)\right) \in C\left[U\right]}$$

$$\frac{f_i^{\,l}\left(e_1^{\,l_1},\ldots,e_n^{\,l_n}\right) \in U \quad f_i\left(x_{i1},\ldots,x_{in}\right) \equiv e^{l'}}{\left(\Phi\left(l_j\right) \subseteq \Phi\left(x_{ij}\right)\right) \in C\left[U\right]}$$

Generating a correct analysis based on a set of constraints

$$\frac{b^{l} \in U}{\left(l \in \Phi\left(l\right)\right) \in C\left[U\right]}$$

$$\frac{x^{l} \in U}{\left(\Phi\left(x\right) \in \Phi\left(l\right)\right) \in C\left[U\right]}$$

$$\frac{f_i^{\,l}\left(e_1^{\,l_1},\ldots,e_n^{\,l_n}\right) \in U \quad f_i\left(x_{i1},\ldots,x_{in}\right) \equiv e^{l'}}{\left(\Phi\left(l_j\right) \subseteq \Phi\left(x_{ij}\right)\right) \in C\left[U\right]}$$

$$\frac{f_i^{\,l}\left(e_1^{\,l_1},\ldots,e_n^{\,l_n}\right) \in U \quad f_i\left(x_{i1},\ldots,x_{in}\right) \equiv e^{l'}}{\left(\Phi\left(l'\right) \subseteq \Phi\left(l\right)\right) \in C\left[U\right]}$$

Generating a correct analysis based on a set of constraints

$$\frac{b^{l} \in U}{\left(l \in \Phi(l)\right) \in C[U]}$$

$$\frac{x^{l} \in U}{\left(\Phi(x) \in \Phi(l)\right) \in C[U]}$$

$$\frac{f_i^{l}\left(e_1^{l_1},\ldots,e_n^{l_n}\right) \in U \quad f_i\left(x_{i1},\ldots,x_{in}\right) \equiv e^{l'}}{\left(\Phi\left(l_j\right) \subseteq \Phi\left(x_{ij}\right)\right) \in C[U]}$$

$$\frac{f_i^{l}\left(e_1^{l_1},\ldots,e_n^{l_n}\right) \in U \quad f_i\left(x_{i1},\ldots,x_{in}\right) \equiv e^{l'}}{\left(\Phi\left(l'\right) \subseteq \Phi(l)\right) \in C[U]}$$

$$\frac{if^{\,l}\ e_0^{\,l_0}\ \text{then}\ e_1^{\,l_1}\ \text{else}\ e_2^{\,l_2} \in U}{\left(\Phi\left(l_1\right) \cup \Phi\left(l_2\right) \subseteq \Phi(l)\right) \in C[U]}$$

Defining a Live Location

1. No location is live in *halt*
2. ℓ is live in $\langle a, \rho, R, K \rangle$ iff either:

   a. ℓ occurs in R, or

   b. there exists $x \in fv(R)$ such that $\rho(x) = \ell$, or

   c. ℓ is live in K

Defining a Sound Live Variable Analysis

A live variable analysis $\mathrm{L}[-]$ is a map from expression labels ℓ to sets of variables. $\mathrm{L}[-]$ is sound iff for each label ℓ, $\mathrm{L}[\ell]$ is a set of variables such that for all reachable store configurations of the form $\langle \mathrm{a}, \rho, \mathrm{e}^{\ell}, K, \Sigma \rangle$, $\rho(\mathrm{x})$ live in K implies $\mathrm{x} \in \mathrm{L}[\ell]$.

Set Constraints for a Sound Variable Analysis

1. if ℓ_i occurs in the context

$$\varphi^{\ell}\left(e_1^{l_1},\ldots,\ e_{i-1}^{l_{i-1}},e_i^{l_i},e_{i+1}^{l_{i+1}},\ldots,\ e_n^{l_n}\right)$$

then for every x $\in fv(e_i)$,

$$x \in L[\ell_i]$$

iff

$$\Phi(x)\cap\left(\bigcup_{j<i}\Phi(\ell_j)\right)\neq\emptyset\text{, or}$$

$$\Phi(x)\cap\left(\bigcup_{j>i}\ \bigcup_{y\in fv(e_j)}\Phi(y)\right)\neq\emptyset\text{, or}$$

$$x \in L[\ell]$$

Set Constraints for a Sound Variable Analysis

2. if ℓ_i occurs in the context

$$\text{if}^{\ell}\ \text{e}_0^{\ell_0}\ \text{then}\ \text{e}_1^{\ell_1}\ \text{else}\ \text{e}_2^{\ell_2}$$

then for every $\text{x} \in fv(e_0)$,

$$\text{x} \in \text{L}[\ell_0]$$

iff

$$\Phi(x) \cap \left(\bigcup_{j=1,2} \bigcup_{y \in fv(e_j)} \Phi(y) \right) \neq \varnothing, \text{ or}$$

$$\text{x} \in L[\ell]$$

Set Constraints for a Sound Variable Analysis

3. if ℓ_i occurs in the context

$$\text{if}^{\ell}\ e_0^{\ell_0}\ \text{then}\ e_1^{\ell_1}\ \text{else}\ e_2^{\ell_2}$$

then for every $x \in fv(e_i)$, $(i = 1,2)$

$$x \in L[\ell_i]$$

iff

$$x \in L[\ell]$$

Set Constraints for a Sound Variable Analysis

4. if $f_k(\mathrm{x}_{\mathrm{k1}},\ldots,x_{kn})=\mathrm{e}^{l'}$ then for each call

$$f_k{}^{l}\left(e_1{}^{l_1},\ldots,e_{i-1}{}^{l_{i-1}},e_i{}^{l_i},e_{i+1}{}^{l_{i+1}},\ldots,e_n{}^{l_n}\right)$$

,

$$\mathrm{x}_{\mathrm{ki}} \in \mathrm{L}[\ell']$$

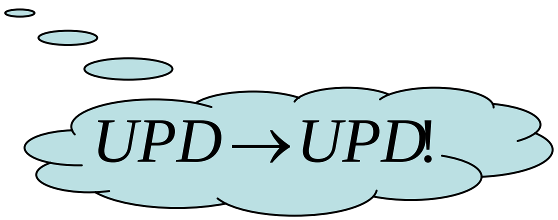
iff

$$\mathrm{L}[\ell]\cap\left\{y\in fv(e_j)\,|\,\left(\Phi\left(\ell_j\right)\cap\Phi(y)\neq\varnothing\right)\right\}\neq\varnothing$$

Work in progress

- “Set Constraints for Destructive Array Optimization”

  Mitch Wand and Will Clinger



Work in progress

- “Set Constraints for Destructive Array Optimization”

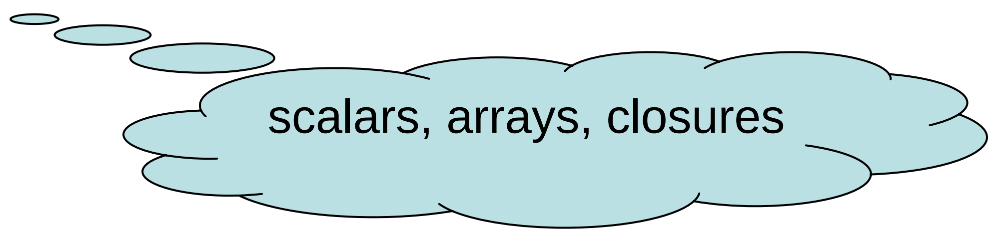
  Mitch Wand and Will Clinger

- Higher-order Languages

Work in progress

- "Set Constraints for Destructive Array Optimization"

  Mitch Wand and Will Clinger
- Higher-order Languages
- Arrays which can store any kind of value



Thank you