CS738: Advanced Compiler Optimizations

Points-to Analysis using Types

Amey Karkare

karkare@cse.iitk.ac.in

http://www.cse.iitk.ac.in/~karkare/cs738

Department of CSE, IIT Kanpur

Reference Papers

- Bjarne Steensgaard: Points-to Analysis in Almost Linear Time. POPL 1996
- Manuvir Das: Unification-based pointer analysis with directional assignments. PLDI 2000

Language

$$S \quad ::= \quad x = y$$

Language

$$\begin{array}{lcl} S & ::= & x = y \\ & | & x = \&y \end{array}$$

Language

$$
\begin{array}{rcl}
S & ::= & x = y \\
 & | & x = \&y \\
 & | & x = *y
\end{array}
$$

Language

$$
\begin{array}{rcl}
S & ::= & x = y \\
& | & x = \&y \\
& | & x = *y \\
& | & x = \text{allocate}(y)
\end{array}
$$

Language

$$
\begin{array}{lcl}
S & ::= & x = y \\
& | & x = \&y \\
& | & x = *y \\
& | & x = \text{allocate}(y) \\
& | & *x = y
\end{array}
$$

Language

$$
\begin{array}{rcl}
S & ::= & x = y \\
& \mid & x = \&y \\
& \mid & x = *y \\
& \mid & x = \text{allocate}(y) \\
& \mid & *x = y \\
& \mid & x = \text{fun}(f_1, \ldots, f_n) \text{ returns } r \text{ in } S^*
\end{array}
$$

Language

$$
\begin{array}{rcl}
S & ::= & x = y \\
 & | & x = \&y \\
 & | & x = *y \\
 & | & x = \text{allocate}(y) \\
 & | & *x = y \\
 & | & x = \text{fun}(f_1, \ldots, f_n) \text{ returns } r \text{ in } S^* \\
 & | & x = \text{p}(y_1, \ldots, y_n)
\end{array}
$$

Language

$$
\begin{array}{rcl}
S & ::= & x = y \\
 & \mid & x = \&y \\
 & \mid & x = *y \\
 & \mid & x = \text{allocate}(y) \\
 & \mid & *x = y \\
 & \mid & x = \text{fun}(f_1, \ldots, f_n) \text{ returns } r \text{ in } S^* \\
 & \mid & x = \text{p}(y_1, \ldots, y_n)
\end{array}
$$

Steensgaard's Analysis

- Non standard Types

$$s \in \text{Symbols}$$

Steensgaard's Analysis

- Non standard Types

$$s \in \text{Symbols}$$

Steensgaard's Analysis

- Non standard Types

$$\begin{array}{rcl} s & \in & \text{Symbols} \\ \tau & \in & \text{Locations} ::= (\varphi, \alpha) \end{array}$$

Steensgaard's Analysis

- Non standard Types

$$
\begin{array}{lcl}
s & \in & \text{Symbols} \\
\tau & \in & \text{Locations} ::= (\varphi, \alpha) \\
\varphi & \in & \text{Ids} ::= \{s_1, \ldots, s_n\}
\end{array}
$$

Steensgaard's Analysis

- Non standard Types

$$
\begin{array}{lcl}
s & \in & \text{Symbols} \\
\tau & \in & \text{Locations} ::= (\varphi, \alpha) \\
\varphi & \in & \text{Ids} ::= \{s_1, \ldots, s_n\} \\
\alpha & \in & \text{Values} ::= \bot \mid \text{ptr}(\tau)
\end{array}
$$

Steensgaard's Analysis

- Non standard Types

$$\begin{array}{rcl} s & \in & \text{Symbols} \\ \tau & \in & \text{Locations} ::= (\varphi, \alpha) \\ \varphi & \in & \text{Ids} ::= \{s_1, \ldots, s_n\} \\ \alpha & \in & \text{Values} ::= \perp \mid \text{ptr}(\tau) \end{array}$$

A denotes type environment.

Steensgaard's Analysis

- Partial Order

$$\alpha_1 \trianglelefteq \alpha_2 \quad \Leftrightarrow \quad (\alpha_1 = \bot) \vee (\alpha_1 = \alpha_2)$$

Steensgaard's Analysis: Typing Rules

$$\frac{A \vdash x : (\varphi, \alpha) \qquad A \vdash y : (\varphi', \alpha') \qquad \alpha' \trianglelefteq \alpha}{A \vdash \mathsf{welltyped}(x = y)}$$

Steensgaard's Analysis: Typing Rules

$$\frac{A \vdash x : (\varphi, \alpha) \qquad A \vdash y : (\varphi', \alpha') \qquad \alpha' \trianglelefteq \alpha}{A \vdash \mathrm{welltyped}(x = y)}$$

$$\frac{A \vdash x : (\varphi, \alpha) \qquad A \vdash y : \tau \qquad \mathrm{ptr}(\tau) \trianglelefteq \alpha}{A \vdash \mathrm{welltyped}(x = \&y)}$$

Steensgaard's Analysis: Typing Rules

$$\frac{A \vdash x : (\varphi, \alpha) \qquad A \vdash y : (\varphi', \alpha') \qquad \alpha' \trianglelefteq \alpha}{A \vdash \mathrm{welltyped}(x = y)}$$

$$\frac{A \vdash x : (\varphi, \alpha) \qquad A \vdash y : \tau \qquad \mathrm{ptr}(\tau) \trianglelefteq \alpha}{A \vdash \mathrm{welltyped}(x = \&y)}$$

$$\frac{A \vdash x : (\varphi, \alpha) \qquad A \vdash y : (\varphi', \mathrm{ptr}(\varphi'', \alpha'')) \qquad \alpha'' \trianglelefteq \alpha}{A \vdash \mathrm{welltyped}(x = *y)}$$

Steensgaard's Analysis: Typing Rules

$$\frac{A \vdash x : (\varphi, \alpha) \qquad A \vdash y : (\varphi', \alpha') \qquad \alpha' \trianglelefteq \alpha}{A \vdash \mathrm{welltyped}(x = y)}$$

$$\frac{A \vdash x : (\varphi, \alpha) \qquad A \vdash y : \tau \qquad \mathrm{ptr}(\tau) \trianglelefteq \alpha}{A \vdash \mathrm{welltyped}(x = \&y)}$$

$$\frac{A \vdash x : (\varphi, \alpha) \qquad A \vdash y : (\varphi', \mathrm{ptr}(\varphi'', \alpha'')) \qquad \alpha'' \trianglelefteq \alpha}{A \vdash \mathrm{welltyped}(x = *y)}$$

$$\frac{A \vdash x : (\varphi, \mathrm{ptr}(\varphi', \alpha')) \qquad A \vdash y : (\varphi'', \alpha'') \qquad \alpha'' \trianglelefteq \alpha'}{A \vdash \mathrm{welltyped}(*x = y)}$$

Steensgaard's Analysis: Typing Rules

$$\frac{A \vdash x : (\varphi, \alpha) \qquad A \vdash y : (\varphi', \alpha') \qquad \alpha' \trianglelefteq \alpha}{A \vdash \mathrm{welltyped}(x = y)}$$

$$\frac{A \vdash x : (\varphi, \alpha) \qquad A \vdash y : \tau \qquad \mathrm{ptr}(\tau) \trianglelefteq \alpha}{A \vdash \mathrm{welltyped}(x = \&y)}$$

$$\frac{A \vdash x : (\varphi, \alpha) \qquad A \vdash y : (\varphi', \mathrm{ptr}(\varphi'', \alpha'')) \qquad \alpha'' \trianglelefteq \alpha}{A \vdash \mathrm{welltyped}(x = *y)}$$

$$\frac{A \vdash x : (\varphi, \mathrm{ptr}(\varphi', \alpha')) \qquad A \vdash y : (\varphi'', \alpha'') \qquad \alpha'' \trianglelefteq \alpha'}{A \vdash \mathrm{welltyped}(*x = y)}$$

$$\frac{A \vdash x : \tau}{A \vdash \mathrm{welltyped}(x = \mathrm{allocate}(y))}$$

Steensgaard's Analysis

- Function Definitions

Steensgaard's Analysis

- Function Definitions
- Need a new type value: $(\tau_1 \ldots \tau_n) \rightarrow \tau$

Steensgaard's Analysis

- Function Definitions
- Need a new type value: $(\tau_1 \ldots \tau_n) \rightarrow \tau$

Steensgaard's Analysis

- Function Definitions
- Need a new type value: $(\tau_1 \ldots \tau_n) \to \tau$

$$A \vdash x : (\tau_1 \ldots \tau_n) \to \tau$$

Steensgaard's Analysis

- Function Definitions
- Need a new type value: $(\tau_1 \ldots \tau_n) \rightarrow \tau$

$$A \vdash x : (\tau_1 \ldots \tau_n) \rightarrow \tau$$
$$\forall i \in \{1 \ldots n\}. A \vdash f_i : \tau_i$$

Steensgaard's Analysis

- Function Definitions
- Need a new type value: $(\tau_1 \ldots \tau_n) \to \tau$

$$\begin{gathered} A \vdash x : (\tau_1 \ldots \tau_n) \to \tau \\ \forall i \in \{1 \ldots n\}. A \vdash f_i : \tau_i \\ A \vdash r : \tau \end{gathered}$$

Steensgaard's Analysis

- Function Definitions
- Need a new type value: $(\tau_1 \ldots \tau_n) \rightarrow \tau$

$$\begin{gathered} A \vdash x : (\tau_1 \ldots \tau_n) \rightarrow \tau \\ \forall i \in \{1 \ldots n\}. A \vdash f_i : \tau_i \\ A \vdash r : \tau \\ \forall s \in S^*. A \vdash \text{welltyped}(s) \end{gathered}$$

Steensgaard's Analysis

- Function Definitions
- Need a new type value: $(\tau_1 \ldots \tau_n) \rightarrow \tau$

$$\frac{\begin{array}{c} A \vdash x : (\tau_1 \ldots \tau_n) \rightarrow \tau \\ \forall i \in \{1 \ldots n\}. A \vdash f_i : \tau_i \\ A \vdash r : \tau \\ \forall s \in S^*. A \vdash \text{welltyped}(s) \end{array}}{A \vdash \text{welltyped}(x = \text{fun}(f_1, \ldots, f_n) \text{ returns } r \text{ in } S^*)}$$

Steensgaard's Analysis

- Function Calls

$$A \vdash x : \tau \qquad\qquad \tau = (\varphi, \alpha)$$

Steensgaard's Analysis

- Function Calls

$$A \vdash x : \tau \qquad\qquad \tau = (\varphi, \alpha)$$

Steensgaard's Analysis

- Function Calls

$$A \vdash x : \tau \qquad\qquad \tau = (\varphi, \alpha)$$
$$A \vdash \mathsf{p} : (\tau_1 \ldots \tau_n) \to \tau' \qquad\qquad \tau_i = (\varphi_i, \alpha_i)$$

Steensgaard's Analysis

- Function Calls

$$\begin{array}{ll} A \vdash x : \tau & \tau = (\varphi, \alpha) \\ A \vdash \mathrm{p} : (\tau_1 \ldots \tau_n) \to \tau' & \tau_i = (\varphi_i, \alpha_i) \\ \forall i \in \{1 \ldots n\}. A \vdash y_i : \tau_i' & \tau_i' = (\varphi_i', \alpha_i') \end{array}$$

Steensgaard's Analysis

- Function Calls

$$\begin{array}{ll} A \vdash x : \tau & \tau = (\varphi, \alpha) \\ A \vdash \mathrm{p} : (\tau_1 \ldots \tau_n) \rightarrow \tau' & \tau_i = (\varphi_i, \alpha_i) \\ \forall i \in \{1 \ldots n\}. A \vdash y_i : \tau_i' & \tau_i' = (\varphi_i', \alpha_i') \\ \alpha_i' \trianglelefteq \alpha_i & \alpha' \trianglelefteq \alpha \end{array}$$

Steensgaard's Analysis

- Function Calls

$$\frac{\begin{array}{ll} A \vdash x : \tau & \tau = (\varphi, \alpha) \\ A \vdash \mathsf{p} : (\tau_1 \ldots \tau_n) \to \tau' & \tau_i = (\varphi_i, \alpha_i) \\ \forall i \in \{1 \ldots n\}. A \vdash y_i : \tau_i' & \tau_i' = (\varphi_i', \alpha_i') \\ \alpha_i' \trianglelefteq \alpha_i & \alpha' \trianglelefteq \alpha \end{array}}{A \vdash \mathsf{welltyped}(x = \mathsf{p}(y_1, \ldots, y_n))}$$

Manuvir Das's *One-level Flow-based* Analysis

$$\alpha_1 \leq \alpha_2 \quad \Leftrightarrow \quad \mathsf{ptr}(\tau_1) \leq \mathsf{ptr}(\tau_2)$$

Manuvir Das's *One-level Flow-based* Analysis

$$
\begin{aligned}
\alpha_1 \leq \alpha_2 \quad & \Leftrightarrow \quad \mathsf{ptr}(\tau_1) \leq \mathsf{ptr}(\tau_2) \\
& \Leftrightarrow \quad \mathsf{ptr}((\varphi', \alpha')) \leq \mathsf{ptr}((\varphi, \alpha))
\end{aligned}
$$

Manuvir Das's *One-level Flow-based* Analysis

$$
\begin{aligned}
\alpha_1 \le \alpha_2 \quad &\Leftrightarrow \quad \mathsf{ptr}(\tau_1) \le \mathsf{ptr}(\tau_2) \\
&\Leftrightarrow \quad \mathsf{ptr}((\varphi', \alpha')) \le \mathsf{ptr}((\varphi, \alpha)) \\
&\Leftrightarrow \quad (\varphi' \subseteq \varphi) \wedge (\alpha' = \alpha)
\end{aligned}
$$

One-level Flow-based Analysis

- Replace $\trianglelefteq$ by $\leq$ in Steensgaard's analysis

One-level Flow-based Analysis

- Replace $\trianglelefteq$ by $\leq$ in Steensgaard's analysis
- Keeps "top" level pointees separate!