Compiler Construction

Lecture 25: Code Optimization II (The Dataflow Analysis Framework)

Thomas Noll

Lehrstuhl für Informatik 2
(Software Modeling and Verification)

RWTH Aachen University

noll@cs.rwth-aachen.de

http://www-i2.informatik.rwth-aachen.de/i2/cc09/

Winter semester 2009/10

Outline

Formalizing Available Expressions Analysis

- Given $C \in Cmd$, $Lab_C/Blk_C/AExp_C$ denote the sets of all labels/blocks/complex arithmetic expressions occurring in C, respectively
- An expression A is killed in a block β if any of the variables in A is modified in β
- Formally: $\text{kill}_{AE} : Blk_C \to 2^{AExp_C}$ is defined by
$$\begin{aligned}\text{kill}_{AE}([I := A]^l) &:= \{A' \in AExp_C \mid I \in \text{FV}(A')\}\\ \text{kill}_{AE}([B]^l) &:= \emptyset\end{aligned}$$
- An expression A is generated in a block β if it is evaluated in and none of its variables are modified by β
- Formally: $\text{gen}_{AE} : Blk_C \to 2^{AExp_C}$ is defined by
$$\begin{aligned}\text{gen}_{AE}([I := A]^l) &:= \{A \mid I \notin \text{FV}(A)\}\\ \text{gen}_{AE}([B]^l) &:= AExp_B\end{aligned}$$

The Available Expressions Equation System

- Analysis itself defined by setting up an equation system
- For each $l \in Lab_C$, $AE_l \subseteq AExp_C$ represents the set of available expressions at the entry of block β^l
- Formally, for $C \in Cmd$ with isolated entry:

$$AE_l = \begin{cases} \emptyset & \text{if } l = \text{init}(C) \\ \bigcap\{\varphi_{l'}(AE_{l'}) \mid (l', l) \in \text{flow}(C)\} & \text{otherwise} \end{cases}$$

  where $\varphi_{l'} : 2^{AExp_C} \to 2^{AExp_C}$ denotes the transfer function of block $\beta^{l'}$, given by

$$\varphi_{l'}(A) := (A \setminus \text{kill}_{AE}(\beta^{l'})) \cup \text{gen}_{AE}(\beta^{l'})$$

- Characterization of analysis:
  - forward: starts in init(C) and proceeds downwards
  - must: $\bigcap$ in equation for AE_l
  - flow–sensitive: results depending on order of assignments
- Later: solution not necessarily unique
  $\implies$ choose greatest one

Formalizing Live Variables Analysis

- A variable on the left-hand side of an assignment is killed by the assignment; tests do not kill
- Formally: $\text{kill}_{LV} : Blk_C \to 2^{Var_C}$ is defined by
  $$\text{kill}_{LV}([I := A]^l) := \{I\}$$
  $$\text{kill}_{LV}([B]^l) := \emptyset$$
- Every reading access generates a live variable
- Formally: $\text{gen}_{LV} : Blk_C \to 2^{Var_C}$ is defined by
  $$\text{gen}_{LV}([I := A]^l) := \text{FV}(A)$$
  $$\text{gen}_{LV}([B]^l) := \text{FV}(B)$$

The Live Variables Equation System

- For each $l \in Lab_C$, $LV_l \subseteq Var_c$ represents the set of live variables at the exit of block β^l
- Formally, for a program $C \in Cmd$ with isolated exits:
$$LV_l = \begin{cases} Var_C & \text{if } l \in \text{final}(C) \\ \bigcup\{\varphi_{l'}(LV_{l'}) \mid (l, l') \in \text{flow}(C)\} & \text{otherwise} \end{cases}$$
where $\varphi_{l'} : 2^{Var_C} \to 2^{Var_C}$ denotes the transfer function of block $\beta^{l'}$, given by
$$\varphi_{l'}(V) := (V \setminus \text{kill}_{LV}(\beta^{l'})) \cup \text{gen}_{LV}(\beta^{l'})$$
- Characterization of analysis:

  backward: starts in final(C) and proceeds upwards

  may: $\bigcup$ in equation for LV_l

  flow-sensitive: results depending on order of assignments

- Later: solution not necessarily unique

  $\Longrightarrow$ choose least one

Outline

Similarities between Analysis Problems

- **Observation:** the analyses presented so far have some similarities

$\Longrightarrow$ Look for underlying framework

- **Advantage:** possibility for designing (efficient) generic algorithms for solving dataflow equations
- **Overall pattern:** for $C \in Cmd$ and $l \in Lab_C$, the analysis information (AI) is described by equations of the form

$$AI_l = \begin{cases} \iota & \text{if } l \in E \\ \bigoplus\{\varphi_{l'}(AI_{l'}) \mid (l', l) \in F\} & \text{otherwise} \end{cases}$$

where

- ι specifies the initial analysis information
- E is $\{\text{init}(C)\}$ or $\text{final}(C)$
- $\bigoplus$ is $\bigcap$ or $\bigcup$
- $\varphi_{l'}$ denotes the transfer function of block $\beta^{l'}$
- F is $\text{flow}(C)$ or $\text{flow}^R(C)$ ($:= \{(l', l) \mid (l, l') \in \text{flow}(C)\}$)

Characterization of Analyses

- **Direction of information flow:**
  - forward:
    - $E = \{\text{init}(C)\}$
    - c has isolated entry
    - $F = \text{flow}(C)$
    - AI_l concerns entry of β^l
  - backward:
    - $E = \text{final}(C)$
    - c has isolated exits
    - $F = \text{flow}^R(C)$
    - AI_l concerns exit of β^l
- **Quantification over paths:**
  - may:
    - $\bigoplus = \bigcup$
    - property satisfied by some path
    - interested in least solution (later)
  - must:
    - $\bigoplus = \bigcap$
    - property satisfied by all paths
    - interested in greatest solution (later)

Outline

Fixpoint Iteration I

Idea: use fixpoint iteration to solve dataflow equation system

1. For $C \in Cmd$ and $l \in Lab_C$, start with "initial" information AI_l ($AE_l = AExp_C$, $LV_l = \emptyset$)
2. Iteratively evaluate dataflow equations until fixpoint reached

Theoretical foundations:

- Analysis information D forms complete lattice ($D_{AE} = 2^{AExp_C}$, $D_{LV} = 2^{Var_C}$)
  - every subset of D has a least upper/greatest lower bound $\implies$ well-definedness of $\bigoplus$
- ... that satisfies the ascending chain condition
  - $d_1 \overset{\supseteq}{\subseteq} d_2 \overset{\supseteq}{\subseteq} \ldots \implies \exists n : d_n = d_{n+1} = \ldots$
- Combination operator and all transfer functions monotonic
  - $d_1 \overset{\supseteq}{\subseteq} d_2 \implies \varphi(d_1) \overset{\supseteq}{\subseteq} \varphi(d_2)$

$\implies$ Fixpoint effectively computable by iteration

Fixpoint Iteration II

Example 25.1 (Available Expressions; cf. Example 24.9)

Program:

$C = [\texttt{x := a+b}]^1;$
$\quad [\texttt{y := a*b}]^2;$
$\quad \texttt{while } [\texttt{y > a+b}]^3 \texttt{ do}$
$\quad\quad [\texttt{a := a+1}]^4;$
$\quad\quad [\texttt{x := a+b}]^5$

Equation system:

$$
\begin{aligned}
AE_1 &= \emptyset \\
AE_2 &= AE_1 \cup \{\texttt{a+b}\} \\
AE_3 &= (AE_2 \cup \{\texttt{a*b}\}) \cap (AE_5 \cup \{\texttt{a+b}\}) \\
AE_4 &= AE_3 \cup \{\texttt{a+b}\} \\
AE_5 &= AE_4 \setminus \{\texttt{a+b}, \texttt{a*b}, \texttt{a+1}\}
\end{aligned}
$$

Fixpoint iteration:

i	1	2	3	4	5
0	$AExp_c$	$AExp_c$	$AExp_c$	$AExp_c$	$AExp_c$
1	$\emptyset$	$AExp_c$	$AExp_c$	$AExp_c$	$\emptyset$
2	$\emptyset$	$\{\texttt{a+b}\}$	$\{\texttt{a+b}\}$	$AExp_c$	$\emptyset$
3	$\emptyset$	$\{\texttt{a+b}\}$	$\{\texttt{a+b}\}$	$\{\texttt{a+b}\}$	$\emptyset$
4	$\emptyset$	$\{\texttt{a+b}\}$	$\{\texttt{a+b}\}$	$\{\texttt{a+b}\}$	$\emptyset$

Example 25.2 (Live Variables; cf. Example 24.12)

Program:

```
[x := 2]^1;
[y := 4]^2;
[x := 1]^3;
if [y > 0]^4 then
  [z := x]^5
else
  [z := y*y]^6;
[x := z]^7
```

Equation system:

$$
\begin{aligned}
LV_1 &= LV_2 \setminus \{y\} \\
LV_2 &= LV_3 \setminus \{x\} \\
LV_3 &= LV_4 \cup \{y\} \\
LV_4 &= ((LV_5 \setminus \{z\}) \cup \{x\}) \cup ((LV_6 \setminus \{z\}) \cup \{y\}) \\
LV_5 &= (LV_7 \setminus \{x\}) \cup \{z\} \\
LV_6 &= (LV_7 \setminus \{x\}) \cup \{z\} \\
LV_7 &= \{x, y, z\}
\end{aligned}
$$

Fixpoint iteration:

i	1	2	3	4	5	6	7
0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
1	$\emptyset$	$\emptyset$	$\{y\}$	$\{x, y\}$	$\{z\}$	$\{z\}$	$\{x, y, z\}$
2	$\emptyset$	$\{y\}$	$\{x, y\}$	$\{x, y\}$	$\{y, z\}$	$\{y, z\}$	$\{x, y, z\}$
3	$\emptyset$	$\{y\}$	$\{x, y\}$	$\{x, y\}$	$\{y, z\}$	$\{y, z\}$	$\{x, y, z\}$

Outline

Uniqueness of Solutions I

Example 25.3 (Available Expressions)

Consider

```
[z := x+y]^1;
while [true]^2 do
  [z := z]^3;
```

$$\Longrightarrow \begin{aligned} AE_1 &= \emptyset \\ AE_2 &= (AE_1 \cup \{\texttt{x+y}\}) \cap AE_3 \\ AE_3 &= AE_2 \end{aligned}$$

$$\Longrightarrow \begin{aligned} AE_1 &= \emptyset \\ AE_2 &= \{\texttt{x+y}\} \cap AE_3 \\ AE_3 &= AE_2 \end{aligned}$$

$\Longrightarrow$ Solutions: $AE_1 = AE_2 = AE_3 = \emptyset$ or
$AE_1 = \emptyset, AE_2 = AE_3 = \{\texttt{x+y}\}$

Here: greatest solution $\{\texttt{x+y}\}$ (maximal potential for optimization)
$\Longrightarrow$ start fixpoint iteration with greatest element $AExp_C$

Uniqueness of Solutions II

Example 25.4 (Live Variables)

Consider

```
while [x>1]^1 do
  [x := x]^2;
[x := x+1]^3;
[y := 0]^4
```

$$\Longrightarrow \begin{aligned} LV_1 &= (LV_2 \cup \{x\}) \cup (LV_3 \cup \{x\}) \\ LV_2 &= LV_1 \cup \{x\} \\ LV_3 &= LV_4 \setminus \{y\} \\ LV_4 &= \{x, y\} \end{aligned}$$

$$\Longrightarrow LV_3 = \{x\}$$

$$\Longrightarrow \begin{aligned} LV_1 &= LV_2 \cup \{x\} \\ &= LV_1 \cup \{x\} \end{aligned}$$

$\Longrightarrow$ Solutions: $LV_1 = LV_2 = \{x\}$ or $\{x, y\}$, $LV_3 = LV_4 = \emptyset$

Here: least solution $\{x\}$ (maximal potential for optimization)

$\Longrightarrow$ start fixpoint iteration with least element $\emptyset$