Static Program Analysis

Lecture 4: Dataflow Analysis III (The 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/spa11/

Summer Semester 2011

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 $l \in L_c$, the analysis information (AI) is described by equations of the form

$$\mathrm{AI}_l = \begin{cases} \iota & \text{if } l \in E \\ \bigsqcup \{\varphi_{l'}(\mathrm{AI}_{l'}) \mid (l', l) \in F\} & \text{otherwise} \end{cases}$$

where

- the set of extremal labels, E, is $\{\mathrm{init}(c)\}$ or $\mathrm{final}(c)$
- ι specifies the extremal analysis information
- the combination operator, $\bigsqcup$, is $\bigcap$ or $\bigcup$
- $\varphi_{l'}$ denotes the transfer function of block $B^{l'}$
- the flow relation F is $\mathrm{flow}(c)$ or $\mathrm{flow}^R(c)$ $(:= \{(l', l) \mid (l, l') \in \mathrm{flow}(c)\})$

Roadmap

Goal: solve dataflow equation system by fixpoint iteration

1. Characterize solution of equation system as fixpoint of a transformation
2. Introduce partial order for comparing analysis results
3. Establish least upper bound as combination operator
4. Ensure monotonicity of transfer functions
5. Guarantee termination of fixpoint iteration by ascending chain condition
6. Optimize fixpoint iteration by worklist algorithm

Motivation

- **Wanted:** solution of (dataflow) equation system
- **Problem:** recursive dependencies between dataflow variables
- **Idea:** characterize solution as fixpoint of transformation:

$$(\mathsf{AI}_l = \tau_l)_{l \in L_c} \iff \Phi((\mathsf{AI}_l)_{l \in L_c}) = (\mathsf{AI}_l)_{l \in L_c}$$

  where $\Phi\left((\mathsf{AI}_l)_{l \in L_c}\right) := (\tau_l)_{l \in L_c}$
- **Approach:** approximate fixpoint by iteration

Partial Orders

The domain of analysis information usually forms a partial order where the ordering relation compares the "precision" of information.

Definition (Partial order)

A partial order (PO) $(D, \sqsubseteq)$ consists of a set D, called domain, and of a relation $\sqsubseteq \subseteq D \times D$ such that, for every $d_1, d_2, d_3 \in D$,

reflexivity: $d_1 \sqsubseteq d_1$

transitivity: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_3 \implies d_1 \sqsubseteq d_3$

antisymmetry: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_1 \implies d_1 = d_2$

It is called total if, in addition, always $d_1 \sqsubseteq d_2$ or $d_2 \sqsubseteq d_1$.

Example

1. $(\mathbb{N}, \leq)$ is a total partial order
2. $(\mathbb{N}, <)$ is not a partial order (since not reflexive)
3. (Live Variables) $(2^{Var_c}, \subseteq)$ is a (non-total) partial order
4. (Available Expressions) $(2^{AExp_c}, \supseteq)$ is a (non-total) partial order

Upper Bounds

In the dataflow equation system, analysis information from several predecessors is combined by taking the least upper bound.

Definition ((Least) upper bound)

Let $(D, \sqsubseteq)$ be a partial order and $S \subseteq D$.

1. An element $d \in D$ is called an upper bound of S if $s \sqsubseteq d$ for every $s \in S$ (notation: $S \sqsubseteq d$).
2. An upper bound d of S is called least upper bound (LUB) or supremum of S if $d \sqsubseteq d'$ for every upper bound d' of S (notation: $d = \bigsqcup S$).

Example

1. $S \subseteq \mathbb{N}$ has a LUB in $(\mathbb{N}, \leq)$ iff it is finite
2. (Live Variables) $(D, \sqsubseteq) = (2^{Var_c}, \subseteq)$. Given $V_1, \ldots, V_n \subseteq Var_c$,
$$\bigsqcup\{V_1, \ldots, V_n\} = \bigcup\{V_1, \ldots, V_n\}$$
3. (Avail. Expr.) $(D, \sqsubseteq) = (2^{AExp_c}, \supseteq)$. Given $A_1, \ldots, A_n \subseteq AExp_c$,
$$\bigsqcup\{A_1, \ldots, A_n\} = \bigcap\{A_1, \ldots, A_n\}$$

Complete Lattices

Since $\{\varphi_{l'}(\mathsf{AI}_{l'}) \mid (l', l) \in F\}$ can contain arbitrary elements, the existence of least upper bounds must be ensured for arbitrary subsets.

Definition (Complete lattice)

A complete lattice is a partial order $(D, \sqsubseteq)$ such that all subsets of D have least upper bounds. In this case,

$$\bot := \bigsqcup \emptyset$$

denotes the least element of D.

Example

1. $(\mathbb{N}, \leq)$ is not a complete lattice as, e.g., $\mathbb{N}$ does not have a LUB
2. (Live Variables)
   $(D, \sqsubseteq) = (2^{Var_c}, \subseteq)$ is a complete lattice with $\bot = \emptyset$
3. (Available Expressions)
   $(D, \sqsubseteq) = (2^{AExp_c}, \supseteq)$ is a complete lattice with $\bot = AExp_c$

Chains

Chains are generated by the approximation of the analysis information in the fixpoint iteration.

Definition (Chain)

Let $(D, \sqsubseteq)$ be a partial order. A subset $S \subseteq D$ is called an (ascending) chain in D if, for every $s_1, s_2 \in S$,

$$s_1 \sqsubseteq s_2 \text{ or } s_2 \sqsubseteq s_1$$

(that is, S is a totally ordered subset of D).

Example

1. Every $S \subseteq \mathbb{N}$ is a chain in $(\mathbb{N}, \leq)$
2. $\{\emptyset, \{0\}, \{0,1\}, \{0,1,2\}, \ldots\}$ is a chain in $(2^{\mathbb{N}}, \subseteq)$
3. $\{\emptyset, \{0\}, \{1\}\}$ is not a chain in $(2^{\mathbb{N}}, \subseteq)$

The Ascending Chain Condition

Termination of fixpoint iteration is guaranteed by the Ascending Chain Condition.

Definition (Ascending Chain Condition)

A partial order $(D, \sqsubseteq)$ satisfies the Ascending Chain Condition (ACC) if each ascending chain $d_1 \sqsubseteq d_2 \sqsubseteq \ldots$ eventually stabilizes, i.e., there exists $n \in \mathbb{N}$ such that $d_n = d_{n+1} = \ldots$

Example

1. $(\mathbb{N}, \leq)$ does not satisfy ACC
2. (Live Variables) $(D, \sqsubseteq) = (2^{Var_c}, \subseteq)$ satisfies ACC since Var_c (unlike Var) is finite
3. (Available Expressions) $(D, \sqsubseteq) = (2^{AExp_c}, \supseteq)$ satisfies ACC since $AExp_c$ (unlike $AExp$) is finite

Outline

Monotonicity of Functions

The monotonicity of transfer functions excludes "oscillating behavior" in fixpoint iteration.

Definition 4.1 (Monotonicity)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be partial orders, and let $\Phi : D \to D'$. Φ is called monotonic (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every $d_1, d_2 \in D$,

$$d_1 \sqsubseteq d_2 \implies \Phi(d_1) \sqsubseteq' \Phi(d_2).$$

Monotonicity of Functions

The monotonicity of transfer functions excludes "oscillating behavior" in fixpoint iteration.

Definition 4.1 (Monotonicity)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be partial orders, and let $\Phi : D \to D'$. Φ is called monotonic (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every $d_1, d_2 \in D$,

$$d_1 \sqsubseteq d_2 \implies \Phi(d_1) \sqsubseteq' \Phi(d_2).$$

Example 4.2

1. Let $T := \{S \subseteq \mathbb{N} \mid S \text{ finite}\}$. Then $\Phi_1 : T \to \mathbb{N} : S \mapsto \sum_{n \in S} n$ is monotonic w.r.t. $(2^{\mathbb{N}}, \subseteq)$ and $(\mathbb{N}, \leq)$.

Monotonicity of Functions

The monotonicity of transfer functions excludes "oscillating behavior" in fixpoint iteration.

Definition 4.1 (Monotonicity)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be partial orders, and let $\Phi : D \to D'$. Φ is called monotonic (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every $d_1, d_2 \in D$,

$$d_1 \sqsubseteq d_2 \implies \Phi(d_1) \sqsubseteq' \Phi(d_2).$$

Example 4.2

1. Let $T := \{S \subseteq \mathbb{N} \mid S \text{ finite}\}$. Then $\Phi_1 : T \to \mathbb{N} : S \mapsto \sum_{n \in S} n$ is monotonic w.r.t. $(2^{\mathbb{N}}, \subseteq)$ and $(\mathbb{N}, \leq)$.
2. $\Phi_2 : 2^{\mathbb{N}} \to 2^{\mathbb{N}} : S \mapsto \mathbb{N} \setminus S$ is not monotonic w.r.t. $(2^{\mathbb{N}}, \subseteq)$ (since, e.g., $\emptyset \subseteq \mathbb{N}$ but $\Phi_2(\emptyset) = \mathbb{N} \not\subseteq \Phi_2(\mathbb{N}) = \emptyset$).

Monotonicity of Functions

The monotonicity of transfer functions excludes "oscillating behavior" in fixpoint iteration.

Definition 4.1 (Monotonicity)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be partial orders, and let $\Phi : D \to D'$. Φ is called monotonic (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every $d_1, d_2 \in D$,

$$d_1 \sqsubseteq d_2 \implies \Phi(d_1) \sqsubseteq' \Phi(d_2).$$

Example 4.2

1. Let $T := \{S \subseteq \mathbb{N} \mid S \text{ finite}\}$. Then $\Phi_1 : T \to \mathbb{N} : S \mapsto \sum_{n \in S} n$ is monotonic w.r.t. $(2^{\mathbb{N}}, \subseteq)$ and $(\mathbb{N}, \leq)$.
2. $\Phi_2 : 2^{\mathbb{N}} \to 2^{\mathbb{N}} : S \mapsto \mathbb{N} \setminus S$ is not monotonic w.r.t. $(2^{\mathbb{N}}, \subseteq)$ (since, e.g., $\emptyset \subseteq \mathbb{N}$ but $\Phi_2(\emptyset) = \mathbb{N} \not\subseteq \Phi_2(\mathbb{N}) = \emptyset$).
3. (Live Variables) $(D, \sqsubseteq) = (D', \sqsubseteq') = (2^{Var_c}, \subseteq)$
   Each transfer function $\varphi_{l'}(V) := (V \setminus \mathrm{kill}_{\mathrm{LV}}(B^{l'})) \cup \mathrm{gen}_{\mathrm{LV}}(B^{l'})$ is obviously monotonic

Monotonicity of Functions

The monotonicity of transfer functions excludes "oscillating behavior" in fixpoint iteration.

Definition 4.1 (Monotonicity)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be partial orders, and let $\Phi : D \to D'$. Φ is called monotonic (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every $d_1, d_2 \in D$,

$$d_1 \sqsubseteq d_2 \implies \Phi(d_1) \sqsubseteq' \Phi(d_2).$$

Example 4.2

1. Let $T := \{S \subseteq \mathbb{N} \mid S \text{ finite}\}$. Then $\Phi_1 : T \to \mathbb{N} : S \mapsto \sum_{n \in S} n$ is monotonic w.r.t. $(2^{\mathbb{N}}, \subseteq)$ and $(\mathbb{N}, \leq)$.
2. $\Phi_2 : 2^{\mathbb{N}} \to 2^{\mathbb{N}} : S \mapsto \mathbb{N} \setminus S$ is not monotonic w.r.t. $(2^{\mathbb{N}}, \subseteq)$ (since, e.g., $\emptyset \subseteq \mathbb{N}$ but $\Phi_2(\emptyset) = \mathbb{N} \not\subseteq \Phi_2(\mathbb{N}) = \emptyset$).
3. (Live Variables) $(D, \sqsubseteq) = (D', \sqsubseteq') = (2^{Var_c}, \subseteq)$
   Each transfer function $\varphi_{l'}(V) := (V \setminus \text{kill}_{\text{LV}}(B^{l'})) \cup \text{gen}_{\text{LV}}(B^{l'})$ is obviously monotonic
4. (Available Expressions) $(D, \sqsubseteq) = (D', \sqsubseteq') = (2^{AExp_c}, \supseteq)$ ditto

Fixpoints

Definition 4.3 (Fixpoint)

Let D be some domain, $d \in D$, and $\Phi : D \to D$. If

$$\Phi(d) = d$$

then d is called a fixpoint of Φ.

Fixpoints

Definition 4.3 (Fixpoint)

Let D be some domain, $d \in D$, and $\Phi : D \to D$. If

$$\Phi(d) = d$$

then d is called a fixpoint of Φ.

Example 4.4

The (only) fixpoints of $\Phi : \mathbb{N} \to \mathbb{N} : n \mapsto n^2$ are 0 and 1

The Fixpoint Theorem I

Alfred Tarski (1901–1983)

Bronislaw Knaster (1893–1990)

Theorem 4.5 (Fixpoint Theorem by Tarski and Knaster)

Let $(D, \sqsubseteq)$ be a complete lattice satisfying ACC and $\Phi : D \to D$ monotonic. Then

$$\text{fix}(\Phi) := \bigsqcup \{\Phi^k(\bot) \mid k \in \mathbb{N}\}$$

is the least fixpoint of Φ where

$$\Phi^0(d) := d \text{ and } \Phi^{k+1}(d) := \Phi(\Phi^k(d)).$$

The Fixpoint Theorem I

Alfred Tarski (1901–1983)

Bronislaw Knaster (1893–1990)

Theorem 4.5 (Fixpoint Theorem by Tarski and Knaster)

Let $(D, \sqsubseteq)$ be a complete lattice satisfying ACC and $\Phi : D \to D$ monotonic. Then

$$\text{fix}(\Phi) := \bigsqcup \{\Phi^k(\bot) \mid k \in \mathbb{N}\}$$

is the least fixpoint of Φ where

$$\Phi^0(d) := d \text{ and } \Phi^{k+1}(d) := \Phi(\Phi^k(d)).$$

Remark: ACC $\implies$ $(\Phi^k(\bot) \mid k \in \mathbb{N})$ stabilizes at some $k_0 \in \mathbb{N}$ with $\text{fix}(\Phi) = \Phi^{k_0}(\bot)$ (where k_0 bounded by maximal chain length in $(D, \sqsubseteq)$)

The proof of Theorem 4.5 requires the following lemma.

Lemma 4.6

Let $(D, \sqsubseteq)$ *be a complete lattice satisfying ACC,* $S \subseteq D$ *a chain, and* $\Phi : D \to D$ *monotonic. Then*

$$\Phi(\bigsqcup S) = \bigsqcup \Phi(S)$$

The Fixpoint Theorem II

The proof of Theorem 4.5 requires the following lemma.

Lemma 4.6

Let $(D, \sqsubseteq)$ *be a complete lattice satisfying ACC,* $S \subseteq D$ *a chain, and* $\Phi : D \to D$ *monotonic. Then*

$$\Phi(\bigsqcup S) = \bigsqcup \Phi(S)$$

Proof (Lemma 4.6).

on the board □

The Fixpoint Theorem II

The proof of Theorem 4.5 requires the following lemma.

Lemma 4.6

Let $(D, \sqsubseteq)$ be a complete lattice satisfying ACC, $S \subseteq D$ a chain, and $\Phi : D \to D$ monotonic. Then

$$\Phi(\bigsqcup S) = \bigsqcup \Phi(S)$$

Proof (Lemma 4.6).

on the board □

Proof (Theorem 4.5).

on the board □

Outline

Dataflow Systems I

Definition 4.7 (Dataflow system)

A dataflow system $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$ consists of

- a finite set of (program) labels L (here: L_c),
- a set of extremal labels $E \subseteq L$ (here: $\{\text{init}(c)\}$ or $\text{final}(c)$),
- a flow relation $F \subseteq L \times L$ (here: $\text{flow}(c)$ or $\text{flow}^R(c)$),
- a complete lattice $(D, \sqsubseteq)$ satisfying ACC (with LUB operator $\bigsqcup$ and least element $\bot$),
- an extremal value $\iota \in D$ (for the extremal labels), and
- a collection of monotonic transfer functions $\{\varphi_l \mid l \in L\}$ of type $\varphi_l : D \to D$.

Example 4.8

Problem	**Available Expressions**	**Live Variables**
E	$\{\mathrm{init}(c)\}$	$\mathrm{final}(c)$
F	$\mathrm{flow}(c)$	$\mathrm{flow}^R(c)$
D	2^{AExp_c}	2^{Var_c}
$\sqsubseteq$	$\supseteq$	$\subseteq$
$\bigsqcup$	$\bigcap$	$\bigcup$
$\bot$	$AExp_c$	$\emptyset$
ι	$\emptyset$	Var_c
φ_l	$\varphi_l(d) = (d \setminus \mathrm{kill}(B^l)) \cup \mathrm{gen}(B^l)$	

Definition 4.9 (Dataflow equation system)

Given: dataflow system $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$, $L = \{1, \ldots, n\}$ (w.l.o.g.)

- S determines the equation system (where $l \in L$)

$$\mathsf{AI}_l = \begin{cases} \iota & \text{if } l \in E \\ \bigsqcup\{\varphi_{l'}(\mathsf{AI}_{l'}) \mid (l', l) \in F\} & \text{otherwise} \end{cases}$$

Dataflow Systems and Fixpoints

Definition 4.9 (Dataflow equation system)

Given: dataflow system $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$, $L = \{1, \ldots, n\}$ (w.l.o.g.)

- S determines the equation system (where $l \in L$)

$$\mathrm{AI}_l = \begin{cases} \iota & \text{if } l \in E \\ \bigsqcup\{\varphi_{l'}(\mathrm{AI}_{l'}) \mid (l', l) \in F\} & \text{otherwise} \end{cases}$$

- $(d_1, \ldots, d_n) \in D^n$ is called a solution if

$$d_l = \begin{cases} \iota & \text{if } l \in E \\ \bigsqcup\{\varphi_{l'}(d_{l'}) \mid (l', l) \in F\} & \text{otherwise} \end{cases}$$

Dataflow Systems and Fixpoints

Definition 4.9 (Dataflow equation system)

Given: dataflow system $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$, $L = \{1, ..., n\}$ (w.l.o.g.)

- S determines the equation system (where $l \in L$)

$$\mathrm{AI}_l = \begin{cases} \iota & \text{if } l \in E \\ \bigsqcup\{\varphi_{l'}(\mathrm{AI}_{l'}) \mid (l', l) \in F\} & \text{otherwise} \end{cases}$$

- $(d_1, \ldots, d_n) \in D^n$ is called a solution if

$$d_l = \begin{cases} \iota & \text{if } l \in E \\ \bigsqcup\{\varphi_{l'}(d_{l'}) \mid (l', l) \in F\} & \text{otherwise} \end{cases}$$

- S determines the transformation

$$\Phi_S : D^n \to D^n : (d_1, \ldots, d_n) \mapsto (d'_1, \ldots, d'_n)$$

where

$$d'_l := \begin{cases} \iota & \text{if } l \in E \\ \bigsqcup\{\varphi_{l'}(d_{l'}) \mid (l', l) \in F\} & \text{otherwise} \end{cases}$$

Dataflow Systems and Fixpoints

Definition 4.9 (Dataflow equation system)

Given: dataflow system $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$, $L = \{1, ..., n\}$ (w.l.o.g.)

- S determines the equation system (where $l \in L$)

$$\mathrm{AI}_l = \begin{cases} \iota & \text{if } l \in E \\ \bigsqcup\{\varphi_{l'}(\mathrm{AI}_{l'}) \mid (l', l) \in F\} & \text{otherwise} \end{cases}$$

- $(d_1, \ldots, d_n) \in D^n$ is called a solution if

$$d_l = \begin{cases} \iota & \text{if } l \in E \\ \bigsqcup\{\varphi_{l'}(d_{l'}) \mid (l', l) \in F\} & \text{otherwise} \end{cases}$$

- S determines the transformation

$$\Phi_S : D^n \to D^n : (d_1, \ldots, d_n) \mapsto (d'_1, \ldots, d'_n)$$

where

$$d'_l := \begin{cases} \iota & \text{if } l \in E \\ \bigsqcup\{\varphi_{l'}(d_{l'}) \mid (l', l) \in F\} & \text{otherwise} \end{cases}$$

Corollary 4.10

$(d_1, \ldots, d_n) \in D^n$ solves the equation system iff it is a fixpoint of Φ_S

Solving Dataflow Problems by Fixpoint Iteration

Remarks:

- $(D, \sqsubseteq)$ being a complete lattice ensures that Φ_S is well defined

Remarks:

- $(D, \sqsubseteq)$ being a complete lattice ensures that Φ_S is well defined
- Since $(D, \sqsubseteq)$ is a complete lattice satisfying ACC, so is $(D^n, \sqsubseteq^n)$ (where $(d_1, \ldots, d_n) \sqsubseteq^n (d'_1, \ldots, d'_n)$ iff $d_i \sqsubseteq d'_i$ for every $1 \leq i \leq n$)

Remarks:

- $(D, \sqsubseteq)$ being a complete lattice ensures that Φ_S is well defined
- Since $(D, \sqsubseteq)$ is a complete lattice satisfying ACC, so is $(D^n, \sqsubseteq^n)$ (where $(d_1, \ldots, d_n) \sqsubseteq^n (d'_1, \ldots, d'_n)$ iff $d_i \sqsubseteq d'_i$ for every $1 \leq i \leq n$)
- Monotonicity of transfer functions φ_l in $(D, \sqsubseteq)$ implies monotonicity of Φ_S in $(D^n, \sqsubseteq^n)$ (since $\bigsqcup$ also monotonic)

Solving Dataflow Problems by Fixpoint Iteration

Remarks:

- $(D, \sqsubseteq)$ being a complete lattice ensures that Φ_S is well defined
- Since $(D, \sqsubseteq)$ is a complete lattice satisfying ACC, so is $(D^n, \sqsubseteq^n)$ (where $(d_1, \ldots, d_n) \sqsubseteq^n (d_1', \ldots, d_n')$ iff $d_i \sqsubseteq d_i'$ for every $1 \leq i \leq n$)
- Monotonicity of transfer functions φ_l in $(D, \sqsubseteq)$ implies monotonicity of Φ_S in $(D^n, \sqsubseteq^n)$ (since $\bigsqcup$ also monotonic)
- Thus the (least) fixpoint is effectively computable by iteration:
$$\text{fix}(\Phi_S) = \bigsqcup \{\Phi_S^k(\bot_{D^n}) \mid k \in \mathbb{N}\}$$
where $\bot_{D^n} = (\underbrace{\bot_D, \ldots, \bot_D}_{n \text{ times}})$

Solving Dataflow Problems by Fixpoint Iteration

Remarks:

- $(D, \sqsubseteq)$ being a complete lattice ensures that Φ_S is well defined
- Since $(D, \sqsubseteq)$ is a complete lattice satisfying ACC, so is $(D^n, \sqsubseteq^n)$ (where $(d_1, \ldots, d_n) \sqsubseteq^n (d'_1, \ldots, d'_n)$ iff $d_i \sqsubseteq d'_i$ for every $1 \leq i \leq n$)
- Monotonicity of transfer functions φ_l in $(D, \sqsubseteq)$ implies monotonicity of Φ_S in $(D^n, \sqsubseteq^n)$ (since $\bigsqcup$ also monotonic)
- Thus the (least) fixpoint is effectively computable by iteration:
$$\text{fix}(\Phi_S) = \bigsqcup\{\Phi_S^k(\bot_{D^n}) \mid k \in \mathbb{N}\}$$
where $\bot_{D^n} = (\underbrace{\bot_D, \ldots, \bot_D}_{n \text{ times}})$
- If maximal length of chains in $(D, \sqsubseteq)$ (= height of $(D, \sqsubseteq)$) is m
  $\Longrightarrow$ maximal length of chains in $(D^n, \sqsubseteq^n)$ is $m \cdot n$
  $\Longrightarrow$ fixpoint iteration requires at most $m \cdot n$ steps

Example: Available Expressions

Example 4.11 (Available Expressions; cf. Example 2.9)

Program:

$c = [$`x := a+b`$]^1$;
$[$`y := a*b`$]^2$;
`while` $[$`y > a+b`$]^3$ `do`
$[$`a := a+1`$]^4$;
$[$`x := a+b`$]^5$

Example: Available Expressions

Example 4.11 (Available Expressions; cf. Example 2.9)

Program:

$c = $
```
[x := a+b]^1;
[y := a*b]^2;
while [y > a+b]^3 do
  [a := a+1]^4;
  [x := a+b]^5
```

Equation system:

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

Example: Available Expressions

Example 4.11 (Available Expressions; cf. Example 2.9)

Program:

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

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	$AExp_c$	$AExp_c$	$AExp_c$	$AExp_c$	$AExp_c$

Example: Available Expressions

Example 4.11 (Available Expressions; cf. Example 2.9)

Program:

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

Equation system:

$$\begin{aligned}
\mathrm{AE}_1 &= \emptyset \\
\mathrm{AE}_2 &= \mathrm{AE}_1 \cup \{\texttt{a+b}\} \\
\mathrm{AE}_3 &= (\mathrm{AE}_2 \cup \{\texttt{a*b}\}) \cap (\mathrm{AE}_5 \cup \{\texttt{a+b}\}) \\
\mathrm{AE}_4 &= \mathrm{AE}_3 \cup \{\texttt{a+b}\} \\
\mathrm{AE}_5 &= \mathrm{AE}_4 \setminus \{\texttt{a+b}, \texttt{a*b}, \texttt{a+1}\}
\end{aligned}$$

Fixpoint iteration:

i	1	2	3	4	5
0	$AExp_c$	$AExp_c$	$AExp_c$	$AExp_c$	$AExp_c$
1	$\emptyset$	$AExp_c$	$AExp_c$	$AExp_c$	$\emptyset$

Example: Available Expressions

Example 4.11 (Available Expressions; cf. Example 2.9)

Program:

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

Equation system:

$$\begin{aligned}
\mathsf{AE}_1 &= \emptyset \\
\mathsf{AE}_2 &= \mathsf{AE}_1 \cup \{\texttt{a+b}\} \\
\mathsf{AE}_3 &= (\mathsf{AE}_2 \cup \{\texttt{a*b}\}) \cap (\mathsf{AE}_5 \cup \{\texttt{a+b}\}) \\
\mathsf{AE}_4 &= \mathsf{AE}_3 \cup \{\texttt{a+b}\} \\
\mathsf{AE}_5 &= \mathsf{AE}_4 \setminus \{\texttt{a+b}, \texttt{a*b}, \texttt{a+1}\}
\end{aligned}$$

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	$\emptyset$	$\{\texttt{a+b}\}$	$\{\texttt{a+b}\}$	$AExp_c$	$\emptyset$

Example: Available Expressions

Example 4.11 (Available Expressions; cf. Example 2.9)

Program:

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

Equation system:

$$
\begin{aligned}
\mathrm{AE}_1 &= \emptyset\\
\mathrm{AE}_2 &= \mathrm{AE}_1 \cup \{\texttt{a+b}\}\\
\mathrm{AE}_3 &= (\mathrm{AE}_2 \cup \{\texttt{a*b}\}) \cap (\mathrm{AE}_5 \cup \{\texttt{a+b}\})\\
\mathrm{AE}_4 &= \mathrm{AE}_3 \cup \{\texttt{a+b}\}\\
\mathrm{AE}_5 &= \mathrm{AE}_4 \setminus \{\texttt{a+b}, \texttt{a*b}, \texttt{a+1}\}
\end{aligned}
$$

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	$\emptyset$	$\{\texttt{a+b}\}$	$\{\texttt{a+b}\}$	$\{\texttt{a+b}\}$	$\emptyset$

Example: Available Expressions

Example 4.11 (Available Expressions; cf. Example 2.9)

Program:

```
c = [x := a+b]^1;
    [y := a*b]^2;
    while [y > a+b]^3 do
      [a := a+1]^4;
      [x := a+b]^5
```

Equation system:

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

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	$\emptyset$	$\{a+b\}$	$\{a+b\}$	$AExp_c$	$\emptyset$
3	$\emptyset$	$\{a+b\}$	$\{a+b\}$	$\{a+b\}$	$\emptyset$
4	$\emptyset$	$\{a+b\}$	$\{a+b\}$	$\{a+b\}$	$\emptyset$

Example: Live Variables

Example 4.12 (Live Variables; cf. Example 3.3)

Program:

$[\texttt{x := 2}]^1;[\texttt{y := 4}]^2;$
$[\texttt{x := 1}]^3;$
$\texttt{if } [\texttt{y > 0}]^4 \texttt{ then}$
$\quad [\texttt{z := x}]^5$
$\texttt{else}$
$\quad [\texttt{z := y*y}]^6;$
$[\texttt{x := z}]^7$

Example: Live Variables

Example 4.12 (Live Variables; cf. Example 3.3)

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:

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

Example: Live Variables

Example 4.12 (Live Variables; cf. Example 3.3)

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	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Example: Live Variables

Example 4.12 (Live Variables; cf. Example 3.3)

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	$\emptyset$	$\emptyset$	$\{y\}$	$\{x, y\}$	$\{z\}$	$\{z\}$	$\{x, y, z\}$

Example: Live Variables

Example 4.12 (Live Variables; cf. Example 3.3)

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	$\emptyset$	$\{y\}$	$\{x, y\}$	$\{x, y\}$	$\{y, z\}$	$\{y, z\}$	$\{x, y, z\}$

Example: Live Variables

Example 4.12 (Live Variables; cf. Example 3.3)

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	$\emptyset$	$\{y\}$	$\{x, y\}$	$\{x, y\}$	$\{y, z\}$	$\{y, z\}$	$\{x, y, z\}$