정적 분석의 샐행시간 전문화에 관한 연구

어현준,이광근

전산학과, KAIST

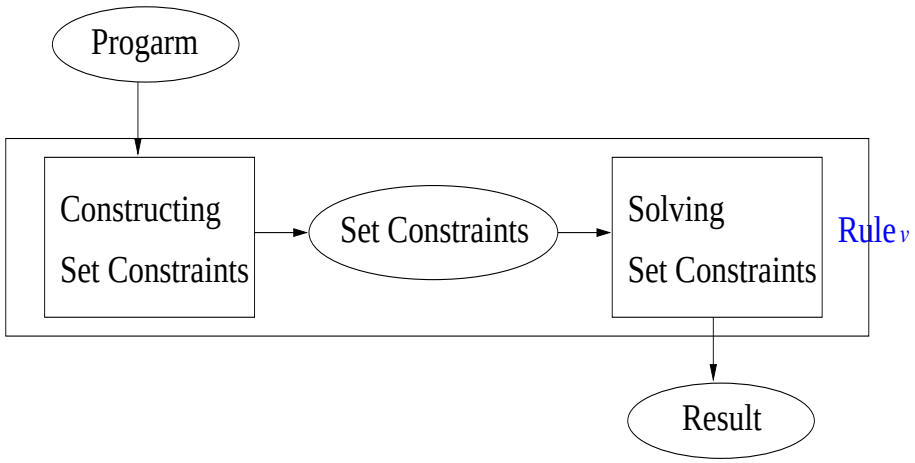
1999년 2월 22일

서론

- 프로그램의 실행중에 사용자의 입력 패턴에 따라서 프로그램 분석을 전문화(specialize)할 수 있다.
- 문제점 - 실행시간 부담
- 해결 방안 - 집합 기반 분석(Set-Based Analysis)의 틀 위에서 값을 자르는 분석(Static Value Slicing)을 사용

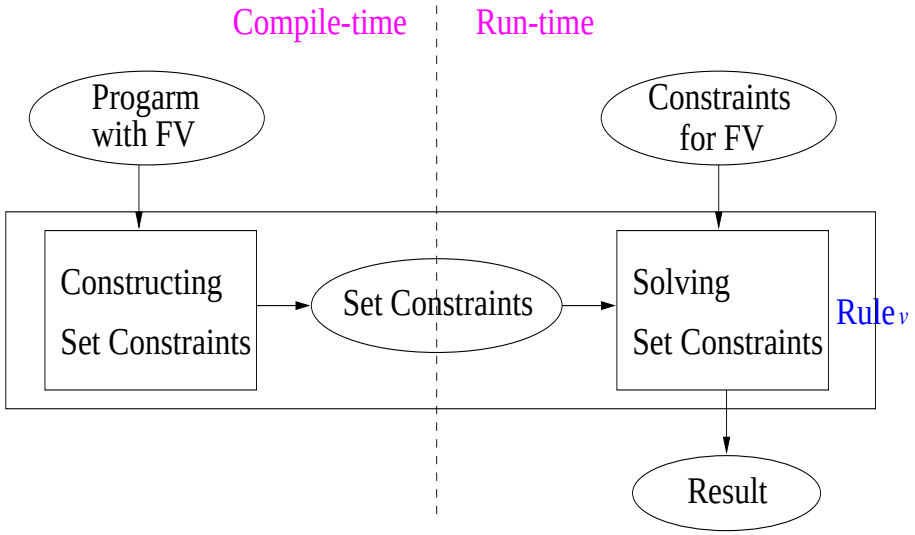
서론

- 일반적인 SBA(Set-Base Analysis) - 정적 분석



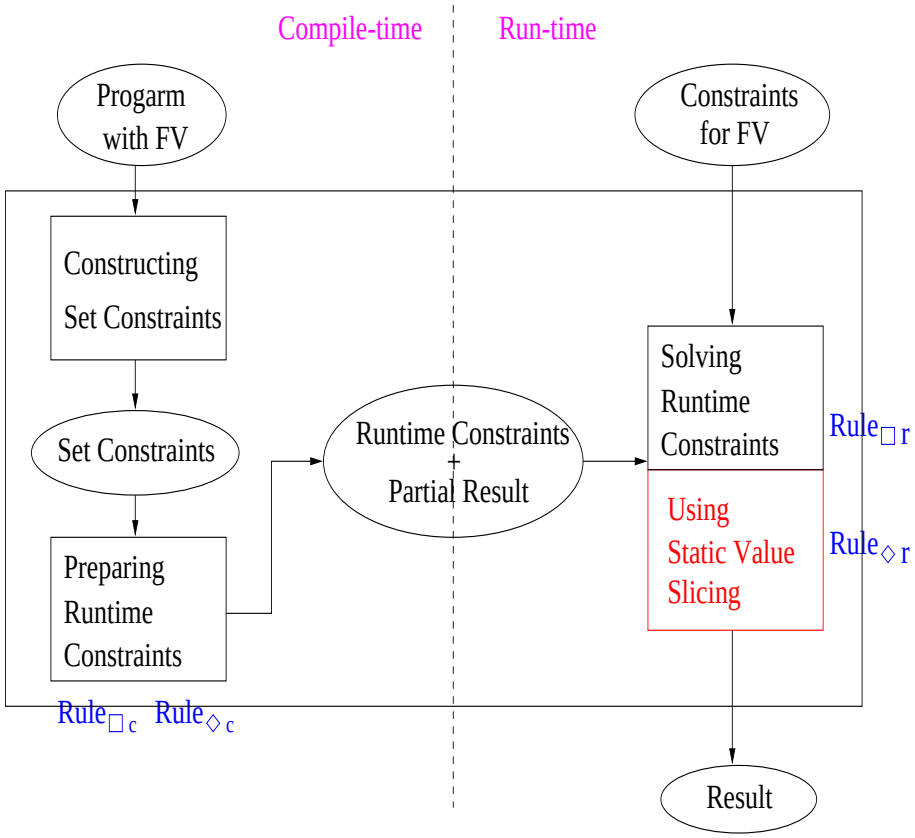
  - free variable은 $\top$(또는 $\mathcal{U}$, universe)으로 간주.

- specializing SBA - naive approach



서론

- specializing SBA - our approach



Language

e	$::=$	$\mathtt{1}$	unit
	$\mid$	x	variable
	$\mid$	ω	free variable
	$\mid$	$\lambda x.e$	abstraction
	$\mid$	$e_1\ e_2$	application
	$\mid$	$\mathtt{fix}\ f = \lambda x.e_1\ \mathtt{in}\ e_2$	recursive function binding
	$\mid$	$\kappa(e)$	data construction
	$\mid$	$\kappa^{-1}(e)$	data deconstruction
	$\mid$	$\mathtt{case}\ e_1\ \kappa\ e_2\ e_3$	switch

Set-Based Analysis

- Set Constraints

$$sc ::= \mathcal{X} \supseteq se$$

- Set expression

$$\begin{array}{rcl} ae & ::= & 1 \\ & | & \lambda x.e \\ & | & \kappa\, \mathcal{X} \end{array}$$

$$\begin{array}{rcl} se & ::= & ae \\ & | & \mathcal{X} \\ & | & \kappa^{-1}\, \mathcal{X} \\ & | & app(\mathcal{X}_1, \mathcal{X}_2) \\ & | & case(\mathcal{X}_1, \kappa, \mathcal{X}_2, \mathcal{X}_3) \end{array}$$

Set-Based Analysis

- Solving Set Constraints

$$\frac{\mathcal{X} \supseteq app(\mathcal{X}_1, \mathcal{X}_2) \quad \mathcal{X}_1 \supseteq \lambda x.e}{\mathcal{X} \supseteq \mathcal{X}_e \quad \mathcal{X}_x \supseteq \mathcal{X}_2}$$

$$\frac{\mathcal{X} \supseteq case(\mathcal{X}_1, \kappa, \mathcal{X}_2, \mathcal{X}_3) \quad \mathcal{X}_1 \supseteq \kappa\, \mathcal{Y}}{\mathcal{X} \supseteq \mathcal{X}_2}$$

$$\frac{\mathcal{X} \supseteq case(\mathcal{X}_1, \kappa, \mathcal{X}_2, \mathcal{X}_3) \quad \mathcal{X}_1 \supseteq \kappa'\, \mathcal{Y} \quad \kappa \neq \kappa'}{\mathcal{X} \supseteq \mathcal{X}_3}$$

$$\frac{\mathcal{X} \supseteq \kappa^{-1}\, \mathcal{Y} \quad \mathcal{Y} \supseteq \kappa\, \mathcal{Z}}{\mathcal{X} \supseteq \mathcal{Z}}$$

$$\frac{\mathcal{X} \supseteq \mathcal{Y} \quad \mathcal{Y} \supseteq ae}{\mathcal{X} \supseteq ae}$$

input a collection $\mathcal{C}$ of set constraints
repeat
    add set constraints according to rules in $Rule_v$
until no changes in $\mathcal{C}$
output $\{\mathcal{X} \supseteq ae \mid \mathcal{X} \supseteq ae \in \mathcal{C}\}$

Simplification Algorithm $\mathcal{S}$

Program Analysis with Input(Free Variable)

- with program $\wp$, free variable ω, $\rhd_1 \wp : \mathcal{C}$
- Compile-time(Static Analysis) ($\mathcal{A}_\mathcal{U}$)
  - Solving all constraints at compile-time
  - $\mathcal{S}(\mathcal{C} \cup \{\mathcal{X}_\omega \supseteq \mathcal{U}\})$
- Run-time ($\mathcal{A}_v$)
  - Solving all constraints at Run-time
  - ω의 값 v를 알고나서 분석을 시작.
  - $\mathcal{S}(\mathcal{C} \cup \{\mathcal{X}_\omega \supseteq v\})$
  - runtime overhead
- Compile-time Preparation / Run-time Solving ($\mathcal{A}_\Box$)
  - our approach
  - $\mathcal{S}_r(\mathcal{S}_c(\mathcal{C}) \cup \{\mathcal{X}_\omega \supseteq v\})$
  - static value slicing을 이용하여 더 빠르게 분석 ($\mathcal{A}_\diamond$)
- $\mathcal{A}_v \subseteq \mathcal{A}_\diamond \subseteq \mathcal{A}_\mathcal{U}$

Compile-time Preperation/Runtime Solving

- Run-time Constraints

$$\mathcal{X} \supseteq \square$$

- Run-time Set expression $\square$

$$\begin{array}{rcl} \square & ::= & \mathcal{X}_\omega \\ & | & \kappa_\square^{-1}\,\mathcal{X} \\ & | & app_\square(\mathcal{X}_1, \mathcal{X}_2) \\ & | & case_\square(\mathcal{X}_1, \kappa, \mathcal{X}_2, \mathcal{X}_3) \end{array}$$

Compile-time Preperation/Runtime Solving

- Compile-time Preparation of $\Box$

$$\frac{\mathcal{X} \supseteq app(\mathcal{X}_1, \mathcal{X}_2) \quad \mathcal{X}_1 \supseteq \lambda x.e}{\mathcal{X} \supseteq \mathcal{X}_e \quad \mathcal{X}_x \supseteq \mathcal{X}_2}$$

$$\frac{\mathcal{X} \supseteq case(\mathcal{X}_1, \kappa, \mathcal{X}_2, \mathcal{X}_3) \quad \mathcal{X}_1 \supseteq \kappa\, \mathcal{Y}}{\mathcal{X} \supseteq \mathcal{X}_2}$$

$$\frac{\mathcal{X} \supseteq case(\mathcal{X}_1, \kappa, \mathcal{X}_2, \mathcal{X}_3) \quad \mathcal{X}_1 \supseteq \kappa'\, \mathcal{Y} \quad \kappa \neq \kappa'}{\mathcal{X} \supseteq \mathcal{X}_3}$$

$$\frac{\mathcal{X} \supseteq \kappa^{-1}\, \mathcal{Y} \quad \mathcal{Y} \supseteq \kappa\, \mathcal{Z}}{\mathcal{X} \supseteq \mathcal{Z}}$$

$$\frac{\mathcal{X} \supseteq \mathcal{Y} \quad \mathcal{Y} \supseteq ae}{\mathcal{X} \supseteq ae}$$

$$\frac{\mathcal{X} \supseteq app(\mathcal{X}_1, \mathcal{X}_2) \quad \mathcal{X}_1 \supseteq \Box}{\mathcal{X} \supseteq app_{\Box}(\mathcal{X}_1, \mathcal{X}_2)}$$

$$\frac{\mathcal{X} \supseteq case(\mathcal{X}_1, \kappa, \mathcal{X}_2, \mathcal{X}_3) \quad \mathcal{X}_1 \supseteq \Box}{\mathcal{X} \supseteq case_{\Box}(\mathcal{X}_1, \kappa, \mathcal{X}_2, \mathcal{X}_3)}$$

$$\frac{\mathcal{X} \supseteq \kappa^{-1}\, \mathcal{Y} \quad \mathcal{Y} \supseteq \Box}{\mathcal{X} \supseteq \kappa_{\Box}^{-1}\, \mathcal{Y}}$$

$$\frac{\mathcal{X} \supseteq \mathcal{Y} \quad \mathcal{Y} \supseteq \Box}{\mathcal{X} \supseteq \Box}$$

Compile-time Preperation/Runtime Solving

- Solving $\Box$ at Runtime

$$\frac{\mathcal{X} \supseteq app_{\Box}(\mathcal{X}_1, \mathcal{X}_2) \quad \mathcal{X}_1 \supseteq \lambda x.e}{\mathcal{X} \supseteq \mathcal{X}_e \quad \mathcal{X}_x \supseteq \mathcal{X}_2}$$

$$\frac{\mathcal{X} \supseteq case_{\Box}(\mathcal{X}_1, \kappa, \mathcal{X}_2, \mathcal{X}_3) \quad \mathcal{X}_1 \supseteq \kappa\, \mathcal{Y}}{\mathcal{X} \supseteq \mathcal{X}_2}$$

$$\frac{\mathcal{X} \supseteq case_{\Box}(\mathcal{X}_1, \kappa, \mathcal{X}_2, \mathcal{X}_3) \quad \mathcal{X}_1 \supseteq \kappa'\, \mathcal{Y} \quad \kappa \neq \kappa'}{\mathcal{X} \supseteq \mathcal{X}_3}$$

$$\frac{\mathcal{X} \supseteq \kappa_{\Box}^{-1}\, \mathcal{Y} \quad \mathcal{Y} \supseteq \kappa\, \mathcal{Z}}{\mathcal{X} \supseteq \mathcal{Z}}$$

$$\frac{\mathcal{X} \supseteq \mathcal{Y} \quad \mathcal{Y} \supseteq ae}{\mathcal{X} \supseteq ae}$$

Static Value Slicing

- $\mathrm{SVS}(\mathcal{X}_e \subseteq V)(\omega)$
- 주어진 식 e가 만들어 내는 값이 자르는 기준 V에 포함되기 위해 ω가 가질 수 있는 값들의 조건
- 예

```
datatype t = A | B | C

case x of
     A => B
   | _ => x
```

$\mathrm{SVS}(\mathcal{X}_{\texttt{case}} \subseteq \texttt{B})(\texttt{x}) = \{\texttt{A}, \texttt{B}\}$
$\mathrm{SVS}(\mathcal{X}_{\texttt{case}} \subseteq \texttt{C})(\texttt{x}) = \{\texttt{C}\}$
$\mathrm{SVS}(\mathcal{X}_{\texttt{case}} \subseteq \texttt{A})(\texttt{x}) = \{\}$

Specialization via SVS

- set constraints

$$\mathcal{X} \supseteq \diamond$$

- set expression $\diamond$

$$\begin{array}{rcl} \diamond & ::= & \mathcal{X}_\omega \\ & | & \kappa_\diamond^{-1}(\mathcal{X}_\omega, S_1, S_2) \\ & | & app_\diamond(\mathcal{X}_\omega, S, \mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3) \\ & | & case_\diamond(\mathcal{X}_\omega, S_1, \mathcal{X}_1, S_2, \mathcal{X}_2) \end{array}$$

$$\begin{array}{lcl} \mathcal{I}(\kappa_\diamond^{-1}(\mathcal{X}_\omega, S_1, S_2)) & = & \{v \mid v \in S_2, \mathcal{X}_\omega \subseteq S_1\} \\ \mathcal{I}(app_\diamond(\mathcal{X}_\omega, S, \mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3)) & = & \{v \mid v \in \mathcal{X}_1, \mathcal{X}_\omega \subseteq S\} \\ & & \text{provided } \mathcal{X}_\omega \subseteq S \text{ implies } \mathcal{X}_2 \supseteq \mathcal{X}_3 \\ \mathcal{I}(case_\diamond(\mathcal{X}_\omega, S_1, \mathcal{X}_1, S_2, \mathcal{X}_2)) & = & \{v \mid v \in \mathcal{X}_1, \mathcal{X}_\omega \subseteq S_1\} \\ & & \cup \{v \mid v \in \mathcal{X}_2, \mathcal{X}_\omega \subseteq S_2\} \end{array}$$

Specialization via SVS

- Preparing $\diamond$

$$\frac{\mathcal{X} \supseteq app(\mathcal{X}_1, \mathcal{X}_2) \quad \mathcal{X}_1 \supseteq \diamond \quad \lambda x.e \in \wp}{\mathcal{X} \supseteq app_\diamond(\mathcal{X}_\omega, \mathrm{SVS}(\mathcal{X}_1 \subseteq \lambda x.e)(\omega), \mathcal{X}_e, \mathcal{X}_x, \mathcal{X}_2)}$$

$$\frac{\mathcal{X} \supseteq case(\mathcal{X}_1, \kappa, \mathcal{X}_2, \mathcal{X}_3) \quad \mathcal{X}_1 \supseteq \diamond}{\mathcal{X} \supseteq case_\diamond(\mathcal{X}_\omega, \mathrm{SVS}(\mathcal{X}_1 \subseteq \kappa\,\mathcal{U})(\omega), \mathcal{X}_2, \mathrm{SVS}(\mathcal{X}_1 \subseteq \overline{\kappa}\,\mathcal{U})(\omega), \mathcal{X}_3)}$$

$$\frac{\begin{array}{c}\mathcal{X} \supseteq \kappa^{-1}\,\mathcal{Y} \quad \mathcal{Y} \supseteq \diamond \\ S_1 = \mathrm{SVS}(\mathcal{Y} \subseteq \kappa\,\mathcal{U})(\omega) \\ S_2 = \{v \mid \kappa\, v \in \mathcal{S}(\{\mathcal{X}_\omega \supseteq S_1\} \cup \mathcal{C})(\mathcal{Y})\}\end{array}}{\mathcal{X} \supseteq \kappa_\diamond^{-1}(\mathcal{X}_\omega, S_1, S_2)}$$

where $\mathcal{C}$ is constraints constructed from program

$\mathcal{S}$ is a static set-based analysis

Specialization via SVS

- Solving $\diamond$

$$\frac{\mathcal{X} \supseteq app_\diamond(\mathcal{X}_\omega, S, \mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3) \quad \mathcal{X}_\omega \subseteq S}{\mathcal{X} \supseteq \mathcal{X}_1 \quad \mathcal{X}_2 \supseteq \mathcal{X}_3}$$

$$\frac{\mathcal{X} \supseteq case_\diamond(\mathcal{X}_\omega, S_1, \mathcal{X}_1, S_2, \mathcal{X}_2) \quad \mathcal{X}_\omega \subseteq S_1}{\mathcal{X} \supseteq \mathcal{X}_1}$$

$$\frac{\mathcal{X} \supseteq case_\diamond(\mathcal{X}_\omega, S_1, \mathcal{X}_1, S_2, \mathcal{X}_2) \quad \mathcal{X}_\omega \subseteq S_2}{\mathcal{X} \supseteq \mathcal{X}_2}$$

$$\frac{\mathcal{X} \supseteq \kappa_\diamond^{-1}(\mathcal{X}_\omega, S_1, S_2) \quad \mathcal{X}_\omega \subseteq S_1}{\mathcal{X} \supseteq S_2}$$

$$\frac{\mathcal{X} \supseteq \mathcal{Y} \quad \mathcal{Y} \supseteq ae}{\mathcal{X} \supseteq ae}$$

Example

- program

```
datatype t = A | B | κ1 of t | κ2 of t

(λx.case1 x of
      κ1 => case2 κ1^-1 x of
                  κ2 => A
                | _ => B
    | _ => x) ω
```

- set constraints constructed from program

$$\mathcal{X}_{app} \supseteq app(\mathcal{X}_{\lambda}, \mathcal{X}_{\omega})$$
$$\mathcal{X}_{\lambda} \supseteq \lambda x.case_1$$
$$\mathcal{X}_{case_1} \supseteq case(\mathcal{X}_x, \kappa_1, \mathcal{X}_{case_2}, \mathcal{X}_x)$$
$$\mathcal{X}_{case_2} \supseteq case(\mathcal{X}_{\kappa_1^{-1}}, \kappa_2, \mathcal{X}_A, \mathcal{X}_B)$$
$$\mathcal{X}_{\kappa_1^{-1}} \supseteq \kappa_1^{-1}(\mathcal{X}_x)$$
$$\mathcal{X}_A \supseteq A$$
$$\mathcal{X}_B \supseteq B$$

Example

- Static Analysis

	Constraints for ω
1	$\mathcal{X}_\omega \supseteq \mathcal{U}$
	from $\mathcal{X}_{app} \supseteq app(\mathcal{X}_\lambda, \mathcal{X}_\omega), \mathcal{X}_\lambda \supseteq \lambda x.case_1$
2	$\mathcal{X}_{app} \supseteq \mathcal{X}_{case_1}$
3	$\mathcal{X}_x \supseteq \mathcal{X}_\omega$
	from $1, 3$
4	$\mathcal{X}_x \supseteq \mathcal{U}$
	from $4, \mathcal{X}_{case_1} \supseteq case(\mathcal{X}_x, \kappa_1, \mathcal{X}_{case_2}, \mathcal{X}_x)$
5	$\mathcal{X}_{case_1} \supseteq \mathcal{X}_{case_2}$
6	$\mathcal{X}_{case_1} \supseteq \mathcal{X}_x$
	from $4, 6$
7	$\mathcal{X}_{case_1} \supseteq \mathcal{U}$
	from $2, 7$
8	$\mathcal{X}_{app} \supseteq \mathcal{U}$

Example

- Runtime Analysis

	Constraints for ω
a	$\mathcal{X}_\omega \supseteq \kappa_1 A$
	from $\mathcal{X}_{app} \supseteq app(\mathcal{X}_\lambda, \mathcal{X}_\omega), \mathcal{X}_\lambda \supseteq \lambda x.case_1$
b	$\mathcal{X}_{app} \supseteq \mathcal{X}_{case_1}$
c	$\mathcal{X}_x \supseteq \mathcal{X}_\omega$
	from a, c
d	$\mathcal{X}_x \supseteq \kappa_1 A$
	from $d, \mathcal{X}_{case_1} \supseteq case(\mathcal{X}_x, \kappa_1, \mathcal{X}_{case_2}, \mathcal{X}_x)$
e	$\mathcal{X}_{case_1} \supseteq \mathcal{X}_{case_2}$
	from $\mathcal{X}_{\kappa_1^{-1}} \supseteq \kappa_1^{-1}(\mathcal{X}_x)$
f	$\mathcal{X}_{\kappa_1^{-1}} \supseteq A$
	from $f, \mathcal{X}_{case_2} \supseteq case(\mathcal{X}_{\kappa_1^{-1}}, \kappa_2, \mathcal{X}_A, \mathcal{X}_B)$
g	$\mathcal{X}_{case_2} \supseteq \mathcal{X}_B$
	from $g, \mathcal{X}_B \supseteq B$
h	$\mathcal{X}_{case_2} \supseteq B$
	from e, h
i	$\mathcal{X}_{case_1} \supseteq B$
	from b, i
j	$\mathcal{X}_{app} \supseteq B$

Example

- Compile-time Preperation

	from $\mathcal{X}_{app} \supseteq app(\mathcal{X}_\lambda, \mathcal{X}_\omega), \mathcal{X}_\lambda \supseteq \lambda x.case_1$
1	$\mathcal{X}_{app} \supseteq \mathcal{X}_{case_1}$
2	$\mathcal{X}_x \supseteq \mathcal{X}_\omega$
	from 2, $\mathcal{X}_{case_1} \supseteq case(\mathcal{X}_x, \kappa_1, \mathcal{X}_{case_2}, \mathcal{X}_x)$
3	$\mathcal{X}_{case_1} \supseteq case_\diamond(\mathcal{X}_\omega, \mathrm{SVS}(\mathcal{X}_x \subseteq \kappa_1\mathcal{U})(\omega), \mathcal{X}_{app}, \mathrm{SVS}(\mathcal{X}_x \subseteq \overline{\kappa_1}\mathcal{U})(\omega), \mathcal{X}_x)$
	from 2, $\mathcal{X}_{\kappa_1^{-1}} \supseteq \kappa_1^{-1}(\mathcal{X}_x)$
4	$\mathcal{X}_{\kappa_1^{-1}} \supseteq {\kappa_1^{-1}}_\diamond(...)$
	from 4, $\mathcal{X}_{case_2} \supseteq case(\mathcal{X}_{\kappa_1^{-1}}, \kappa_2, \mathcal{X}_A, \mathcal{X}_B)$
5	$\mathcal{X}_{case_2} \supseteq case_\diamond(\mathcal{X}_\omega, \mathrm{SVS}(\mathcal{X}_y \subseteq \kappa_2\mathcal{U})(\omega), \mathcal{X}_A, \mathrm{SVS}(\mathcal{X}_y \subseteq \overline{\kappa_2}\mathcal{U})(\omega), \mathcal{X}_B)$

$$\mathrm{SVS}(\mathcal{X}_x \subseteq \kappa_1\mathcal{U})(\omega) = \{\kappa_1\mathcal{U}\}$$
$$\mathrm{SVS}(\mathcal{X}_x \subseteq \overline{\kappa_1}\mathcal{U})(\omega) = \{\kappa_2\mathcal{U}, A, B\}$$
$$\mathrm{SVS}(\mathcal{X}_x \subseteq \kappa_1\mathcal{U})(\omega) = \{\kappa_1\kappa_2\mathcal{U}\}$$
$$\mathrm{SVS}(\mathcal{X}_x \subseteq \overline{\kappa_1}\mathcal{U})(\omega) = \{\kappa_1\kappa_1\mathcal{U}, \kappa_1 A, \kappa_1 B\}$$

Example

- Runtime Solving

	Constraints for ω
a	$\mathcal{X}_\omega \supseteq \kappa_1 A$
	from $3, a$
b	$\mathcal{X}_{case_1} \supseteq \mathcal{X}_{case_2}$
	from $5, a$
c	$\mathcal{X}_{case_2} \supseteq \mathcal{X}_B$
	from $c, \mathcal{X}_B \supseteq B$
d	$\mathcal{X}_{case_2} \supseteq B$
	from b, d
e	$\mathcal{X}_{case_1} \supseteq B$
	from $1, e$
f	$\mathcal{X}_{app} \supseteq B$

결론

- 실행시간 입력을 바탕으로 분석을 전문화 하는 방법 제안
- 본 연구에서 제안한 분석 방법은 정적 분석보다 더 자세한 결과를 내면서 실행시간에 가질 수 있는 값을 모두 포함하는 안전한 분석이다.