



프로그램 따로분석의 이론적 기틀

이준협

2024년 4월 19일

ROPAS S&T

따로분석?

- 따로분석이란,
- 모든 것이 알려지기 전에,
- 프로그램의 의미를 포섭하는 분석이다.

왜 필요한가?

```
int *x;
void f(void) {
    if (*x == *x) g(); /* call to unknown function */
    *x = 42; /* what happens here? */
}
```

- 코드 전체가 주어지지 않는 일이 빈번하다.
ex. 외부 라이브러리 함수를 부르는 경우
- 미리 최대한 분석하고 싶다.

$$\llbracket e \rrbracket(\Sigma_0 \times \Sigma) \subseteq \Sigma_0 \times \llbracket e \rrbracket \Sigma$$

- Σ_0 : 내가 모르던 외부 환경
- $\llbracket e \rrbracket \Sigma$: 몰라도 최선을 다해 실행한 결과
- \times : 합치기(linking) 연산

따로분석 = 안전한 $\llbracket \cdot \rrbracket^\#$ + 안전한 $\times^\#$

$$\text{분석의 안전성} : \llbracket e \rrbracket(\Sigma_0 \times \Sigma) \subseteq \overbrace{\gamma(\sigma_0^\# \times^\# \underbrace{\llbracket e \rrbracket^\# \sigma^\#}_{\text{따로분석}})}^{\text{나중에 합치기}}$$

Identifiers	x	\in	Var	
Expression	e	\rightarrow	$x \mid \lambda x.e \mid e e$	λ -calculus
		\mid	$e \bowtie e$	linked expression
		\mid	ε	empty module
		\mid	$x = e ; e$	(recursive) binding

의미공간 소개

Environment	σ	\in	Env	
Location	ℓ	\in	Loc	
Value	v	\in	$\text{Val} \triangleq \text{Env} + \text{Var} \times \text{Expr} \times \text{Env}$	
Weak Value	w	\in	$\text{WVal} \triangleq \text{Val} + \underline{\text{Val}}$	
Environment	σ	\rightarrow	<ul style="list-style-type: none">•$(x, w) :: \sigma$$(x, \ell) :: \sigma$	<ul style="list-style-type: none">empty stackweak value bindingfree location binding
Value	v	\rightarrow	<ul style="list-style-type: none">σ$\langle \lambda x. e, \sigma \rangle$	<ul style="list-style-type: none">exported environmentclosure
Weak Value	w	\rightarrow	<ul style="list-style-type: none">v$\mu \ell. v$	<ul style="list-style-type: none">valuerecursive value

Event	E	\rightarrow	Init	initial environment
			Read(E, x)	read event
			Call(E, v)	call event
Environment	σ	\rightarrow	...	
			[E]	answer to an event
Value	v	\rightarrow	...	
			E	answer to an event

실행의미 소개

$$\sigma \vdash e \Downarrow v$$

$$\frac{\text{ID} \quad \sigma(x) = v}{\sigma \vdash x \Downarrow v}$$

$$\frac{\text{RECID} \quad \sigma(x) = \mu\ell.v}{\sigma \vdash x \Downarrow v[\mu\ell.v/\ell]}$$

$$\frac{\text{FN}}{\sigma \vdash \lambda x.e \Downarrow \langle \lambda x.e, \sigma \rangle}$$

$$\frac{\text{APP} \quad \sigma \vdash e_1 \Downarrow \langle \lambda x.e, \sigma_1 \rangle \quad \sigma \vdash e_2 \Downarrow v_2 \quad (x, v_2) :: \sigma_1 \vdash e \Downarrow v}{\sigma \vdash e_1 e_2 \Downarrow v}$$

$$\frac{\text{LINK} \quad \sigma \vdash e_1 \Downarrow \sigma_1 \quad \sigma_1 \vdash e_2 \Downarrow v}{\sigma \vdash e_1 \times e_2 \Downarrow v}$$

$$\frac{\text{EMPTY}}{\sigma \vdash \varepsilon \Downarrow \bullet}$$

$$\frac{\text{BIND} \quad \ell \notin \text{FLoc}(\sigma) \quad (x, \ell) :: \sigma \vdash e_1 \Downarrow v_1 \quad (x, \mu\ell.v_1) :: \sigma \vdash e_1 \Downarrow \sigma_2}{\sigma \vdash x = e_1; e_2 \Downarrow (x, \mu\ell.v_1) :: \sigma_2}$$

$$\begin{array}{c}
 \text{LINKEVENT} \\
 \frac{\sigma \vdash e_1 \Downarrow E \quad [E] \vdash e_2 \Downarrow v}{\sigma \vdash e_1 \times e_2 \Downarrow v}
 \end{array}
 \qquad
 \begin{array}{c}
 \text{APPEVENT} \\
 \frac{\sigma \vdash e_1 \Downarrow E \quad \sigma \vdash e_2 \Downarrow v}{\sigma \vdash e_1 e_2 \Downarrow \text{Call}(E, v)}
 \end{array}$$

$$\begin{array}{c}
 \text{BINDEVENT} \\
 \frac{\ell \notin \text{FLoc}(\sigma) \quad (x, \ell) :: \sigma \vdash e_1 \Downarrow v_1 \quad (x, \mu\ell.v_1) :: \sigma \vdash e_1 \Downarrow E_2}{\sigma \vdash x = e_1; e_2 \Downarrow (x, \mu\ell.v_1) :: [E_2]}
 \end{array}$$

합치기

$$\sigma_0 \bowtie \cdot \in \text{Event} \rightarrow 2^{\text{Val}}$$

$$\sigma_0 \bowtie \text{Init} \triangleq \{\sigma_0\}$$

$$\sigma_0 \bowtie \text{Read}(E, x) \triangleq \{v_+ | \sigma_+ \in \sigma_0 \bowtie E \wedge \sigma_+(x) = v_+\}$$

$$\cup \{v_+[\mu\ell.v_+/\ell] | \sigma_+ \in \sigma_0 \bowtie E \wedge \sigma_+(x) = \mu\ell.v_+\}$$

$$\sigma_0 \bowtie \text{Call}(E, v) \triangleq \{v'_+ | \langle \lambda x.e, \sigma_+ \rangle \in \sigma_0 \bowtie E \wedge v_+ \in \sigma_0 \bowtie v \wedge (x, v_+) :: \sigma_+ \vdash e \Downarrow v'_+\}$$

$$\cup \{\text{Call}(E_+, v_+) | E_+ \in \sigma_0 \bowtie E \wedge v_+ \in \sigma_0 \bowtie v\}$$

$$\sigma_0 \bowtie \cdot \in \text{Env} \rightarrow 2^{\text{Env}}$$

...

$$\sigma_0 \bowtie \cdot \in \text{Val} \rightarrow 2^{\text{Val}}$$

...

$$\sigma_0 \bowtie \cdot \in \text{WVal} \rightarrow 2^{\text{WVal}}$$

$$\sigma_0 \bowtie \mu\ell.v \triangleq \{\mu\ell'.v_+ | \ell' \notin \text{FLoc}(v) \cup \text{FLoc}(\sigma_0) \wedge v_+ \in \sigma_0 \bowtie v[\ell'/\ell]\}$$

$$\text{eval}(e, \sigma) \triangleq \{v \mid \sigma \vdash e \Downarrow v\} \quad \text{eval}(e, \Sigma) \triangleq \bigcup_{\sigma \in \Sigma} \text{eval}(e, \sigma) \quad \Sigma_0 \bowtie W \triangleq \bigcup_{\substack{\sigma_0 \in \Sigma_0 \\ w \in W}} (\sigma_0 \bowtie w)$$

Theorem (Advance)

$$\text{eval}(e, \Sigma_0 \bowtie \Sigma) \subseteq \Sigma_0 \bowtie \text{eval}(e, \Sigma)$$

- 증명은 $\sigma \vdash e \Downarrow v$ 에 대한 귀납법으로
- Coq으로 엄검증 완료! ✓

1. 증가하는 $\gamma \in 2^{\text{WVal}} \rightarrow \text{WVal}^\#$
 2. 안전한 $\text{eval}^\#$: $\Sigma_0 \subseteq \gamma(\sigma_0^\#) \Rightarrow \text{eval}(e, \Sigma_0) \subseteq \gamma(\text{eval}^\#(e, \sigma_0^\#))$
 3. 안전한 $\bowtie^\#$: $\Sigma_0 \subseteq \gamma(\sigma_0^\#)$ 이고 $W \subseteq \gamma(w^\#) \Rightarrow \Sigma_0 \bowtie W \subseteq \gamma(\sigma_0^\# \bowtie^\# w^\#)$
- $\Sigma_0 \subseteq \gamma(\sigma_0^\#)$ 이고 $\Sigma \subseteq \gamma(\sigma^\#) \Rightarrow \text{eval}(e, \Sigma_0 \bowtie \Sigma) \subseteq \gamma(\sigma_0^\# \bowtie^\# \text{eval}^\#(e, \sigma^\#))$

$$\Sigma_0 \subseteq \gamma(\sigma_0^\#)$$

이고

$$[\text{Init}] \in \gamma(\text{Init}^\#)$$

이면

$$\text{eval}(e_1 \bowtie e_2, \Sigma_0) \subseteq \gamma(\underbrace{\text{eval}^\#(e_1, \sigma_0^\#)}_{\text{따로}} \bowtie^\# \underbrace{\text{eval}^\#(e_2, \text{Init}^\#)}_{\text{따로}})$$

예시: CFA

Program point	p	\in	\mathbb{P}
Abstract event	$E^\#$	\in	$\text{Event}^\#$
Abstract environment	$\sigma^\#$	\in	$\text{Env}^\# \triangleq (\text{Var} \xrightarrow{\text{fin}} 2^{\mathbb{P}}) \times 2^{\text{Event}^\#}$
Abstract closure	$\langle \lambda x.p, p' \rangle$	\in	$\text{Clos}^\# \triangleq \text{Var} \times \mathbb{P} \times \mathbb{P}$
Abstract value	$v^\#$	\in	$\text{Val}^\# \triangleq \text{Env}^\# \times 2^{\text{Clos}^\#}$
Abstract semantics	$t^\#$	\in	$\mathbb{T}^\# \triangleq \mathbb{P} \rightarrow \text{Env}^\# \times \text{Val}^\#$
Abstract event	$E^\#$	\rightarrow	$\text{Init}^\# \mid \text{Read}^\#(p, x) \mid \text{Call}^\#(p, p)$

$$t^\# = p \mapsto (\sigma^\#, v^\#)$$

모든 p 마다, 입력 $\sigma^\#$ 와 출력 $v^\#$ 가 달려있음.

$$v^\# = ((x \mapsto \{p\}, \overbrace{\{\text{Init}^\#, \text{Read}^\#(p, x), \text{Call}^\#(p_1, p_2)\}}^{E^\# \text{ 부분}}), \underbrace{\{\langle \lambda x.p, p' \rangle\}}_{\text{Clos}^\# \text{ 부분}})$$

$\underbrace{\hspace{15em}}_{\sigma^\# \text{ 부분}}$

$x \mapsto \{p\}$: $\sigma(x)$ 가 ℓ^p 이거나 p 의 출력.

$\text{Read}^\#(p, x)$: p 에 들어오는 $[E]$ 에서 x 를 읽음.

$\text{Call}^\#(p_1, p_2)$: p_1 에서 나온 E 를 p_2 에서 나온 v 에 적용.

$\langle \lambda x.p, p' \rangle$: p' 에 들어온 σ 에서 실행되는 함수.

γ 정의하기

$$\sigma \preceq (\sigma^\#, t^\#)$$

$$\begin{array}{c} \text{CONC-NIL} \\ \hline \bullet \preceq \sigma^\# \end{array} \quad \begin{array}{c} \text{CONC-ENIL} \\ \hline E \preceq (\sigma^\#, \emptyset) \\ [E] \preceq \sigma^\# \end{array} \quad \begin{array}{c} \text{CONC-CONSLoc} \\ \hline p \in \sigma^\#.1(x) \quad \sigma \preceq \sigma^\# \\ (x, \ell^p) :: \sigma \preceq \sigma^\# \end{array} \quad \begin{array}{c} \text{CONC-CONSWVAL} \\ \hline p \in \sigma^\#.1(x) \quad w \preceq t^\#(p).2 \quad \sigma \preceq \sigma^\# \\ (x, w) :: \sigma \preceq \sigma^\# \end{array}$$

$$w \preceq (v^\#, t^\#)$$

$$\begin{array}{c} \text{CONC-CLOS} \\ \hline \langle \lambda x.p, p' \rangle \in v^\#.2 \quad \sigma \preceq t^\#(p').1 \\ \langle \lambda x.p, \sigma \rangle \preceq v^\# \end{array} \quad \begin{array}{c} \text{CONC-REC} \\ \hline v \preceq t^\#(p).2 \quad v \preceq v^\# \\ \mu \ell^p.v \preceq v^\# \end{array} \quad \begin{array}{c} \text{CONC-INIT} \\ \hline \text{Init}^\# \in v^\#.1.2 \\ \text{Init} \preceq v^\# \end{array}$$

$$\begin{array}{c} \text{CONC-READ} \\ \hline \text{Read}^\#(p, x) \in v^\#.1.2 \quad [E] \preceq t^\#(p).1 \\ \text{Read}(E, x) \preceq v^\# \end{array} \quad \begin{array}{c} \text{CONC-CALL} \\ \hline \text{Call}^\#(p_1, p_2) \in v^\#.1.2 \quad E \preceq t^\#(p_1).2 \quad v \preceq t^\#(p_2).2 \\ \text{Call}(E, v) \preceq v^\# \end{array}$$

γ 정의하기

$$\gamma(v^\#, t^\#) \triangleq \{w \mid w \preceq (v^\#, t^\#)\}$$

■ $t^\# \sqsupseteq \text{Step}^\#(t^\#)$ 이면

$$\sigma \vdash p \Downarrow v \Rightarrow \sigma \in \gamma(t^\#(p).1, t^\#) \Rightarrow v \in \gamma(t^\#(p).2, t^\#)$$

가 성립하는

- $\sigma_0 \in \gamma(\sigma_0^\#, t_0^\#)$ 인 $\sigma_0^\#, t_0^\#$ 와
- $t^\#$ 가 주어졌을 때,
- $t_+^\# \supseteq t_0^\#$ 이고 $t_+^\# \supseteq \text{Link}^\#(\sigma_0^\#, t^\#, t_+^\#)$ 이면

$$w_+ \in \sigma_0 \times w \Rightarrow w \in \gamma(t^\#(p).1, t^\#) \Rightarrow w_+ \in \gamma(t_+^\#(p).1, t_+^\#)$$

와

$$w_+ \in \sigma_0 \times w \Rightarrow w \in \gamma(t^\#(p).2, t^\#) \Rightarrow w_+ \in \gamma(t_+^\#(p).2, t_+^\#)$$

가 성립하는

$$\begin{aligned}\llbracket p_0 \rrbracket^\#(\sigma_0^\#, t_0^\#) &\triangleq \text{lfp}(\lambda t^\#. \text{Step}^\#(t^\#) \sqcup [p_0 \mapsto (\sigma_0^\#, \perp)] \sqcup t_0^\#) \\ (\sigma_0^\#, t_0^\#)_{\infty^\#} t^\# &\triangleq \text{lfp}(\lambda t_+^\#. \text{Link}^\#(\sigma_0^\#, t^\#, t_+^\#) \sqcup t_0^\#)\end{aligned}$$

감사합니다