모듈별 프로그램 따로 분석

이준협

2023년 6월 30일

ROPAS Show & Tell

Problem Statement

- Given two program segments e_1 and e_2,
- Want to derive a sound approximation of $[\![e_1 ! e_2]\!]$ utilizing information obtained from *separately* analyzing e_1 and e_2

Abstract Syntax

$$
\begin{array}{lcll}
x & \in & \text{ExprVar} & \\
M & \in & \text{ModVar} & \\
e & \in & \text{Expr} & \\
e & ::= & x & \textit{identifier, expression} \\
 & | & \lambda x.e & \textit{function} \\
 & | & e \;\; e & \textit{application} \\
 & | & e!e & \textit{linked expression} \\
 & | & \varepsilon & \textit{empty module} \\
 & | & M & \textit{identifier, module} \\
 & | & \texttt{let}\ x\ \ e\ \ e & \textit{let-binding, expression} \\
 & | & \texttt{let}\ M\ \ e\ \ e & \textit{let-binding, module}
\end{array}
$$

Operational Semantics : What Configurations Look Like

$$
\begin{array}{lcll}
t & \in & \mathbb{T} & \\
v & \in & \mathsf{Val}\,\mathbb{T} & \\
C & \in & \mathsf{Ctx}\,\mathbb{T} & \\
V & \in & \mathsf{Val}\,\mathbb{T} + \mathsf{Ctx}\,\mathbb{T} & \\
\sigma & \in & \mathsf{Mem}\,\mathbb{T} \triangleq \mathbb{T} \xrightarrow{\mathsf{fin}} \mathsf{Val}\,\mathbb{T} & \\
s & \in & \mathsf{Config}\,\mathbb{T} \triangleq \mathsf{Ctx}\,\mathbb{T} \times \mathsf{Mem}\,\mathbb{T} \times \mathbb{T} & \\
r & \in & \mathsf{Result}\,\mathbb{T} \triangleq (\mathsf{Val}\,\mathbb{T} + \mathsf{Ctx}\,\mathbb{T}) \times \mathsf{Mem}\,\mathbb{T} \times \mathbb{T} & \\
C & ::= & \texttt{[]} & \textit{hole} \\
 & \mid & \lambda x^t.C & \textit{param} \\
 & \mid & \texttt{let}\ x^t\ C & \textit{let } x \\
 & \mid & \texttt{let}\ M\ C\ C & \textit{let } M \\
v & ::= & \langle \lambda x.e, C \rangle & \textit{closure}
\end{array}
$$

Time?

Concrete Time

$(\mathbb{T}, \leq, \mathrm{tick})$ is a *concrete time* when

1. $(\mathbb{T}, \leq)$ is a total order.
2. $\mathrm{tick} \in \mathbb{T} \to \mathbb{T}$ satisfies: $\forall t \in \mathbb{T} : t < \mathrm{tick}\, t$.

Big-Step Evaluation Relation

$$\Downarrow \subseteq (\mathsf{Expr} \times \mathsf{Config}\ \mathbb{T}) \times \mathsf{Result}\ \mathbb{T}$$

Big-Step Evaluation Relation

$$\Downarrow \subseteq (\mathsf{Expr} \times \mathsf{Config}\ \mathbb{T}) \times \mathsf{Result}\ \mathbb{T}$$

$$[\textsc{ExprVar}]\ \frac{t_x = \mathsf{addr}(C, x) \qquad v = \sigma(t_x)}{(x, C, \sigma, t) \Downarrow (v, \sigma, t)}$$

Big-Step Evaluation Relation

$$\Downarrow \subseteq (\mathsf{Expr} \times \mathsf{Config}\ \mathbb{T}) \times \mathsf{Result}\ \mathbb{T}$$

$$[\textsc{ExprVar}]\ \frac{t_x = \mathsf{addr}(C, x) \qquad v = \sigma(t_x)}{(x, C, \sigma, t) \Downarrow (v, \sigma, t)}$$

$$[\textsc{App}]\ \frac{\begin{array}{c}(e_1, C, \sigma, t) \Downarrow (\langle \lambda x.e_\lambda, C_\lambda \rangle, \sigma_\lambda, t_\lambda) \\ (e_2, C, \sigma_\lambda, t_\lambda) \Downarrow (v, \sigma_a, t_a) \\ (e_\lambda, C_\lambda[\lambda x^{t_a}.[]], \sigma_a[t_a \mapsto v], \mathsf{tick}\ t_a) \Downarrow (v', \sigma', t')\end{array}}{(e_1\ e_2, C, \sigma, t) \Downarrow (v', \sigma', t')}$$

Big-Step Evaluation Relation

$$\Downarrow \subseteq (\mathsf{Expr} \times \mathsf{Config}\ \mathbb{T}) \times \mathsf{Result}\ \mathbb{T}$$

$$[\textsc{ExprVar}]\ \frac{t_x = \mathsf{addr}(C, x) \qquad v = \sigma(t_x)}{(x, C, \sigma, t) \Downarrow (v, \sigma, t)}$$

$$[\textsc{App}]\ \frac{\begin{array}{c}(e_1, C, \sigma, t) \Downarrow (\langle \lambda x.e_\lambda, C_\lambda \rangle, \sigma_\lambda, t_\lambda) \\ (e_2, C, \sigma_\lambda, t_\lambda) \Downarrow (v, \sigma_a, t_a) \\ (e_\lambda, C_\lambda[\lambda x^{t_a}.[]], \sigma_a[t_a \mapsto v], \mathsf{tick}\ t_a) \Downarrow (v', \sigma', t')\end{array}}{(e_1\ e_2, C, \sigma, t) \Downarrow (v', \sigma', t')}$$

$$[\textsc{Empty}]\ \frac{}{(\varepsilon, C, \sigma, t) \Downarrow (C, \sigma, t)}$$

Big-Step Reachability Relation

$$\rightsquigarrow \subseteq (\mathsf{Expr} \times \mathsf{Config}\ \mathbb{T}) \times (\mathsf{Expr} \times \mathsf{Config}\ \mathbb{T})$$

Big-Step Reachability Relation

$$\rightsquigarrow \subseteq (\mathsf{Expr} \times \mathsf{Config}\ \mathbb{T}) \times (\mathsf{Expr} \times \mathsf{Config}\ \mathbb{T})$$

$$[\textsc{LinkL}]\ \frac{}{(e_1!e_2, C, \sigma, t) \rightsquigarrow (e_1, C, \sigma, t)}$$

Big-Step Reachability Relation

$$\rightsquigarrow \subseteq (\mathsf{Expr} \times \mathsf{Config}\ \mathbb{T}) \times (\mathsf{Expr} \times \mathsf{Config}\ \mathbb{T})$$

$$[\textsc{LinkL}]\ \frac{}{(e_1 ! e_2, C, \sigma, t) \rightsquigarrow (e_1, C, \sigma, t)}$$

$$[\textsc{LinkR}]\ \frac{(e_1, C, \sigma, t) \Downarrow (C', \sigma', t')}{(e_1 ! e_2, C, \sigma, t) \rightsquigarrow (e_2, C', \sigma', t')}$$

Collecting Semantics

Collecting Semantics

The semantics for an expression e under configuration $s \in \text{Config } \mathbb{T}$ is an element in $(\text{Expr} \times \text{Config } \mathbb{T}) \rightarrow (\wp(\text{Result } \mathbb{T}))_{\perp}$ defined as:

$$[\![e]\!](s) \triangleq \bigsqcup_{(e,s) \rightsquigarrow^{*} (e',s')} [(e', s') \mapsto \{r | (e', s') \Downarrow r\}]$$

Example

The semantics for the non-terminating lambda expression $\Omega = (\lambda x.xx)(\lambda x.xx)$ satisfies:

$$[\![\Omega]\!]([\,], \emptyset, 0)(\Omega, [\,], \emptyset, 0) = \emptyset$$

Example

The semantics for the non-terminating lambda expression $\Omega = (\lambda x.xx)(\lambda x.xx)$ satisfies:

$$[\![\Omega]\!]([\,], \emptyset, 0)(\Omega, [\,], \emptyset, 0) = \emptyset$$

$$[\![\Omega]\!]([\,], \emptyset, 0)(Y, \_, \_, \_) = \bot$$

Concrete Linking

$$[\![e_1!e_2]\!](s) = [\![e_1]\!](s) \sqcup \bigsqcup_{s' \in [\![e_1]\!](s)(e_1,s)} ([\![e_2]\!](s') \sqcup [(e_1!e_2, s) \mapsto [\![e_2]\!](s')(e_2, s')])$$

(Linked expression) =(Exporting expression)$\sqcup$

(Consuming expression under the exported config)

Abstract Semantics : What Configurations Look Like

$$
\begin{array}{lcl}
t^\# & \in & \mathbb{T}^\# \\
v^\# & \in & \mathsf{Val}\,\mathbb{T}^\# \\
C^\# & \in & \mathsf{Ctx}\,\mathbb{T}^\# \\
V^\# & \in & \mathsf{Val}\,\mathbb{T}^\# + \mathsf{Ctx}\,\mathbb{T}^\# \\
\sigma^\# & \in & \mathsf{Mem}^\#\,\mathbb{T}^\# \triangleq \mathbb{T}^\# \xrightarrow{\mathsf{fin}} \wp(\mathsf{Val}\,\mathbb{T}^\#) \\
s^\# & \in & \mathsf{Config}^\#\,\mathbb{T}^\# \triangleq \mathsf{Ctx}\,\mathbb{T}^\# \times \mathsf{Mem}^\#\,\mathbb{T}^\# \times \mathbb{T}^\# \\
r^\# & \in & \mathsf{Result}^\#\,\mathbb{T}^\# \triangleq (\mathsf{Val}\,\mathbb{T}^\# + \mathsf{Ctx}\,\mathbb{T}^\#) \times \mathsf{Mem}^\#\,\mathbb{T}^\# \times \mathbb{T}^\#
\end{array}
$$

Time?

Definition (Abstract time)

$(\mathbb{T}^\#, \mathsf{tick}^\#)$ is an *abstract time* when
$\mathsf{tick}^\# \in \mathsf{Ctx}\,\mathbb{T}^\# \to \mathsf{Mem}^\#\,\mathbb{T}^\# \to \mathbb{T}^\# \to \mathsf{ExprVar} \to \mathsf{Val}\,\mathbb{T}^\# \to \mathbb{T}^\#$ is the policy for advancing the timestamp.

Big-Step Evaluation Relation

$$[\textsc{ExprVar}]\ \frac{{t_x}^\# = \mathsf{addr}(C^\#, x) \qquad v^\# \in \sigma^\#({t_x}^\#)}{(x, C^\#, \sigma^\#, t^\#) \Downarrow^\# (v^\#, \sigma^\#, t^\#)}$$

Big-Step Evaluation Relation

$$[\textsc{ExprVar}]\ \frac{{t_x}^{\#} = \mathsf{addr}(C^{\#}, x) \qquad v^{\#} \in \sigma^{\#}({t_x}^{\#})}{(x, C^{\#}, \sigma^{\#}, t^{\#}) \Downarrow^{\#} (v^{\#}, \sigma^{\#}, t^{\#})}$$

$$[\textsc{App}]\ \frac{\begin{array}{c}(e_1, C^{\#}, \sigma^{\#}, t^{\#}) \Downarrow^{\#} (\langle \lambda x.e_\lambda, {C_\lambda}^{\#}\rangle, {\sigma_\lambda}^{\#}, {t_\lambda}^{\#}) \\ (e_2, C^{\#}, {\sigma_\lambda}^{\#}, {t_\lambda}^{\#}) \Downarrow^{\#} (v^{\#}, {\sigma_a}^{\#}, {t_a}^{\#}) \\ (e_\lambda, {C_\lambda}^{\#}[\lambda x^{{t_a}^{\#}}.[]], {\sigma_a}^{\#}[{t_a}^{\#} \mapsto^{\#} v^{\#}], \mathsf{tick}^{\#}\ C^{\#}\ {\sigma_a}^{\#}\ {t_a}^{\#}\ x\ v^{\#}) \Downarrow^{\#} ({v'}^{\#}, {\sigma'}^{\#}, {t'}^{\#})\end{array}}{(e_1\ e_2, C^{\#}, \sigma^{\#}, t^{\#}) \Downarrow^{\#} ({v'}^{\#}, {\sigma'}^{\#}, {t'}^{\#})}$$

Abstract Semantics

The semantics for an expression e under configuration $s^\# \in \mathsf{Config}^\# \, \mathbb{T}^\#$ is an element in $(\mathsf{Expr} \times \mathsf{Config}^\# \, \mathbb{T}^\#) \to (\wp(\mathsf{Result}^\# \, \mathbb{T}^\#))_\perp$ defined as:

$$[\![e]\!]^\#(s^\#) \triangleq \bigsqcup_{(e,s^\#) \rightsquigarrow^{\#^*} (e',s'^\#)} [(e', s'^\#) \mapsto \{r^\# | (e', s'^\#) \Downarrow^\# r^\#\}]$$

$$[\![e]\!]^\#(s^\#) = \mathsf{lfp}(\lambda f^\#.F^\#([(e, s^\#) \mapsto \emptyset] \sqcup f^\#))$$

Transfer Function

Definition (Transfer function)

Given an element $f^\#$ of $(\mathsf{Expr} \times \mathsf{Config}^\# \, \mathbb{T}^\#) \to (\wp(\mathsf{Result}^\# \, \mathbb{T}^\#))_\perp$,

- Define $\Downarrow^\#_{f^\#}$ and $\rightsquigarrow^\#_{f^\#}$ by replacing all assumptions of the form $s^\# \Downarrow^\# r^\#$ to $r^\# \in f^\#(s^\#)$ in $\Downarrow^\#$ and $\rightsquigarrow^\#$.

We define the transfer function $F^\#$ by:

$$F^\#(f^\#) \triangleq f^\# \sqcup \bigsqcup_{\substack{(e, s^\#) \in \mathsf{dom}(f^\#) \\ (e, s^\#) \rightsquigarrow^\#_{f^\#} (e', s'^\#)}} [(e', s'^\#) \mapsto \{r^\# | (e', s'^\#) \Downarrow^\#_{f^\#} r^\#\}]$$

Soundness Given α

Definition (α-soundness between results)

- Let $(V, \sigma, t) \in \text{Result}\ \mathbb{T}$ and $(V^\#, \sigma^\#, t^\#) \in \text{Result}^\#\ \mathbb{T}^\#$.
- Let $\alpha : \mathbb{T} \to \mathbb{T}^\#$, and extend α to a function in $\text{Ctx}\ \mathbb{T} \to \text{Ctx}\ \mathbb{T}^\#$ by mapping α over all timestamps.
- Extend α to a function in $(\text{Ctx}\ \mathbb{T} + \text{Val}\ \mathbb{T}) \to (\text{Ctx}\ \mathbb{T}^\# + \text{Val}\ \mathbb{T}^\#)$
- Extend α to a function in $\text{Mem}\ \mathbb{T} \to \text{Mem}^\#\ \mathbb{T}^\#$ by defining

$$\alpha(\sigma) \triangleq \bigsqcup_{t \in \text{dom}(\sigma)} [\alpha(t) \mapsto \{\alpha(\sigma(t))\}]$$

We say that $(V^\#, \sigma^\#, t^\#)$ is an *α-sound approximation* of (V, σ, t) when $\alpha(V) = V^\#$, $\alpha(\sigma) \sqsubseteq \sigma^\#$, and $\alpha(t) = t^\#$.

Soundness

Definition (Soundness between semantics)

- Let $f \in (\mathsf{Expr} \times \mathsf{Config}\,\mathbb{T}) \to (\wp(\mathsf{Result}\,\mathbb{T}))_{\perp}$ and $f^{\#} \in (\mathsf{Expr} \times \mathsf{Config}^{\#}\,\mathbb{T}^{\#}) \to (\wp(\mathsf{Result}^{\#}\,\mathbb{T}^{\#}))_{\perp}$.

We say that $f^{\#}$ is a sound approximation of f if:

$$\begin{aligned}\forall e \in \mathsf{Expr}, s \in \mathsf{Config}\,\mathbb{T}, r \in \mathsf{Result}\,\mathbb{T} \ :\\ r \in f(e, s) \Rightarrow\\ \exists \alpha, \alpha', s^{\#}, r^{\#} \ : \ \alpha(s) \sqsubseteq s^{\#} \wedge \alpha'(r) \sqsubseteq r^{\#} \in f^{\#}(e, s^{\#})\end{aligned}$$

Why?

Preservation of soundness

- Let $s \in \text{Config}\ \mathbb{T}$ and $s^{\#} \in \text{Config}^{\#}\ \mathbb{T}^{\#}$.
- Let all timestamps in the C and σ component of s be strictly less than the t component.
- Let $s^{\#}$ be an α-sound approximation of s for some α.

Then for all e, $[\![e]\!]^{\#}(s^{\#})$ is a sound approximation of $[\![e]\!](s)$.

Abstract Linking

Define an *injection* operator that, given $s^\# \in \mathsf{Config}^\#\ \mathbb{T}^\#, r^\# \in \mathsf{Result}^\#\ \mathbb{T}'^\#$, gives $s^\# \triangleright r^\# \in \mathsf{Result}^\#\ (\mathbb{T}^\# + \mathbb{T}'^\#)$, that satisfies:

1. $\alpha(s) \sqsubseteq s^\# \Rightarrow \exists \alpha' : \alpha'(s) \sqsubseteq (s^\# \triangleright \emptyset)$.
2. $\exists \mathsf{tick}_+^\#$ such that $(\mathbb{T}^\# + \mathbb{T}'^\#, \mathsf{tick}_+^\#)$ is an abstract time, and

$$s^\# \triangleright \llbracket e \rrbracket^\#(s'^\#) \sqsubseteq \llbracket e \rrbracket^\#(s^\# \triangleright s'^\#)$$

Abstract Linking

Recall:

$$\llbracket e_1!e_2 \rrbracket(s) = \llbracket e_1 \rrbracket(s) \sqcup \bigsqcup_{s' \in \llbracket e_1 \rrbracket(s)(e_1,s)} (\llbracket e_2 \rrbracket(s') \sqcup [(e_1!e_2, s) \mapsto \llbracket e_2 \rrbracket(s')(e_2, s')])$$

Abstract Linking

Recall:

$$[\![e_1!e_2]\!](s) = [\![e_1]\!](s) \sqcup \bigsqcup_{s' \in [\![e_1]\!](s)(e_1,s)} \left([\![e_2]\!](s') \sqcup [(e_1!e_2, s) \mapsto [\![e_2]\!](s')(e_2, s')]\right)$$

1. Given a sound approximation $f^\#$ of $[\![e_1]\!](s)$ under $\mathbb{T}^\#$, extract a set containing α-sound approximations of all exported configurations s'.

Abstract Linking

Recall:

$$[\![e_1!e_2]\!](s) = [\![e_1]\!](s) \sqcup \bigsqcup_{s' \in [\![e_1]\!](s)(e_1,s)} \left([\![e_2]\!](s') \sqcup [(e_1!e_2, s) \mapsto [\![e_2]\!](s')(e_2, s')]\right)$$

1. Given a sound approximation $f^\#$ of $[\![e_1]\!](s)$ under $\mathbb{T}^\#$, extract a set containing α-sound approximations of all exported configurations s'.
2. Inject the exported context onto the separately analyzed results $[\![e_2]\!]^\#(\emptyset)$, then perform the fixpoint computation starting from there to obtain $[\![e_2]\!]^\#(s'^\#)$ which is a sound approximation of $[\![e_2]\!](s')$.

Why Can We Compute $[\![e]\!]^\#$?

Theorem (Finiteness of time implies finiteness of abstraction)

If $\mathbb{T}^\#$ *is finite,*

$$\forall e, s^\# \ : \ |[\![e]\!]^\#(s^\#)| < \infty$$

감사합니다