Extending Separation Logic with Fixpoints and Postponed Substitution

Élodie-Jane Sims
Elodie-Jane.Sims@polytechnique.fr

Motivations

- We want to do Pointer analyses: check dereferencing problems, aliases,...
- Separation Logic allows descriptions of properties of the memory, e.g.
  - x points to a list of [1;2;3]
    $\exists x_2, x_3.\ (x \hookrightarrow 1, x_2) * (x_2 \hookrightarrow 2, x_3) * (x_3 \hookrightarrow 3, \mathtt{nil})$
  - x and y are aliased pointers
    $x = y \wedge \exists x_1, x_2.\ (x \hookrightarrow x_1, x_2)$
  - Partitioning: x and y belong to two disjoint pieces of a heap

➤ We want to use this logic as an interface language between analyses, or as an intermediate language between the program and other analyses

Results

- **But** currently, recursive formulae could not be expressed within the logic, and moreover pre-conditions (wlp) and post-conditions (sp) for `while` loops could not be expressed

➤ So we added to the logic fixpoints and postponed substitution, and expressed the wlp and sp for any command and any formula and proved their correctness.

Plan

- Programs
- The extended logic: $BI^{\mu\nu}$
- Backward Analysis: wlp,
  Forward analysis: sp
- Conclusions

Semantic domain: $Memory$

$$
\begin{array}{rcll}
Val & = & Int \cup Bool \cup Atoms \cup Loc & Values \\
S & = & Var \rightharpoonup Val & Stacks \\
H & = & Loc \rightharpoonup Val \times Val & Heaps \\
Memory & = & S \times H &
\end{array}
$$

We write s, h as well as m for elements of $Memory$.

Commands

$$
\begin{array}{rcl}
C & ::= & x := E \\
& | & x := E.i \\
& | & E.i := E' \\
& | & x := \texttt{cons}(E_1, E_2) \\
& | & \texttt{dispose}(E) \\
& | & C_1; C_2 \\
& | & \texttt{if}\ \ E\ \ \texttt{then}\ \ C_1\ \ \texttt{else}\ \ C_2 \\
& | & \texttt{while}\ \ E\ \ \texttt{do}\ \ C_1 \\
& | & \texttt{skip}
\end{array}
$$

$$
E \quad ::= \quad x \mid n \mid \texttt{nil} \mid True \mid False \mid E_1\ \ op\ \ E_2
$$

$n \in \mathbb{Z},\ i \in \{1, 2\}$

Operational semantics of commands

$$C, m \rightsquigarrow C', m'$$
$$C, m \rightsquigarrow m'$$

Example for cons

$$\frac{l \in Loc, \quad l \notin dom(h) \cup range(h) \cup range(s), \quad v_1 = [\![E_1]\!]^s, v_2 = [\![E_2]\!]^s}{x := \mathtt{cons}(E_1, E_2), s, h \rightsquigarrow [s|x \to l], [h|l \to \langle v_1, v_2 \rangle]}$$

The extended logic: $BI^{\mu\nu}$

	Classical	*connectives*				
	$E = E'$		$\mid$	`false`		
$\mid$	$P \Rightarrow Q$		$\mid$	$\exists x.P$		
	Spatial	*connectives*				
$\mid$	`emp`	Empty Heap	$\mid$	$E \mapsto E_1, E_2$	Points to
$\mid$	$P * Q$	Spatial Conj.	$\mid$	$P \mathrel{-\!\!*} Q$	Spatial Imp.
	Fixpoint	*connectives*				
$\mid$	X_v	Formula Variable	$\mid$	$P[E/x]$	Postponed Substitution
$\mid$	$\boldsymbol{\nu} X_v . P$	Greatest Fixpoint	$\mid$	$\boldsymbol{\mu} X_v . P$	Least Fixpoint

$Var_v = \{X_v, Y_v, ...\}$ an infinite set of formula variables

Operational semantics of formulae

$$\rho : Var_v \rightharpoonup \mathcal{P}(Memory) \text{ : environment}$$

$$[\![P]\!]_\rho \in \mathcal{P}(Memory) \text{ : semantics}$$

$$m \models P \text{ iff } m \in [\![P]\!]_\emptyset$$

$$P \equiv Q \text{ iff } \forall \rho.([\![P]\!]_\rho = [\![Q]\!]_\rho) \vee ([\![P]\!]_\rho \text{ and } [\![Q]\!]_\rho \text{ both do not exist})$$

Classical connectives semantics

$$
\begin{array}{rcl}
[\![E = E']\!]_\rho & = & \{s, h \mid [\![E]\!]^s = [\![E']\!]^s\} \\
[\![\mathtt{false}]\!]_\rho & = & \emptyset \\
[\![P \Rightarrow Q]\!]_\rho & = & (\mathit{Memory} \setminus [\![P]\!]_\rho) \cup [\![Q]\!]_\rho \\
[\![\exists x.\ P]\!]_\rho & = & \{s, h \mid \exists v \in \mathit{Val}.\ [s \mid x \to v], h \in [\![P]\!]_\rho\}
\end{array}
$$

Spatial connectives semantics

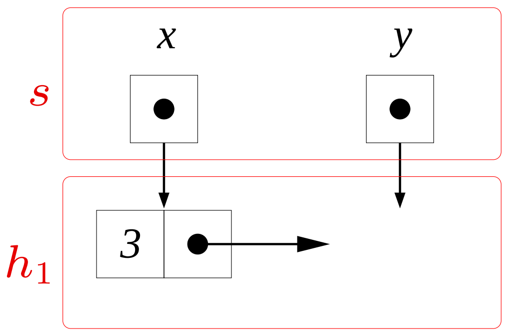
$$
\begin{array}{rcl}
[\![\texttt{emp}]\!]_\rho & = & \{s, h \mid dom(h) = \emptyset\} \\
[\![E \mapsto E_1, E_2]\!]_\rho & = & \{s, h \mid dom(h) = \{[\![E]\!]^s\} \text{ and } h([\![E]\!]^s) = \langle [\![E_1]\!]^s, [\![E_2]\!]^s \rangle\} \\
[\![P * Q]\!]_\rho & = & \{s, h_0 \cdot h_1 \mid h_0 \sharp h_1,\ \ s, h_0 \in [\![P]\!]_\rho \text{ and } s, h_1 \in [\![Q]\!]_\rho\} \\
[\![P \mathrel{-\!\!*} Q]\!]_\rho & = & \{s, h \mid \forall \boldsymbol{h'}, \text{ if } h \sharp h' \text{ and } s, h' \in [\![P]\!]_\rho \text{ then } s, h \cdot h' \in [\![Q]\!]_\rho\}
\end{array}
$$

$h \sharp h'$: $dom(h)$ and $dom(h')$ are disjoint
$h \cdot h'$: union of disjoint heaps h and h'

Examples of formulae

Ex. 1

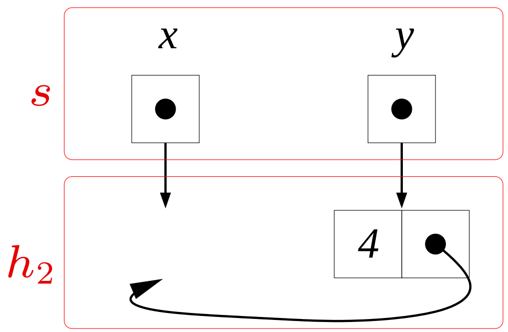
$$s = [x \to l_1, y \to l_2]$$
$$h_1 = [l_1 \to \langle 3, l_2 \rangle]$$



$$\models \quad (x \mapsto 3, y)$$
$$\models \quad (y \mapsto 4, x) \; -\!\!\circledast \; ((x \mapsto 3, y) * (y \mapsto 4, x))$$

Ex. 2

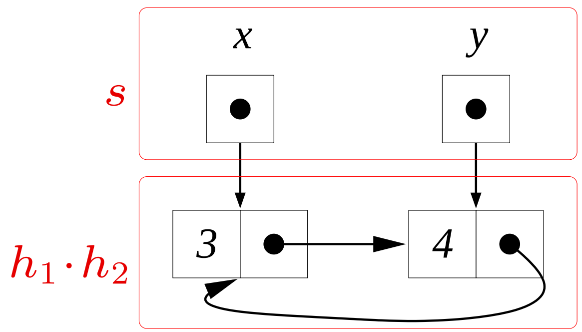
$$s = [x \to l_1, y \to l_2]$$
$$h_2 = [l_2 \to \langle 4, l_1 \rangle]$$



$$\models (y \mapsto 4, x)$$

Ex. 3

$$s = [x \to l_1, y \to l_2]$$
$$h_1 \cdot h_2 = \begin{bmatrix} l_1 \to \langle 3, l_2 \rangle, \\ l_2 \to \langle 4, l_1 \rangle \end{bmatrix}$$



$$\models (x \mapsto 3, y) * (y \mapsto 4, x)$$
$$\not\models (x \mapsto 3, y) \wedge (y \mapsto 4, x)$$

Fixpoint connectives semantics

$$
\begin{array}{rcl}
[\![X_v]\!]_\rho & = & \rho(X_v) \text{ if } X_v \in dom(\rho) \\
[\![\mu X_v.P]\!]_\rho & = & \mathrm{lfp}^{\subseteq}_{\emptyset} \lambda Y.\ [\![P]\!]_{[\rho \mid X_v \to Y]} \\
[\![\nu X_v.P]\!]_\rho & = & \mathrm{gfp}^{\subseteq}_{\emptyset} \lambda Y.\ [\![P]\!]_{[\rho \mid X_v \to Y]} \\
[\![P[E/x]]\!]_\rho & = & \left\{ s, h \mid \begin{array}{l} [\![E]\!]^s \text{ exists and} \\ [s \mid x \to [\![E]\!]^s], h \in [\![P]\!]_\rho \end{array} \right\}
\end{array}
$$

Remarks:

- $[\![\ ]\!]$ may not be defined: e.g. $[\![X_v]\!]_\emptyset$, $[\![\mu X_v.\neg X_v]\!]_\rho$
- $[\![\mathtt{true}[y/x]]\!] = \{s, h \mid [\![y]\!]^s \text{ exists } \}$ Postponed substitution connective
  $[\![\mathtt{true}\{y/x\}]\!] = [\![\mathtt{true}]\!] = \mathit{Memory}$ Capture avoiding substitution

Example: List formula

"x points to a finite non-cyclic list of integers"

$\texttt{nclist}(x) \triangleq \boldsymbol{\mu X_v}.\ (x = \texttt{nil}) \vee \exists x_1, x_2.(\texttt{isint}(x_1) \wedge (x \mapsto x_1, x_2 \ * \ \boldsymbol{X_v[x_2/x]}))$

$\texttt{isint}(x) \triangleq \exists n. n = x + 1$

Notice the combination of fixpoint and postponed substitution to write recursive definitions
"$\texttt{nclist}(x) = (x = \texttt{nil}) \vee \exists x_1, x_2.\texttt{isint}(x_1) \wedge (x \mapsto x_1, x_2 \ * \ \texttt{nclist}(x_2))$"

Some properties of the logic

- $[\ /\ ]$ is not $\{\ /\ \}$
  with
  $\{\ /\ \}$: capture avoiding substitution
  $[\ /\ ]$: postponed substitution connective

- Unfolding theorems holds
  $\mu X_v.\ P \equiv P\{\mu X_v.\ P/X_v\}$
  $\nu X_v.\ P \equiv P\{\nu X_v.\ P/X_v\}$

- $\{\ /\ \}$: no variable renaming theorem (see next slides)

- some simplification on $[\ /\ ]$ (see next slides)

$\{\ /\ \}$: no variable renaming theorem

$$\exists y.P \not\equiv \exists z.P\{z/y\}$$

with $z \notin Var(P)$ *(when $y \neq z$)*

Counter examples:

- $[\![\nu X_v.y = 3 \wedge \exists \boldsymbol{y}.(X_v \wedge \boldsymbol{y} = 5)]\!]_\emptyset = \emptyset \quad \neq \quad [\![\nu X_v.y = 3 \wedge \exists \boldsymbol{z}.(X_v \wedge \boldsymbol{z} = 5)]\!]_\emptyset = [\![y = 3]\!]_\emptyset$
- $[\![\exists \boldsymbol{y}.\nu X_v.\boldsymbol{y} = 3 \wedge \exists y.(X_v \wedge y = 5)]\!]_\emptyset = \emptyset \quad \neq \quad [\![\exists \boldsymbol{z}.\nu X_v.\boldsymbol{z} = 3 \wedge \exists y.(X_v \wedge y = 5)]\!]_\emptyset = Memory$

Definition of full substitution

$\{[\ /\ ]\}$: full syntactical variable substitution

$P\{[z/y]\}$ is P in which all y are replaced by z wherever they occur,

for example:

$$(\exists y.P)\{[z/y]\} \triangleq \exists z.(P\{[z/y]\})$$

$$(P[E/x])\{[z/y]\} \triangleq (P\{[z/y]\})[E\{z/y\}/x\{z/y\}]$$

Variable renaming theorem for $BI^{\mu\nu}$

If

- P is υ-closed (variables in Var_{υ} are all closed by μ or ν)
- $z \notin Var(P)$
- $y \notin FV(P)$

then

$$P \equiv P\{[z/y]\}$$

in particular $\exists y.P \equiv \exists z.(P\{[z/y]\})$

Equivalences on $[\ /\ ]$

– If P has no $\mu, \nu, X_v, [\ /\ ]$ then

$$P[E/x] \equiv P\{E/x\} \wedge is(E)$$

in particular $(\exists x.P)[E/x] \equiv (\exists x.P) \wedge is(E)$.

– If $\left[\begin{array}{l} P \text{ is } v\text{-closed} \\ x_1 \notin Var(E) \\ x_1 \neq x_2 \end{array}\right.$ then $(\exists x_1.P)[E/x_2] \equiv \exists x_1.(P[E/x_2])$

– $(A \vee C)[E/x] \equiv (A[E/x]) \vee (C[E/x])$

– If $y \notin Var(P)$ then $\begin{array}{l}(\mu X_v.P)[y/x] \equiv (\mu X_v.P\{[y/x]\}) \wedge is(y) \\ (\nu X_v.P)[y/x] \equiv (\nu X_v.P\{[y/x]\}) \wedge is(y)\end{array}$

To understand the last rule, we can come back to the program point of view seeing

- fixpoints as `while` loops
- $[\ /\ ]$ as assignments

C	$wlp(\mathtt{true}, C)$
$x := w;$ `while` $x = y$ `do` $x := x + 1$	$(\nu X_v.(x \neq y) \vee ((x = y) \wedge X_v[x + 1/x]))[w/x]$
$\equiv$	$\equiv$
`while` $w = y$ `do` $w := w + 1$	$(\nu X_v.(w \neq y) \vee ((w = y) \wedge X_v[w + 1/w])) \wedge is(w)$

Backward Analysis: wlp

wlp : weakest liberal precondition, such that

$$[\![wlp(P, C)]\!]_\emptyset = \left\{ m \mid \begin{array}{l} -\ C, m \text{ cannot run to an error} \\ -\ \{m' \mid C, m \leadsto^* m'\} \subseteq [\![P]\!]_\emptyset \end{array} \right\}$$

wlp is expressed for any P and any C

- $wlp(P, x := E) = P[E/x]$

- $wlp(P, \texttt{while } E \texttt{ do } C) = \boldsymbol{\nu X_v}.\ ((E = \texttt{true} \wedge wlp(\boldsymbol{X_v}, C)) \vee (E = \texttt{false} \wedge P))$

Remark: $[\![wlp(\texttt{true}, C)]\!]_\emptyset = \{m \mid C, m \text{ cannot run to an error}\}$

Forward Analysis: sp

sp : strongest postcondition, such that

$$[\![sp(P,C)]\!]_\emptyset = \{m' \mid \exists m \in [\![P]\!]_\emptyset .\ C, m \rightsquigarrow^* m'\}$$

sp is expressed for any P and any C

- $sp(P, x := E) = \exists x'.\ P[x'/x] \wedge x = E\{x'/x\}$ with $x' \notin FV(E,P)$
- $sp(P, \texttt{while } E \texttt{ do } C) = (E = \texttt{false}) \wedge (\boldsymbol{\mu X_v}.sp(\boldsymbol{X_v} \wedge E = \texttt{true}, C) \vee P)$

Conclusions

- $\surd$ We have extended separation logic so that recursive formulae can be expressed within the logic, can be used to instantiate existing and new triple-rules
- $\surd$ We have a backward (wlp) and a forward (sp) analyses with the proofs of their correctness for any formula and any command including `while` loops
- ➟ The use of separation logic as an intermediate or interface language is still an ongoing work...

Example : unfolding `nclist42`

$\texttt{nclist42}(x) \triangleq \mu X_v.(x = \texttt{nil}) \vee \exists x_2.((x \mapsto 42, x_2) * X_v[x_2/x])$ (with $x_2 \neq x$).
Then
$\texttt{nclist42}(x_2) = \mu X_v.(x_2 = \texttt{nil}) \vee \exists x_3.((x_2 \mapsto 42, x_3) * X_v[x_3/x_2])$ (with $x_3 \neq x_2$).

We can prove that $X_v[x_2/x]$ is equivalent to $\texttt{nclist42}(x_2)$.

$$
\begin{array}{lcl}
\texttt{nclist42}(x) & \triangleq & \mu X_v.(x = \texttt{nil}) \vee \exists x_2.((x \mapsto 42, x_2) * X_v[x_2/x]) \\
(unfolding) & = & (x = \texttt{nil}) \vee \exists x_2.((x \mapsto 42, x_2) * ((\mu X_v.(x = \texttt{nil}) \\
 & & \quad \vee \exists x_2.((x \mapsto 42, x_2) * X_v[x_2/x]))[x_2/x])) \\
(\text{variable renaming for } BI^{\mu\nu}) & = & (x = \texttt{nil}) \vee \exists x_2.((x \mapsto 42, x_2) * ((\mu X_v.(x = \texttt{nil}) \\
 & & \quad \vee \exists x_3.((x \mapsto 42, x_3) * X_v[x_3/x]))[x_2/x])) \\
(\text{simplification } [\ /\ ] \text{ case } \mu) & = & (x = \texttt{nil}) \vee \exists x_2.((x \mapsto 42, x_2) * (\mu X_v.(x_2 = \texttt{nil}) \\
 & & \quad \vee \exists x_3.((x_2 \mapsto 42, x_3) * X_v[x_3/x_2]))) \\
 & \triangleq & (x = \texttt{nil}) \vee \exists x_2.((x \mapsto 42, x_2) * \texttt{nclist42}(x_2))
\end{array}
$$

Expressions semantics

$$[\![x]\!]^s \triangleq s(x),\ [\![42]\!]^s \triangleq 42,\ [\![\mathtt{false}]\!]^s \triangleq false,\ [\![E_1 + E_2]\!]^s \triangleq [\![E_1]\!]^s + [\![E_2]\!]^s,\ \ldots$$

Triples

$$\{P\}C\{P'\} \text{ iff}$$

$$\begin{array}{ll} \forall m & \text{if } m \models P \text{ then} \\ & \text{- } C \text{ can be executed from } m \text{ without error} \\ & \text{- if } C, m \rightsquigarrow^* m' \text{ then } m' \models P' \end{array}$$

This definition differ from the usual one of Hoare triples.
(If $m \models P$ and $C, m \rightsquigarrow^* m'$ then $m' \models P'$)

In particular with our definition, if $\{P\}C\{\mathtt{true}\}$, then C can be executed without error from any memory satisfying P.

Backward Analysis: wlp

wlp : weakest liberal precondition, such that

$$\{wlp(P, C)\}C\{P\}$$

wlp is expressed for any P and any C

- $\{P[E/x]\}x := E\{P\}$
- $\{\nu X_v.\ ((E = \mathtt{true} \wedge wlp(X_v, C)) \vee (E = \mathtt{false} \wedge P))\}\ \mathtt{while}\ E\ \mathtt{do}\ C\ \{P\}$

Forward analysis: sp

We would like to have sp such that:

$$\{P\}C\{sp(P,C)\}$$

But it may happens that :

$$\nexists Q.\ \{P\}C\{Q\}$$

e.g. : $\forall Q.\ \neg(\{\mathtt{true}\}x := \mathtt{nil}; x.1 := 3\{Q\})$

➜ A two steps analysis

① Express the conditions of no error $wlp(\texttt{true}, C)$

② Express the sp for any P and any C such that

If $m \models P$ and $C, m \leadsto^* m'$ then $m' \models sp(P, C)$.
(the usual definition of Hoare triples).

We then have:

$$\{P \wedge wlp(\texttt{true}, C)\}C\{sp(P, C)\}$$

$$sp(P,\ \texttt{while}\ E\ \texttt{do}\ C) =\ (E = \texttt{false}) \wedge (\mu X_v.sp(X_v \wedge E = \texttt{true}, C) \vee P)$$

Difference between $*$ and $\wedge$

- $(x \mapsto 1, 2) * (x \mapsto 1, 2) \equiv \texttt{false}$
- $(x \mapsto 1, 2) \wedge (x \mapsto 1, 2) \equiv (x \mapsto 1, 2)$
- $(x \mapsto 1, 2) \wedge \neg(x \mapsto 1, 2) \equiv \texttt{false}$
- $(x \mapsto 1, 2) * \neg(x \mapsto 1, 2) \equiv (x \mapsto 1, 2) * \texttt{true}$

Tree formula

"x points to a tree of integers"

$$\mathtt{tree}(x) \triangleq \mu X_v. \begin{array}{l} (x = \mathtt{nil}) \vee \exists x_v, x_l, x_r, x'.\mathtt{isint}(x_v) \wedge \\ (x \mapsto x_v, x' \ * \ x' \mapsto x_l, x_r \ * \ X_v[x_l/x] \ * \ X_v[x_r/x]) \end{array}$$

$[\ /\ ]$ is not $\{\ /\ \}$

$\{\ /\ \}$: capture avoiding substitution
$[\ /\ ]$: postponed substitution connective

$\{\texttt{true}\}x := y\{\texttt{true}\}$ is false since the command will be stuck from a state that has no value on its stack for y

but $\{is(y)\}x := y\{\texttt{true}\}$ is true

so $\{P\{y/x\}\}x := y\{P\}$ is unsound

but $\{P[y/x]\}x := y\{P\}$ is sound

With $is(E) \triangleq (E = E)$, since $[\![E = E]\!]_\rho = \{s, h \mid [\![E]\!]^s$ has a value$\}$

Detailed semantics of $\nu X_v.y = 3 \wedge \exists y.(X_v \wedge y = 5)$

$$
\begin{array}{ll}
 & [\![\nu X_v.y = 3 \wedge \exists y.(X_v \wedge y = 5)]\!]_\emptyset \\
= & \mathrm{gfp}_\emptyset^\subseteq \lambda Y.\ [\![y = 3 \wedge \exists y.(X_v \wedge y = 5)]\!]_{[X_v \to Y]} \\
= & \mathrm{gfp}_\emptyset^\subseteq \lambda Y.\ [\![y = 3]\!]_{[X_v \to Y]} \cap [\![\exists y.(X_v \wedge y = 5)]\!]_{[X_v \to Y]} \\
= & \mathrm{gfp}_\emptyset^\subseteq \lambda Y.\ \{s, h \mid s(y) = 3\} \cap \{s, h \mid \exists v.[s \mid y \to v], h \in [\![X_v \wedge y = 5]\!]_{[X_v \to Y]}\} \\
= & \mathrm{gfp}_\emptyset^\subseteq \lambda Y.\ \{s, h \mid s(y) = 3\} \cap \{s, h \mid \exists v.[s \mid y \to v], h \in Y \wedge [s \mid y \to v](y) = 5\} \\
= & \mathrm{gfp}_\emptyset^\subseteq \lambda Y.\ \{s, h \mid s(y) = 3\} \cap \{s, h \mid [s \mid y \to 5], h \in Y\} \\
= & \emptyset
\end{array}
$$

Detailed semantics of $\nu X_v.y = 3 \wedge \exists z.(X_v \wedge z = 5)$

$$
\begin{array}{rl}
& [\![\nu X_v.y = 3 \wedge \exists z.(X_v \wedge z = 5)]\!]_{\emptyset} \\
= & \mathrm{gfp}_{\emptyset}^{\subseteq} \lambda Y.\ [\![y = 3 \wedge \exists z.(X_v \wedge z = 5)]\!]_{[X_v \to Y]} \\
= & \mathrm{gfp}_{\emptyset}^{\subseteq} \lambda Y.\ [\![y = 3]\!]_{[X_v \to Y]} \cap [\![\exists z.(X_v \wedge z = 5)]\!]_{[X_v \to Y]} \\
= & \mathrm{gfp}_{\emptyset}^{\subseteq} \lambda Y.\ \{s, h \mid s(y) = 3\} \cap \{s, h \mid \exists v.[s \mid z \to v], h \in [\![X_v \wedge z = 5]\!]_{[X_v \to Y]}\} \\
= & \mathrm{gfp}_{\emptyset}^{\subseteq} \lambda Y.\ \{s, h \mid s(y) = 3\} \cap \{s, h \mid \exists v.[s \mid z \to v], h \in Y \wedge [s \mid z \to v](z) = 5\} \\
= & \mathrm{gfp}_{\emptyset}^{\subseteq} \lambda Y.\ \{s, h \mid s(y) = 3\} \cap \{s, h \mid [s \mid z \to 5], h \in Y\} \\
= & \{s, h \mid s(y) = 3\}
\end{array}
$$