

## Theorem Problem

SNU 4541.664A Program Analysis

Spring 2006

Note 11

Prof. Kwangkeun Yi

## 요약해석 디자인과 구현의 예

변수가 있는 정수식 프로그램의 요약해석

명령형 언어 프로그램의 요약해석

함수호출과 고차함수 프로그램의 요약해석

$$\underbrace{x := 1}_{1}; \text{ while } (0 < x) \underbrace{x := x + 1}_{4}$$

$\underbrace{\hspace{10em}}_{3}$

$\underbrace{\hspace{15em}}_{2}$

$\underbrace{\hspace{20em}}_{0}$

# $\hat{\mathbb{Z}}$ 의 높이가 무한한 경우

$$\hat{\mathbb{Z}} = \{\perp\} \cup \{[a, b] \mid a \leq b, a, b \in \mathbb{Z} \cup \{\infty, -\infty\}\}$$

일 때, 방정식은:

$$\hat{X}_0 = \lambda \hat{m}. \hat{X}_2(\hat{X}_1 \hat{m})$$

$$\hat{X}_1 = \lambda \hat{m}. \hat{m}\{x \mapsto [1, 1]\}$$

$$\hat{X}_2 = \lambda \hat{m}. \hat{m} \sqcup (\hat{X}_2 (\hat{X}_3 \hat{m}))$$

$$\hat{X}_3 = \lambda \hat{m}. \hat{m}\{x \mapsto (\hat{m} x) \dot{\dagger} [1, 1]\}$$

프로그램 시작점에서의 모든 메모리 상태를 포섭하는 것이

$$\{\} \in \text{Memory} = \text{Var} \stackrel{\text{fn}}{\mapsto} \text{Value}$$

라고하면,

$$\hat{X}_0 \{\}$$

에서 부터 “연쇄반응”을 일으키는 방정식들만 풀면 된다.

따라서 풀어야 할 방정식들:

$$\hat{X}_0 \{ \} = \hat{X}_2(\hat{X}_1 \{ \}) \quad \text{따라서}$$

$$\hat{X}_1 \{ \} = \{x \mapsto [1, 1]\} \quad \text{따라서}$$

$$\hat{X}_2 \{x \mapsto [1, 1]\} = \{x \mapsto [1, 1]\} \sqcup (\hat{X}_2 (\hat{X}_3 \{x \mapsto [1, 1]\})) \quad \text{따라서}$$

$$\hat{X}_3 \{x \mapsto [1, 1]\} = \{x \mapsto [1, 1]\} \hat{+} [1, 1] \quad \text{따라서}$$

$$\hat{X}_2 \{x \mapsto [2, 2]\} = \{x \mapsto [2, 2]\} \sqcup (\hat{X}_2 (\hat{X}_3 \{x \mapsto [2, 2]\})) \quad \text{따라서}$$

$$\hat{X}_3 \{x \mapsto [2, 2]\} = \{x \mapsto [3, 3]\} \quad \text{따라서}$$

$$\hat{X}_2 \{x \mapsto [3, 3]\} = \dots$$

$$\hat{X}_3 \{x \mapsto [3, 3]\} = \dots$$

⋮

풀어야 할 방정식이 무한히 많아짐.

# 방정식의 재구성: 요약, 요약, 요약

명령문  $C_k$ 의 요약의미

$$\hat{X}_k \in \hat{Memory} \rightarrow Memory$$

중에서

- 관심있는 메모리들에 대한 경우만을 추려 낸 방정식들

$$\begin{aligned} \hat{X}_k \hat{m}_1 &= RHS_1 \\ &\vdots \\ \hat{X}_k \hat{m}_n &= RHS_n \end{aligned}$$

은 요약의미를 공간

$$2^{Memory \times Memory}$$

에서 구하는 셈.

- 한번 더 나아가, 위의 방정식을 하나의 방정식

$$\hat{X}_k \left( \bigsqcup_{i=1, \dots, n} \hat{m}_i \right) = \bigsqcup_{i=1, \dots, n} RHS_i.$$

으로 추리는 것은 요약의미를 공간

$$Memory \times Memory$$

에서 구하는 셈. 단,  $Memory$ 가  $\sqcup$ -complete해야.

# 방정식의 재구성

단수를 낮추어서

$$\hat{X}_i^\uparrow \in \hat{Memory} \quad \text{명령문 } i \text{의 출력}$$

$$\hat{X}_i^\downarrow \in \hat{Memory} \quad \text{명령문 } i \text{의 입력}$$

에 대한 방정식으로 재구성 할 수 있다.

방정식의 갯수 =  $O(\text{프로그램 크기})$ .

아래 방정식으로 부터

$$\hat{X}_0 = \lambda \hat{m}. \hat{X}_2(\hat{X}_1 \hat{m})$$

$$\hat{X}_1 = \lambda \hat{m}. \hat{m}\{x \mapsto [1, 1]\}$$

$$\hat{X}_2 = \lambda \hat{m}. \hat{m} \sqcup (\hat{X}_2 (\hat{X}_3 \hat{m}))$$

$$\hat{X}_3 = \lambda \hat{m}. \hat{m}\{x \mapsto (\hat{m} x) \hat{\dagger} [1, 1]\}$$

다음을 유도할 수 있다:

$$\hat{X}_0^\downarrow = \{\}$$

$$\hat{X}_1^\downarrow = \hat{X}_0^\downarrow$$

$$\hat{X}_2^\downarrow = \hat{X}_1^\downarrow \sqcup \hat{X}_3^\downarrow$$

$$\hat{X}_3^\downarrow = \hat{X}_2^\downarrow$$

$$\hat{X}_0^\uparrow = \hat{X}_2^\uparrow$$

$$\hat{X}_1^\uparrow = \hat{X}_1^\downarrow\{x \mapsto [1, 1]\}$$

$$\hat{X}_2^\uparrow = \hat{X}_2^\downarrow \sqcup \hat{X}_2^\uparrow$$

$$\hat{X}_3^\uparrow = \hat{X}_3^\downarrow\{x \mapsto (\hat{X}_3^\downarrow x) \hat{\dagger} [1, 1]\}$$



위의 방정식을 다시쓰면

$$\begin{pmatrix} \hat{X}_0^\uparrow, \dots, \hat{X}_3^\uparrow \\ \hat{X}_0^\downarrow, \dots, \hat{X}_3^\downarrow \end{pmatrix} = \hat{G}_C \begin{pmatrix} \hat{X}_0^\uparrow, \dots, \hat{X}_3^\uparrow \\ \hat{X}_0^\downarrow, \dots, \hat{X}_3^\downarrow \end{pmatrix}$$

- $\hat{G}_C$ 는  $\hat{F}_C$ 를 그대로 반영하고, 그 밖으로는  $\sqcup_{Memory}$  연산뿐.
- 위 방정식의 해는 연속함수  $\hat{G}_C$ 의 최소 고정점.
- 그 해는,  $\mathcal{F}_C$ 로 부터 같은 방법으로 재구성한  $\mathcal{G}_C$ 의 최소 고정점을 포섭;  $\alpha(lfp \mathcal{F}_C) \sqsubseteq lfp \hat{F}_C$  이므로.

허나, 최소 고정점  $\sqcup_i \hat{G}_C^i \hat{\perp}$  계산은 끝나지 않음;  $\hat{Z}$ 의 높이가 무한.

$$\text{Memory} = \text{Var} \xrightarrow{\text{fin}} \hat{Z}$$

$$\hat{m}_1 \sqsubseteq \hat{m}_2 \stackrel{\text{def}}{=} \forall x \in \text{dom}(\hat{m}_1) \cup \text{dom}(\hat{m}_2) : \hat{m}_1 x \sqsubseteq \hat{m}_2 x$$

$$\forall x \notin \text{dom}(\hat{m}) : \hat{m} x \stackrel{\text{def}}{=} \perp_{\hat{Z}}$$

(메모리에서  $x$  값만 표시)

$\hat{G}_C^i \hat{\perp}, \quad i = 1, 2, \dots$								
	1	2	3	4	5	6	7	...
$\hat{X}_2^\downarrow$		[1,1]	[1,1]	[1,1]	[1,2]	[1,2]	[1,3]	...
$\hat{X}_3^\downarrow$			[1,1]	[1,1]	[1,1]	[1,2]	[1,2]	...
$\hat{X}_0^\uparrow$				[1,1]	[1,1]	[1,1]	[1,2]	...
$\hat{X}_1^\uparrow$	[1,1]	[1,1]	[1,1]	[1,1]	[1,1]	[1,1]	[1,1]	...
$\hat{X}_2^\uparrow$			[1,1]	[1,1]	[1,1]	[1,2]	[1,3]	...
$\hat{X}_3^\uparrow$				[2,2]	[2,2]	[2,3]	[2,3]	...

계속 증가하는  $\hat{G}_C^i \hat{\perp}$

$$\begin{pmatrix} \hat{X}_0^\uparrow, \dots, \hat{X}_3^\uparrow \\ \hat{X}_0^\downarrow, \dots, \hat{X}_3^\downarrow \end{pmatrix} = \hat{\mathcal{G}}_C \begin{pmatrix} \hat{X}_0^\uparrow, \dots, \hat{X}_3^\uparrow \\ \hat{X}_0^\downarrow, \dots, \hat{X}_3^\downarrow \end{pmatrix}$$

위의 해를 포섭하는 것을 유한번만에 계산하려면 축지법(widening)( $\nabla$ )을 쓴  $\{\hat{Y}_i\}_i$ 을 계산해야:

$$\hat{Y}_0 = \hat{\perp}$$

$$\hat{Y}_{i+1} = \begin{cases} \hat{Y}_i & \text{if } \hat{\mathcal{G}}_C(\hat{Y}_i) \sqsubseteq \hat{Y}_i \\ \hat{Y}_i \nabla \hat{\mathcal{G}}_C(\hat{Y}_i) & \text{otherwise} \end{cases}$$

- 유한한 체인  $\{\hat{Y}_i\}_i$ 의 끝  $\lim_i \hat{Y}_i$ 은  $\sqcup_i \hat{\mathcal{G}}_C^i \hat{\perp}$ 을 포섭 (Theorem [widen's safety])
- $\lim_i \hat{Y}_i$ 를 좁히기(narrowing)( $\Delta$ )로 다듬고: 좁힌 결과도  $\sqcup_i \hat{\mathcal{G}}_C^i \hat{\perp}$ 를 포섭 (Theorem [narrow's safety])

## ▽ 과 △의 예

$$\perp \nabla X = X$$

$$X \nabla \perp = X$$

$$[l, u] \nabla [l', u'] = [(l' < l? -\infty : l), (u' > l? \infty : u)]$$

$$\perp \triangle X = \perp$$

$$X \triangle \perp = \perp$$

$$[l, u] \triangle [l', u'] = [(l = -\infty ? l' : l), (u = \infty ? u' : u)]$$

읽기: “Comparing the Galois Connection and Widening/Narrowing Approaches to Abstract Interpretation”

축지법으로 뛰는 체인  $\{\hat{Y}_i\}_i$ :

$$\hat{Y}_0 = \hat{1}$$

$$\hat{Y}_{i+1} = \begin{cases} \hat{Y}_i & \text{if } \hat{\mathcal{G}}_C(\hat{Y}_i) \sqsubseteq \hat{Y}_i \\ \hat{Y}_i \nabla \hat{\mathcal{G}}_C(\hat{Y}_i) & \text{otherwise} \end{cases}$$

(메모리에서  $x$  값만 표시)

$\hat{Y}_i, \quad i = 3, 4, \dots$					
	3	4	5	6	7
$\hat{X}_2^\downarrow$	[1,1]	[1,1]	$\nabla[1,2]=[1,\infty]$	[1, $\infty$ ]	[1, $\infty$ ]
$\hat{X}_3^\downarrow$	[1,1]	[1,1]	[1,1]	$\nabla[1,\infty]=[1,\infty]$	[1, $\infty$ ]
$\hat{X}_0^\uparrow$		[1,1]	[1,1]	[1,1]	$\nabla[1,\infty]=[1,\infty]$
$\hat{X}_1^\uparrow$	[1,1]	[1,1]	[1,1]	[1,1]	[1,1]
$\hat{X}_2^\uparrow$	[1,1]	[1,1]	[1,1]	$\nabla[1,\infty]=[1,\infty]$	[1, $\infty$ ]
$\hat{X}_3^\uparrow$		[2,2]	[2,2]	$\nabla[2,3]=[2,\infty]$	[2, $\infty$ ]

$\lim_i \hat{Y}_i = \hat{Y}_7$ .

좁히기로 재결음하는 체인  $\{\hat{Z}_i\}_i$ :

$$\begin{aligned}\hat{Z}_0 &= \lim_i \hat{Y}_i \\ \hat{Z}_{i+1} &= \hat{Z}_i \Delta \hat{G}_C(\hat{Z}_i)\end{aligned}$$

(메모리에서  $x$  값만 표시)

$\hat{Z}_i, \quad i = 0, 1, \dots$	0	1
$\hat{X}_2^\downarrow$	[1, $\infty$ ]	[1, $\infty$ ]
$\hat{X}_3^\downarrow$	[1, $\infty$ ]	[1, $\infty$ ]
$\hat{X}_0^\uparrow$	[1, $\infty$ ]	[1, $\infty$ ]
$\hat{X}_1^\uparrow$	[1, 1]	[1, 1]
$\hat{X}_2^\uparrow$	[1, $\infty$ ]	[1, $\infty$ ]
$\hat{X}_3^\uparrow$	[2, $\infty$ ]	[2, $\infty$ ]

$\lim_i \hat{Z}_i = Z_1$  (그리고 더 좋아지는 건 없음):

$\Delta$ 가 효과를 발휘하는 예

$$\hat{C} \text{ while } E C \hat{m} = \hat{m} \sqcup (\hat{C} \text{ while } E C (\hat{C} C (\hat{m} \sqcap \text{True}(E))))$$

이면, 프로그램  $x := 1; \text{ while } (x < 10) \underbrace{x := x + 1}_3$  에 대한  $\hat{X}_3^\downarrow$ 의

2

방정식은  $\hat{X}_3^\downarrow = \hat{X}_2^\downarrow \sqcap [-\infty, 10]$ .

축지법 과정은:

	4	5	6
$\hat{X}_2^\downarrow$	[1,1]	$\nabla [1,2] = [1,\infty]$	[1, $\infty$ ]
$\hat{X}_3^\downarrow$	[1,1]	[1,1]	$\nabla [1,10] = [1,\infty]$

좁히기 과정은:

	0	1
$\hat{X}_2^\downarrow$	[1, $\infty$ ]	
$\hat{X}_3^\downarrow$	[1, $\infty$ ]	$\Delta[-\infty,10] = [1,10]$

## 함수호출과 고차함수 프로그램의 요약해석

$E$	$\rightarrow$	$n$	$(n \in \mathbb{Z})$
		$E + E$	
		$- E$	
		$x$	변수
		$f \lambda x. E$	재귀 함수
		$E E$	함수 호출

의미함수

$$\mathcal{V} : Exp \rightarrow 2^{Env} \rightarrow 2^{Val}$$

의미공간

$$\begin{aligned} \sigma \in Env &= Var \xrightarrow{\text{fin}} Val \\ Val &= \mathbb{Z} + Closure \\ Closure &= Exp \times Env \end{aligned}$$



## 환경에서 변수관계 무시

$$2^{Env} \xrightleftharpoons[\alpha]{\gamma} Var \xrightarrow{fin} 2^{Val}$$

즉, 의미공간

$$\begin{aligned} \sigma \in Env &= Var \xrightarrow{fin} Val \\ Val &= 2^{\mathbb{Z}} + 2^{Closure} \\ Closure &= Exp \times Env \end{aligned}$$

의미함수

$$\mathcal{V} : Exp \rightarrow Env \rightarrow Val$$

$$\mathcal{V} n \sigma = \{n\}$$

$$\mathcal{V} x \sigma = \sigma x$$

$$\mathcal{V} E_1 + E_2 \sigma = (\mathcal{V} E_1 \sigma) \dot{+} (\mathcal{V} E_2 \sigma)$$

$$\mathcal{V} - E \sigma = \dot{-} (\mathcal{V} E \sigma)$$

$$\mathcal{V} f \lambda x. E \sigma = \{ \langle f \lambda x. E, \sigma \rangle \}$$

$$\begin{aligned} \mathcal{V} E_1 E_2 \sigma &= \cup \{ \mathcal{V} E \sigma' \{ x \mapsto \mathcal{V} E_2 \sigma \} \{ f \mapsto \{ \langle f \lambda x. E, \sigma' \rangle \} \} \\ &\quad | \langle f \lambda x. E, \sigma' \rangle \in \mathcal{V} E_1 \sigma \} \end{aligned}$$

## 주의

프로그램  $E$ 의 의미  $\mathcal{V} E$ 를 정의하는가?

$E$ 가 재귀함수를 호출하고 그 함수내에 재귀호출 " $f E$ "이 있는 경우:

$$\begin{aligned} \mathcal{V} E &= \dots \mathcal{V} f E' \dots \\ &= \dots (\dots \mathcal{V} f E' \dots) \dots \\ &= \dots \end{aligned}$$

$\mathcal{V}$ 은  $E$ 를 가지고 만드는 방정식을 표현하는 것 뿐.

그 방정식의 해가 프로그램의  $E$ 의 의미  $\mathcal{V} E$ .

식  $E$ 의 방정식을 만들어주는 함수

$$\mathcal{F} : (Exp \rightarrow Env \rightarrow Val) \rightarrow (Exp \rightarrow Env \rightarrow Val)$$

는 다음아니라:

$$\mathcal{F} \mathcal{V} n \sigma = \{n\}$$

$$\mathcal{F} \mathcal{V} x \sigma = \sigma x$$

$$\mathcal{F} \mathcal{V} E_1 + E_2 \sigma = \{z_1 + z_2 \mid z_i \in \mathcal{V} E_i \sigma\}$$

$$\mathcal{F} \mathcal{V} - E \sigma = \{-z \mid z \in \mathcal{V} E \sigma\}$$

$$\mathcal{F} \mathcal{V} f \lambda x. E \sigma = \{\langle f \lambda x. E, \sigma \rangle\}$$

$$\begin{aligned} \mathcal{F} \mathcal{V} E_1 E_2 \sigma &= \cup \{ \mathcal{V} E \sigma' \{x \mapsto \mathcal{V} E_2 \sigma\} \{f \mapsto \langle f \lambda x. E, \sigma' \rangle\} \\ &\quad \mid \langle f \lambda x. E, \sigma' \rangle \in \mathcal{V} E_1 \sigma \} \end{aligned}$$

프로그램  $E$ 의 의미는 그 의미  $\mathcal{V} E$ 에 대한 방정식

$$\mathcal{V} E = \mathcal{F} \mathcal{V} E$$

과  $E$ 안의 모든 식  $E_i$ 들의 의미  $\mathcal{V} E_i$ 에 대한 방정식

$$\mathcal{V} E_i = \mathcal{F} \mathcal{V} E_i$$

들의 해를 가지고 정의됨.

그 연립방정식을 하나의 함수  $\mathcal{F}_E$ 를 이용해서 표현하면

$$\begin{pmatrix} X_0 \\ \vdots \\ X_n \end{pmatrix} = \mathcal{F}_E \begin{pmatrix} X_0 \\ \vdots \\ X_n \end{pmatrix}$$

이 되고, 방정식의 주인공  $X_i$ 는  $\mathcal{V} E_i$ 대신에 쓴 것이고, 프로그램  $E$ 의 모든 하부 식들은  $n$ 개. ( $\mathcal{F}_E$ 의 정의는  $\mathcal{F}$ 로 부터 명백)

다음 프로그램

$$1 + (k \lambda x. k (- x)) 0$$

을 생각하자. 위 프로그램의 모든 부품식들마다 번호를 붙이자.

$$\underbrace{1}_{1} + \left( \underbrace{(k \lambda x.}_{6} \underbrace{k}_{7} \underbrace{(-}_{8} \underbrace{x}_{4}))}_{5} \right) \underbrace{0}_{4}$$

$\underbrace{\hspace{10em}}_{3}$   
 $\underbrace{\hspace{15em}}_{2}$   
 $\underbrace{\hspace{20em}}_{0}$

각 프로그램 식  $E_i$ 에 대해서 방정식

$$\mathcal{V} E_i = \mathcal{F} \mathcal{V} E_i$$

을 모은 연립방정식은

$$\mathcal{V} E_0 = \lambda\sigma. \{z_1 + z_2 \mid z_1 \in \mathcal{V} E_1 \sigma, z_2 \in \mathcal{V} E_2 \sigma\}$$

$$\mathcal{V} E_1 = \lambda\sigma. \{1\}$$

$$\mathcal{V} E_2 = \lambda\sigma. \cup \{ \mathcal{V} E \sigma' \{x \mapsto \mathcal{V} E_4 \sigma\} \{f \mapsto \langle f \lambda x. E, \sigma' \rangle\} \\ \mid \langle f \lambda x. E, \sigma' \rangle \in \mathcal{V} E_3 \sigma \}$$

$$\mathcal{V} E_3 = \lambda\sigma. \{ \langle k \lambda x. E_5, \sigma \rangle \}$$

$$\mathcal{V} E_4 = \lambda\sigma. \{0\}$$

$$\mathcal{V} E_5 = \lambda\sigma. \cup \{ \mathcal{V} E \sigma' \{x \mapsto \mathcal{V} E_7 \sigma\} \{f \mapsto \langle f \lambda x. E, \sigma' \rangle\} \\ \mid \langle f \lambda x. E, \sigma' \rangle \in \mathcal{V} E_6 \sigma \}$$

$$\mathcal{V} E_6 = \lambda\sigma. \sigma k$$

$$\mathcal{V} E_7 = \lambda\sigma. \{-z \mid z \in \mathcal{V} E_8 \sigma\}$$

$$\mathcal{V} E_8 = \lambda\sigma. \sigma x$$

방정식의 주인공을  $\nu E_i$  대신에  $X_i$ 로 다시 쓰면:

$$X_0 = \lambda\sigma.\{z_1 + z_2 \mid z_1 \in X_1 \sigma, z_2 \in X_2 \sigma\}$$

$$X_1 = \lambda\sigma.\{1\}$$

$$X_2 = \lambda\sigma.\cup \{X_E \sigma' \{x \mapsto X_4 \sigma\} \{f \mapsto \langle f \lambda x.E, \sigma' \rangle\} \\ \mid \langle f \lambda x.E, \sigma' \rangle \in X_3 \sigma\}$$

$$X_3 = \lambda\sigma.\{\langle k \lambda x.X_5, \sigma \rangle\}$$

$$X_4 = \lambda\sigma.\{0\}$$

$$X_5 = \lambda\sigma.\cup \{X_E \sigma' \{x \mapsto X_7 \sigma\} \{f \mapsto \langle f \lambda x.E, \sigma' \rangle\} \\ \mid \langle f \lambda x.E, \sigma' \rangle \in X_6 \sigma\}$$

$$X_6 = \lambda\sigma.\sigma k$$

$$X_7 = \lambda\sigma.\{-z \mid z \in X_8 \sigma\}$$

$$X_8 = \lambda\sigma.\sigma x$$

- 특이점: 방정식의 주인공들이 완전히 드러나지 않음.  
 $X_2$ 의 방정식을 보면, " $X_E$ "에서  $E$ 의 정체는  $X_3$ 의 해가 밝혀질 때 알 수 있음.

$$\langle f \lambda x. E, \sigma' \rangle \in X_3 \sigma$$

- 함수가 자유롭게 흘러다닐 수 있는(*higher-order*) 언어인 경우 생기는 현상.  
혹은, "if the language has the control a first-class object."



$lfp\mathcal{F}_E$  는  $lfp\mathcal{F}$ 의 일부분

방정식의 해

$$lfp\hat{\mathcal{F}}_E$$

는

$$lfp\hat{\mathcal{F}} \in Exp \rightarrow Env \rightarrow Val$$

중에서 프로그램  $E$ 를 구성하는 식들의 의미들로 구성된다:

$$lfp\hat{\mathcal{F}}_E = \langle (lfp\hat{\mathcal{F}}) E_0, (lfp\hat{\mathcal{F}}) E_1, \dots, (lfp\hat{\mathcal{F}}) E_n \rangle$$

## 요약공간

$$\begin{aligned}\hat{\sigma} \in \hat{Env} &= \text{Var} \xrightarrow{\text{fin}} \hat{Val} \\ \hat{Val} &= \hat{\mathbb{Z}} + \hat{Closure} \\ \hat{Closure} &= 2^{Exp}\end{aligned}$$

프로그램  $E$  의 요약 의미는 연속함수  $\hat{\mathcal{F}}$  가  $E$  로 부터 만들어 내는 연립방정식

$$\begin{pmatrix} X_0 \\ \vdots \\ X_n \end{pmatrix} = \hat{\mathcal{F}}_E \begin{pmatrix} X_0 \\ \vdots \\ X_n \end{pmatrix}$$

의 해.

$$\hat{\mathcal{F}} \in (Exp \rightarrow \hat{Env} \rightarrow \hat{Val}) \rightarrow (Exp \rightarrow \hat{Env} \rightarrow \hat{Val})$$

$$\hat{\mathcal{F}} \hat{\mathcal{V}} n \hat{\sigma} = \alpha_2 \{n\}$$

$$\hat{\mathcal{F}} \hat{\mathcal{V}} x \hat{\sigma} = \hat{\sigma} x$$

$$\hat{\mathcal{F}} \hat{\mathcal{V}} E_1 + E_2 \hat{\sigma} = (\hat{\mathcal{V}} E_1 \hat{\sigma}) \hat{+} (\hat{\mathcal{V}} E_2 \hat{\sigma})$$

$$\hat{\mathcal{F}} \hat{\mathcal{V}} - E \hat{\sigma} = \ominus(\hat{\mathcal{V}} E \hat{\sigma})$$

$$\hat{\mathcal{F}} \hat{\mathcal{V}} \lambda x.E \hat{\sigma} = \{\lambda x.E\}$$

$$\hat{\mathcal{F}} \hat{\mathcal{V}} E_1 E_2 \hat{\sigma} = \sqcup \{ \hat{\mathcal{V}} E \hat{\sigma} \{ x \mapsto \hat{\sigma} x \cup \hat{\mathcal{V}} E_2 \hat{\sigma} \} \mid \lambda x.E \in \hat{\mathcal{V}} E_1 \hat{\sigma} \}$$

- $\hat{\mathcal{F}}$ 는 연속 함수이므로,  $\hat{\mathcal{F}}_E$ 도 연속함수. 따라서,  $lfp\hat{\mathcal{F}}$ 와  $lfp\hat{\mathcal{F}}_E$ 가 존재.
- 연립 방정식의 해  $lfp\hat{\mathcal{F}}_E$ 는  $lfp\hat{\mathcal{F}}$ 의 일부분:

$$lfp\hat{\mathcal{F}}_E = \langle (lfp\hat{\mathcal{F}}) E_0, \dots, (lfp\hat{\mathcal{F}}) E_n \rangle$$

## 올바름

$$\alpha(\text{lfp}\mathcal{F}) \sqsubseteq \text{lfp}\hat{\mathcal{F}}?$$

요약해석 틀에 의해, 증명할 것은

$$\alpha \circ \mathcal{F} \sqsubseteq \hat{\mathcal{F}} \circ \alpha \quad (\text{혹은, } \alpha(f) \sqsubseteq g \implies \alpha(\mathcal{F} f) \sqsubseteq \hat{\mathcal{F}} g)$$

즉, 모든 프로그램 식  $E$ 에 대해서

$$\forall f : (\alpha(\mathcal{F} f)) E \sqsubseteq (\hat{\mathcal{F}}(\alpha f)) E.$$

## 구현

분석의 구현은, 분석할 프로그램  $E$ 가 주어졌을 때 연립방정식

$$\begin{pmatrix} \hat{X}_0 \\ \vdots \\ \hat{X}_n \end{pmatrix} = \hat{\mathcal{F}}_E \begin{pmatrix} \hat{X}_0 \\ \vdots \\ \hat{X}_n \end{pmatrix}$$

을 푸는것, 즉  $\text{lfp} \hat{\mathcal{F}}_E$ 를 계산하는 것.

두가지 방법으로 계산할 수 있다.

- $\hat{Env}$ 의 원소수가 유한한 경우.

$\hat{Env} \rightarrow \hat{Val}$ 의 원소는 유한한 사이즈의 테이블. 방정식의 해

$$\bigsqcup_{i \in \mathbb{N}} \hat{\mathcal{F}}_E^i(\perp, \dots, \perp)$$

를 직접 계산.

- $\hat{Env}$ 의 원소가 무한한 경우. 혹은, 모든 경우를 계산할 필요가 없을 경우.

하나의 요약환경  $\hat{\sigma}_0$ 이 모든 초기환경을 포섭하도록하면, 관심있는 것은

$$((lfp \hat{\mathcal{F}}_E) \downarrow E) \hat{\sigma}_0.$$

위를 계산하는 데 필요한 만큼만 연립 방정식에서 계산.

“필요한 만큼만 연립 방정식에서 계산”?

- 연립방정식

$$\begin{pmatrix} \hat{X}_0 \\ \vdots \\ \hat{X}_n \end{pmatrix} = \hat{\mathcal{F}}_E \begin{pmatrix} \hat{X}_0 \\ \vdots \\ \hat{X}_n \end{pmatrix}$$

을 확장;  $\hat{X}_i$ 를  $Env$ 의 원소 수 만큼 확장:

$$\begin{pmatrix} \hat{X}_0 \sigma_0 \\ \vdots \\ \hat{X}_0 \sigma_\omega \\ \vdots \\ \hat{X}_n \sigma_0 \\ \vdots \\ \hat{X}_n \sigma_\omega \end{pmatrix} = \hat{\mathcal{F}}_E^{Env} \begin{pmatrix} \hat{X}_0 \sigma_0 \\ \vdots \\ \hat{X}_0 \sigma_\omega \\ \vdots \\ \hat{X}_n \sigma_0 \\ \vdots \\ \hat{X}_n \sigma_\omega \end{pmatrix}$$

위의 방정식들 중에서 프로그램  $E$ 와 초기 요약환경  $\hat{\sigma}_0$ 에  
해당하는 주인공  $\hat{X}_E \hat{\sigma}_0$ 의 방정식을 푸는 데 필요한 방정식  
들만을 골라서 푼다



필요한 방정식들이 무한히 많아 지는 경우는? 필요한 방정식이  $\hat{X}_i$ 에 대해 복수로 나오게 되는 경우, 즉,  $\hat{X}_i \sigma_1$ 에 대한 방정식과  $\hat{X}_i \sigma_2$ 에 대한 방정식이 필요하게 되면?

- 대신에  $\hat{X}_i (\sigma_1 \sqcup \sigma_2)$ 에 대한 방정식만을 생각한다면, 계산에 넣어야 할 방정식은  $E\hat{n}v$ 의 높이가 유한하다면 항상 유한개의 방정식만 다루게 된다.
- 그렇게 해서 계산된  $\hat{X}_i (\sigma_1 \sqcup \sigma_2)$ 의 해는  $\hat{X}_i \sigma_1$ 의 해와  $\hat{X}_i \sigma_2$ 의 해를 모두 포섭.