# SNU 4541.664A Program Analysis

# Spring 2005

# Note 10

Prof. Kwangkeun Yi

## 요약해석 디자인과 구현의 예

- 변수가 있는 정수식 프로그램의 요약해석
- 명령형 언어 프로그램의 요약해석

# 변수가 있는 정수식 프로그램의 요약해석

$$
\begin{array}{rcll}
E & \to & n & (n \in \mathbb{Z}) \\
 & | & E \ \texttt{+} \ E & \\
 & | & \texttt{-} \ E & \\
 & | & x & \text{변수} \\
 & | & \texttt{let} \ x \ \texttt{=} \ E_1 \ \texttt{in} \ E_2 & \text{지역 변수}
\end{array}
$$

의미공간

$$\sigma \in Env = Var \xrightarrow{\text{fin}} 2^{\mathbb{Z}}$$

의미함수

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

$$\begin{array}{rcl}
\mathcal{V}\, n\, \sigma & = & \{n\} \\
\mathcal{V}\, x\, \sigma & = & \sigma\, x \\
\mathcal{V}\, E_1 \texttt{+} E_2\, \sigma & = & \{z_1 + z_2 \mid z_i \in \mathcal{V}\, E_i\, \sigma\} \\
\mathcal{V}\, \texttt{-} E\, \sigma & = & \{-z \mid z \in \mathcal{V}\, E\, \sigma\} \\
\mathcal{V}\, \texttt{let}\ x \texttt{=} E_1\ \texttt{in}\ E_2\, \sigma & = & \mathcal{V}\, E_2\, \sigma\{x \mapsto \mathcal{V}\, E_1\, \sigma\}
\end{array}$$

모든 식 $E$의 의미는 조립식으로 $\mathcal{V}\, E$, 즉 상수함수 $\lambda x.\mathcal{V}\, E$의 최소고정점.

요약된 의미함수

$$\hat{\mathcal{V}} : Exp \to \hat{Env} \to \hat{\mathbb{Z}}$$

의 의미공간 사이의 갈로아 연결

$$Env \to 2^{Z} \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \hat{Env} \to \hat{\mathbb{Z}}$$

은 조립식으로, $Env \underset{\alpha_1}{\overset{\gamma_1}{\leftrightarrows}} \hat{Env}$와 $2^{\mathbb{Z}} \underset{\alpha_2}{\overset{\gamma_2}{\leftrightarrows}} \hat{\mathbb{Z}}$을 가지고:

$$\begin{aligned} \alpha\ f &= \alpha_2 \circ f \circ \gamma_1 \\ \gamma\ \hat{f} &= \gamma_2 \circ \hat{f} \circ \alpha_1 \end{aligned}$$

$$
\begin{array}{rcl}
\hat{\mathcal{V}}\, n\, \hat{\sigma} & = & \alpha_2\, \{n\} \\
\hat{\mathcal{V}}\, E_1 \texttt{ + } E_2\, \hat{\sigma} & = & (\hat{\mathcal{V}}\, E_1\, \hat{\sigma}) \hat{+} (\hat{\mathcal{V}}\, E_2\, \hat{\sigma}) \\
\hat{\mathcal{V}}\, \texttt{-}\, E\, \hat{\sigma} & = & \hat{-}(\hat{\mathcal{V}}\, E\, \hat{\sigma}) \\
\hat{\mathcal{V}}\, \texttt{let}\ x \texttt{ = } E_1 \texttt{ in } E_2\, \hat{\sigma} & = & \hat{\mathcal{V}}\, E_2\, \hat{\sigma}\{x \mapsto \hat{\mathcal{V}}\, E_1\, \hat{\sigma}\}
\end{array}
$$

정수식 $E$의 요약 의미는 조립식: $\hat{\mathcal{V}}\, E$. 확인할 것: 모든 정수식 $E$에 대해서

$$\alpha(\mathcal{V}\, E) \sqsubseteq \hat{\mathcal{V}}\, E$$

# 구현

프로그램 $E$에 대해서, $\hat{\mathcal{V}}\ E \in \hat{Env} \to \hat{\mathbb{Z}}$의 계산:

- $\hat{Env}$의 원소가 유한하다면?
  모든 $\hat{\sigma} \in \hat{Env}$에 대해서 $\hat{\mathcal{V}}\ E\ \hat{\sigma}$를 각각 계산.
  $\hat{\mathcal{V}}\ E\ \hat{\sigma}$의 계산은 항상 끝남 ($E$에 대한 조립).
- $\hat{Env}$의 원소가 무한히 많다면? 혹은, 유한하지만 모든 경우가 필요 없다면?
  - 하나의 요약 환경 $\hat{\sigma}_0$으로 초기 환경을 모두 포섭시키고, $\hat{\mathcal{V}}\ E\hat{\sigma}_0$을 계산.
  - 반드시 하나의 요약환경일 필요 없슴.

# 구현 예1

자유변수가 없는 프로그램

$$E = (\texttt{let}\ x = 1\ \texttt{in}\ (\texttt{let}\ y = 2\ \texttt{in}\ (\texttt{let}\ x = 3\ \texttt{in}\ x + y)) + x)$$

$$\begin{aligned} \hat{\mathcal{V}}\, E \hat{\sigma} &= \hat{\mathcal{V}}\, E_2\, \hat{\sigma}\{x \mapsto \hat{\mathcal{V}}\, 1\, \hat{\sigma}\} \\ &= \hat{\mathcal{V}}\, E_2\, \hat{\sigma}\{x \mapsto 1\} \\ &= \cdots \end{aligned}$$

- $\hat{\mathcal{V}}\, E$는 상수함수, $\hat{\sigma}$와 상관없는.
- $\hat{\mathcal{V}}\, E$는 $E$의 구조를 타고 조립. 유한시간에 계산가능.

# 구현 예2

자유변수(입력)가 있는 프로그램

$$E = (\texttt{let } y = 2 \texttt{ in } (\texttt{let } x = 3 \texttt{ in } x + y)) + x$$

$$\begin{aligned} \hat{\mathcal{V}}\, E\hat{\sigma} &= (\hat{\mathcal{V}}\, E_1\, \hat{\sigma})\hat{+}(\hat{\mathcal{V}}\, x\, \hat{\sigma}) \\ &= \cdots \end{aligned}$$

함수 $\hat{\mathcal{V}}\, E$는 $E$의 구조를 타고 조립식으로 정의되지만, 어떤 함수일까?

- $\hat{\mathcal{V}}\, E\, \{x \mapsto \top\}$만 계산
- 혹은 각 경우를 모두 계산:

$$\{\hat{\mathcal{V}}\, E\, \{x \mapsto \bot\}, \hat{\mathcal{V}}\, E\, \{x \mapsto 0\}, \hat{\mathcal{V}}\, E\, \{x \mapsto +\}, \cdots\}$$

# 명령형 언어 프로그램의 요약해석

$$
\begin{array}{rcll}
C & \to & \mathtt{skip} & \\
 & \mid & x \mathrel{\mathtt{:=}} E & \\
 & \mid & \mathtt{if}\ E\ C\ C & \\
 & \mid & C\ ;\ C & \\
 & \mid & \mathtt{while}\ E\ \mathtt{do}\ C & \\
E & \to & n & (n \in \mathbb{Z}) \\
 & \mid & x & \\
 & \mid & E \mathbin{\mathtt{+}} E \mid E \mathbin{\mathtt{<}} E &
\end{array}
$$

의미공간은

$$
\begin{array}{rcl}
\mathit{Memory} & = & \mathit{Loc} \xrightarrow{\text{fin}} \mathit{Value} \\
\mathit{Value} & = & 2^{\mathbb{Z}} + 2^{\mathbb{B}}
\end{array}
$$

# 세 갈래 길

- 프로그램 $C$의 의미를 조립식으로 (최소고정점)
- 프로그램 $C$의 의미를 방정식의 해로 (최소고정점)
- 프로그램 $C$의 의미를 기계상태의 전이과정으로 (최소고정점)

# $C$의 의미를 방정식의 해로

일단, 프로그램 $C$의 의미를 다음의 함수로 정의해 볼까:

$$\mathcal{C} \in Cmd \to Memory \to Memory$$
$$\mathcal{V} \in Expr \to Memory \to Value$$

$$\begin{array}{rcl}
\mathcal{C}\ \texttt{skip}\ m & = & m \\
\mathcal{C}\ x\ \texttt{:=}\ E\ m & = & m\{x \mapsto \mathcal{V}\ E\ m\} \\
\mathcal{C}\ \texttt{if}\ E\ C_1\ C_2\ m & = & \mathcal{V}\ E\ m\ ?\ \mathcal{C}\ C_1\ m\ :\ \mathcal{C}\ C_2\ m \\
\mathcal{C}\ C_1\ ;\ C_2\ m & = & \mathcal{C}\ C_2\ (\mathcal{C}\ C_1\ m) \\
\mathcal{C}\ \texttt{while}\ E\ \texttt{do}\ C\ m & = & \mathcal{V}\ E\ m\ ?\ \mathcal{C}\ \texttt{while}\ E\ \texttt{do}\ C\ (\mathcal{C}\ C\ m)\ :\ m \\
\mathcal{V}\ n\ m & = & \{n\} \\
\mathcal{V}\ x\ m & = & m\ x
\end{array}$$

# 문제

프로그램 $C$의 의미는 $\mathcal{C}\,C$로 정의되는가? No.

while-문의 경우:

$$
\begin{array}{rcl}
\mathcal{C}\ \texttt{while}\ E\ \texttt{do}\ C & = & \cdots \mathcal{C}\ \texttt{while}\ E\ \texttt{do}\ C \cdots \\
& = & \cdots (\cdots \mathcal{C}\ \texttt{while}\ E\ \texttt{do}\ C \cdots) \cdots \\
& = & \cdots
\end{array}
$$

답) $\mathcal{C}\,C$는 명령문 프로그램 $C$를 가지고 만드는 방정식을 표현하는 것 뿐. 그 방정식의 해가 프로그램 $C$의 의미.

프로그램 $C$의 의미 방정식

$$\mathcal{C}\, C = \cdots$$

의 오른편은 위에서 표현한 함수 $\mathcal{C}$의 내용을 고스란히 가지는 상위의 함수

$$\mathcal{F} : (Cmd \to Memory \to Memory) \to (Cmd \to Memory \to Memory)$$

가 정의하는 $\mathcal{F}\,\mathcal{C}\,C$가 된다:

$$\begin{array}{rcl}
\mathcal{F}\,\mathcal{C}\,\texttt{skip}\,m & = & m \\
\mathcal{F}\,\mathcal{C}\,x \texttt{ := } E\,m & = & m\{x \mapsto \mathcal{V}\,E\,m\} \\
\mathcal{F}\,\mathcal{C}\,\texttt{if}\ E\,C_1\,C_2\,m & = & \mathcal{V}\,E\,m\,?\,\mathcal{C}\,C_1\,m\ :\ \mathcal{C}\,C_2\,m \\
\mathcal{F}\,\mathcal{C}\,C_1\,;\,C_2\,m & = & \mathcal{C}\,C_2\,(\mathcal{C}\,C_1\,m) \\
\mathcal{F}\,\mathcal{C}\,\texttt{while}\ E\ \texttt{do}\ C\,m & = & \mathcal{V}\,E\,m\,?\,\mathcal{C}\,\texttt{while}\ E\ \texttt{do}\ C\,(\mathcal{C}\,C\,m)\ :\ m
\end{array}$$

# $C$의 의미 방정식

프로그램 $C$의 의미는 그 의미 $\mathcal{C}\,C$에 대한 방정식

$$\mathcal{C}\,C = \mathcal{F}\,\mathcal{C}\,C$$

와 $C$안의 모든 명령문 $C_i$들에 대한 방정식

$$\mathcal{C}\,C_i = \mathcal{F}\,\mathcal{C}\,C_i$$

들의 해를 가지고 정의된다.

그러한 $\mathcal{F}$를 통해서 프로그램 $C$로 부터 도출되는 연립방정식을 하나의 함수 $\mathcal{F}_C$를 가지고 표현하면

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

이 되고, 방정식의 주인공 $X_i$는 $\mathcal{C}\ C_i$를 대신에 쓴 것. ($C$안의 명령문들의 갯수는 $n$)

- $\mathcal{F}_C$의 정의는 $\mathcal{F}$와 $\mathcal{V}$ 로 부터 명백
- 방정식의 해는 $\mathcal{F}_C$의 최소고정점
- $\mathcal{F}_C$의 최소고정점은 존재: (why?)

$$lfp\mathcal{F}_C = \bigsqcup_i (\mathcal{F}_C^i \langle \bot, \cdots, \bot \rangle)$$

# $\mathit{lfp}\mathcal{F}_C$는 $\mathit{lfp}\mathcal{F}$의 일부

방정식의 해

$$\mathit{lfp}\mathcal{F}_C$$

는

$$\mathit{lfp}\mathcal{F} \in \mathit{Cmd} \rightarrow \mathit{Memory} \rightarrow \mathit{Memory}$$

중에서 프로그램 $C$를 구성하는 명령문들의 의미들로 구성된다:

$$\mathit{lfp}\mathcal{F}_C = \langle (\mathit{lfp}\mathcal{F})C_0, (\mathit{lfp}\mathcal{F})C_1, \cdots, (\mathit{lfp}\mathcal{F})C_n \rangle$$

- $\mathcal{F}_C$를 구성하는 속 내용은 모두 $\mathcal{F}$의 정의 그대로.
- $\mathit{lfp}\mathcal{F} \in \mathit{Cmd} \rightarrow \mathit{Memory} \rightarrow \mathit{Memory}$는 모든 명령문에 대한 의미.

# 앞으로

- 연속인(*continuous*) 요약 의미 함수를 정의:

$$\hat{\mathcal{F}} \in (Cmd \to \hat{Memory} \to \hat{Memory}) \to (Cmd \to \hat{Memory} \to \hat{Memory})$$

- 프로그램 $C$의 요약 의미는 $\hat{\mathcal{F}}$로 부터 도출되는 연립방정식

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

  의 해, 즉 $\mathit{lfp}\hat{\mathcal{F}}_C$.

- 방정식의 해 $\mathit{lfp}\hat{\mathcal{F}}_C$는 $\mathit{lfp}\mathcal{F}$의 일부:

$$\mathit{lfp}\hat{\mathcal{F}}_C = \langle (\mathit{lfp}\hat{\mathcal{F}})\, C_0, (\mathit{lfp}\hat{\mathcal{F}})\, C_1, \cdots, (\mathit{lfp}\hat{\mathcal{F}})\, C_n \rangle$$

- 올바른가 검증:

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

  즉, 요약해석의 틀에 의해서 다음을 증명하면 됨:

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

# 요약된 의미 방정식

프로그램 $C$의 요약된 의미는 다음의 요약공간

$$\begin{array}{rcl} \widehat{Memory} & = & Loc \xrightarrow{\text{fin}} \widehat{Value} \\ \widehat{Value} & = & \hat{\mathbb{Z}} + \hat{\mathbb{B}} \\ \hat{\mathbb{B}} & = & \{\bot, T, F, \top\} \\ Loc & = & Var \end{array}$$

을 가지고 다음의 재귀함수의 내용으로 만들어지는 방정식의 해가 되겠다:

$$\hat{\mathcal{C}} \in Cmd \to \widehat{Memory} \to \widehat{Memory}$$
$$\hat{\mathcal{V}} \in Expr \to \widehat{Memory} \to \widehat{Value}$$

$$
\begin{array}{rcl}
\hat{\mathcal{C}}\ \texttt{skip}\ \hat{m} & = & \hat{m} \\
\hat{\mathcal{C}}\ x\ \texttt{:=}\ E\ \hat{m} & = & \hat{m}\{x \mapsto \hat{\mathcal{V}}\ E\ \hat{m}\} \\
\hat{\mathcal{C}}\ \texttt{if}\ E\ C_1\ C_2\ \hat{m} & = & (\hat{\mathcal{C}}\ C_1\ \hat{m}) \sqcup (\hat{\mathcal{C}}\ C_2\ \hat{m}) \\
\hat{\mathcal{C}}\ C_1\ \texttt{;}\ C_2\ \hat{m} & = & \hat{\mathcal{C}}\ C_2(\hat{\mathcal{C}}\ C_1\ \hat{m}) \\
\hat{\mathcal{C}}\ \texttt{while}\ E\ \texttt{do}\ C\ \hat{m} & = & \hat{m} \sqcup (\hat{\mathcal{C}}\ \texttt{while}\ E\ \texttt{do}\ C\ (\hat{\mathcal{C}}\ C\ \hat{m})) \\
\hat{\mathcal{V}}\ n\ \hat{m} & = & \alpha\{n\} \\
\hat{\mathcal{V}}\ x\ \hat{m} & = & \hat{m}\ x \\
\hat{\mathcal{V}}\ E_1\ \texttt{+}\ E_2\ \hat{m} & = & (\hat{\mathcal{V}}\ E_1\ \hat{m})\ \hat{+}\ (\hat{\mathcal{V}}\ E_2\ \hat{m}) \\
\hat{\mathcal{V}}\ E_1\ \texttt{<}\ E_2\ \hat{m} & = & (\hat{\mathcal{V}}\ E_1\ \hat{m}) \hat{<} (\hat{\mathcal{V}}\ E_2\ \hat{m})
\end{array}
$$

명령문 $C$의 요약해석 방정식

$$\hat{\mathcal{C}}\, C = \cdots$$

의 오른편은 위에서 표현한 재귀함수 $\hat{\mathcal{C}}$의 내용을 고스란히 가지는 상위의 함수

$$\hat{\mathcal{F}} : (Cmd \to \widehat{Memory} \to \widehat{Memory}) \to (Cmd \to \widehat{Memory} \to \widehat{Memory})$$

가 정의하는 $\hat{\mathcal{F}}\,\hat{\mathcal{C}}\,C$가 되겠다:

$$\begin{array}{rcl}
\hat{\mathcal{F}}\,\hat{\mathcal{C}}\,\texttt{skip}\,\hat{m} & = & \hat{m} \\
\hat{\mathcal{F}}\,\hat{\mathcal{C}}\,x\,\texttt{:=}\,E\,\hat{m} & = & \hat{m}\{x \mapsto \hat{\mathcal{V}}\,E\,\hat{m}\} \\
\hat{\mathcal{F}}\,\hat{\mathcal{C}}\,\texttt{if}\,E\,C_1\,C_2\,\hat{m} & = & (\hat{\mathcal{C}}\,C_1\,\hat{m}) \sqcup (\hat{\mathcal{C}}\,C_2\,\hat{m}) \\
\hat{\mathcal{F}}\,\hat{\mathcal{C}}\,C_1\,;\,C_2\,\hat{m} & = & \hat{\mathcal{C}}\,C_2(\hat{\mathcal{C}}\,C_1\,\hat{m}) \\
\hat{\mathcal{F}}\,\hat{\mathcal{C}}\,\texttt{while}\,E\,\texttt{do}\,C\,\hat{m} & = & \hat{m} \sqcup (\hat{\mathcal{C}}\,\texttt{while}\,E\,\texttt{do}\,C\,(\hat{\mathcal{C}}\,C\,\hat{m}))
\end{array}$$

요약해석 틀에 의해, $lfp\hat{\mathcal{F}}$이 $lfp\mathcal{F}$의 안전한 요약이려면, 증명할 것은 아래 둘 중 하나이다:

- $\alpha \circ \mathcal{F} \sqsubseteq \hat{\mathcal{F}} \circ \alpha$, 즉, 모든 프로그램 $C$에 대해서

$$\forall f : (\alpha(\mathcal{F}\ f))\ C \sqsubseteq (\hat{\mathcal{F}}(\alpha\ f))\ C,$$

  혹은,

- $\alpha\ f \sqsubseteq \hat{f}$ 이면 $\alpha(\mathcal{F}\ f) \sqsubseteq \hat{\mathcal{F}}\ \hat{f}$, 즉, 모든 프로그램 $C$에 대해서

$$(\alpha(\mathcal{F}\ f))\ C \sqsubseteq (\hat{\mathcal{F}}\ \hat{f})\ C.$$

여기서

$$(Cmd \to Memory \to Memory) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (Cmd \to \hat{Memory} \to \hat{Memory})$$

이고, 증명은 $C$의 각 경우별로 따지면 되는 데, 이 때

$$Memory \underset{\alpha_1}{\overset{\gamma_1}{\leftrightarrows}} \hat{Memory}$$

를 가지고 조립식으로 정의된 $\alpha$가 사용된다.

# 구현

분석의 구현은, 분석할 프로그램 $C$가 주어졌을 때 연립방정식

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

을 풀면된다. 연립방정식은 $C$안에 있는 모든 명령문들 $C_i$와 식들 $E_i$의 요약 의미

$$\hat{\mathcal{C}}\ C_i \in \widehat{Memory} \to \widehat{Memory}$$
$$\hat{\mathcal{V}}\ E_i \in \widehat{Memory} \to \widehat{Value}$$

에 대한 방정식.

방정식의 해(프로그램의 요약된 의미)는 메모리에서 메모리로 가는 함수가 된다.

# 예

다음 프로그램

$$x \mathrel{:=} 1 \ ; \ \texttt{while} \ (0 < x) \ \texttt{do} \ x \mathrel{:=} x + 1$$

을 생각하자. 각 부품마다 번호를 붙이자.

$$\underbrace{\underbrace{x \mathrel{:=} 1}_{1} \ ; \ \underbrace{\texttt{while} \ (0 < x) \ \texttt{do} \ \underbrace{x \mathrel{:=} \underbrace{x + 1}_{4}}_{3}}_{2}}_{0}$$

각 부품의 의미 $\hat{\mathcal{C}}\, C_i$와 $\hat{\mathcal{V}}\, E_i$에 대한 방정식은 각각 $\hat{\mathcal{F}}\, C_i$와 $\hat{\mathcal{V}}\, E_i$에 의해서 아래와 같이 정의된다:

$$
\begin{array}{rcl}
\hat{\mathcal{C}}\, C_0 & = & \lambda\hat{m}.\hat{\mathcal{C}}\, C_2(\hat{\mathcal{C}}\, C_1\ \hat{m}) \\
\hat{\mathcal{C}}\, C_1 & = & \lambda\hat{m}.\hat{m}\{x \mapsto \hat{\mathcal{V}}\, 1\ \hat{m}\} \\
 & = & \lambda\hat{m}.\hat{m}\{x \mapsto \alpha\{1\}\} \\
\hat{\mathcal{C}}\, C_2 & = & \lambda\hat{m}.\hat{m} \sqcup (\hat{\mathcal{C}}\, C_2\ (\hat{\mathcal{C}}\, C_3\ \hat{m})) \\
\hat{\mathcal{C}}\, C_3 & = & \lambda\hat{m}.\hat{m}\{x \mapsto \hat{\mathcal{V}}\, E_4\ \hat{m}\} \\
 & = & \lambda\hat{m}.\hat{m}\{x \mapsto (\hat{m}\ x)\ \hat{+}\ \alpha\{1\}\}
\end{array}
$$

방정식의 주인공 $\hat{\mathcal{C}}\, C_i$를 $\hat{X}_i$로 해서 다시 쓰면,

$$
\begin{array}{rcl}
\hat{X}_0 & = & \lambda\hat{m}.\hat{X}_2(\hat{X}_1\ \hat{m}) \\
\hat{X}_1 & = & \lambda\hat{m}.\hat{m}\{x \mapsto \alpha\{1\}\} \\
\hat{X}_2 & = & \lambda\hat{m}.\hat{m} \sqcup (\hat{X}_2\ (\hat{X}_3\ \hat{m})) \\
\hat{X}_3 & = & \lambda\hat{m}.\hat{m}\{x \mapsto (\hat{m}\ x)\ \hat{+}\ \alpha\{1\}\}
\end{array}
$$

# $2^{\mathbb{Z}} \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \hat{\mathbb{Z}}$의 두가지 경우를 살피자

- $\hat{\mathbb{Z}}$의 높이가 유한한 경우
- $\hat{\mathbb{Z}}$의 높이가 무한한 경우

$$
\begin{array}{rcl}
\hat{X}_0 & = & \lambda\hat{m}.\hat{X}_2(\hat{X}_1\,\hat{m}) \\
\hat{X}_1 & = & \lambda\hat{m}.\hat{m}\{x \mapsto \alpha\{1\}\} \\
\hat{X}_2 & = & \lambda\hat{m}.\hat{m} \sqcup (\hat{X}_2\,(\hat{X}_3\,\hat{m})) \\
\hat{X}_3 & = & \lambda\hat{m}.\hat{m}\{x \mapsto (\hat{m}\,x) \mathbin{\hat{+}} \alpha\{1\}\}
\end{array}
$$

# $\hat{\mathbb{Z}}$의 높이가 유한한 경우

예를들어, $\hat{\mathbb{Z}} = \{\bot, -, +, \top\}$ 일 때, 위의 방정식은:

$$\begin{array}{rcl}
\hat{X}_0 & = & \lambda\hat{m}.\hat{X}_2(\hat{X}_1\ \hat{m}) \\
\hat{X}_1 & = & \lambda\hat{m}.\hat{m}\{x \mapsto +\} \\
\hat{X}_2 & = & \lambda\hat{m}.\hat{m} \sqcup (\hat{X}_2\ (\hat{X}_3\ \hat{m})) \\
\hat{X}_3 & = & \lambda\hat{m}.\hat{m}\{x \mapsto (\hat{m}\ x)\ \hat{+}\ +\}
\end{array}$$

프로그램 시작에서의 모든 메모리 상태를 포섭하는 것이

$$\{\} \in \widehat{Memory} = Var \xrightarrow{\text{fin}} \widehat{Value}$$

라고하면,

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

에서 부터 "연쇄반응"을 일으키는 방정식들만 푼다.

연쇄반응을 따라, 풀어야 할 방정식들:

$$\begin{array}{rcl} \hat{X}_0\,\{\} & = & \hat{X}_2(\hat{X}_1\,\{\}) \quad \text{따라서} \\ \hat{X}_1\,\{\} & = & \{x \mapsto +\} \quad \text{따라서} \\ \hat{X}_2\,\{x \mapsto +\} & = & \{x \mapsto +\} \sqcup (\hat{X}_2\,(\hat{X}_3\,\{x \mapsto +\})) \quad \text{따라서} \\ \hat{X}_3\,\{x \mapsto +\} & = & \{x \mapsto +\}\{x \mapsto + \hat{+} +\} \end{array}$$

다시 정리하면

$$\begin{array}{rcl} \hat{X}_0\,\{\} & = & \hat{X}_2\,\{x \mapsto +\} \\ \hat{X}_1\,\{\} & = & \{x \mapsto +\} \\ \hat{X}_2\,\{x \mapsto +\} & = & \{x \mapsto +\} \sqcup (\hat{X}_2\,\{x \mapsto +\}) \\ \hat{X}_3\,\{x \mapsto +\} & = & \{x \mapsto +\} \end{array}$$

위의 방정식을 다시 쓰면 ($\hat{X}_i\,\star$을 $\hat{Y}_i$로)

$$\begin{aligned}\hat{Y}_0 &= \hat{Y}_2\\ \hat{Y}_1 &= \{x \mapsto +\}\\ \hat{Y}_2 &= \{x \mapsto +\} \sqcup \hat{Y}_2\\ \hat{Y}_3 &= \{x \mapsto +\}\end{aligned}$$

위의 방정식의 최소해는 고정점 계산(*fixpoint iteration*) ($\sqcup_i (f^i \hat{\perp})$의 계산) 으로:

$$\begin{aligned}\hat{Y}_0 &= \{x \mapsto +\}\\ \hat{Y}_1 &= \{x \mapsto +\}\\ \hat{Y}_2 &= \{x \mapsto +\}\\ \hat{Y}_3 &= \{x \mapsto +\}\end{aligned}$$

분석결과: 프로그램의 모든 명령문 실행 후 $x$는 음이아닌 정수를 가진다.