

# SNU 4541.664A Program Analysis

## Note 18

Prof. Kwangkeun Yi



타입 스타일(형식 논리 시스템 스타일)로 타입이외의 것도 유추해보자.

- 예) 프로그램 실행중에 사용하는 메모리 부위는 어떻게 되나? effect analysis, region analysis
- 예) 프로그램 실행중에 호출되는 함수들은 어떤 것 들인가? control flow analysis
- $(\lambda x.(\lambda y.y (x 1))) (\lambda z.\lambda w.w + z) (\lambda a.a 2)$

$$(\lambda_0 x. (\lambda_1 y. y (x 1))) (\lambda_2 z. \lambda_3 w. w + z) (\lambda_4 a. a 2)$$

- 추론규칙(*inference rules*)은 " $\Gamma \vdash e : \tau, c$ " 꼴을 유추하는 규칙들
- $\llbracket e : \tau, c \rrbracket = \text{true}$   
iff " $e$ 가 문제없이 실행되며, 끝난다면 그 결과는  $\tau$  타입이며 그 실행중에  $c$ 에 있는 함수들이 호출된다."
- 가정들  $\Gamma$ 
  - 변수들에 대한 가정
  - $x + 1 : \iota, \emptyset$ , 가정  $x : \iota$  아래서.
  - $f 1 : \iota, \{l\}$ , 가정  $f : \iota \xrightarrow{\{l\}} \iota$  아래서
  - 가정이  $x : \tau$ 가 아니고  $x : \tau, c$  일 필요? cbv v.s. cbn

$$\overline{\Gamma \vdash n : \iota, \emptyset} \quad \overline{\Gamma \vdash x : \tau, \emptyset} \quad x : \tau \in \Gamma$$

$$\frac{\Gamma \vdash e_1 : \iota, c_1 \quad \Gamma \vdash e_2 : \iota, c_2}{\Gamma \vdash e_1 + e_2 : \iota, c_1 \cup c_2}$$

$$\frac{\Gamma + x : \tau \vdash e : \tau', c}{\Gamma \vdash \lambda x. e : \tau \rightarrow \tau', \emptyset}$$

$$\frac{\Gamma \vdash e_1 : \tau \rightarrow \tau', c_1 \quad \Gamma \vdash e_2 : \tau, c_2}{\Gamma \vdash e_1 e_2 : \tau, c_1 \cup c_2}$$

그러면,

$$\vdash (\lambda_0 x. x 1)(\lambda_1 y. 2) : \iota, ?$$

# 어떤가?

$$\tau \rightarrow \iota \mid \tau \xrightarrow{c} \tau \quad c \in 2^{LamLabel}$$

$$\overline{\Gamma \vdash n : \iota, \emptyset} \quad \overline{\Gamma \vdash x : \tau, \emptyset} \quad x : \tau \in \Gamma$$

$$\frac{\Gamma \vdash e_1 : \iota, c_1 \quad \Gamma \vdash e_2 : \iota, c_2}{\Gamma \vdash e_1 + e_2 : \iota, c_1 \cup c_2} \quad \frac{\Gamma + x : \tau \vdash e : \tau', c}{\Gamma \vdash \lambda_1 x. e : \tau \xrightarrow{c} \tau', \emptyset}$$

$$\frac{\Gamma \vdash e_1 : \tau \xrightarrow{c} \tau', c_1 \quad \Gamma \vdash e_2 : \tau, c_2}{\Gamma \vdash e_1 e_2 : \tau, c_1 \cup c_2 \cup c}$$

그래서,

$$\vdash (\lambda_0 x. x : 1)(\lambda_1 y. 2) : \iota, ?$$

# 어떤가?

$$\overline{\Gamma \vdash n : \iota, \emptyset} \quad \overline{\Gamma \vdash x : \tau, \emptyset} \quad x : \tau \in \Gamma$$

$$\frac{\Gamma \vdash e_1 : \iota, c_1 \quad \Gamma \vdash e_2 : \iota, c_2}{\Gamma \vdash e_1 + e_2 : \iota, c_1 \cup c_2} \quad \frac{\Gamma + x : \tau \vdash e : \tau', c}{\Gamma \vdash \lambda_l x. e : \tau \xrightarrow{c \cup \{l\}} \tau', \emptyset}$$

$$\frac{\Gamma \vdash e_1 : \tau \xrightarrow{c} \tau', c_1 \quad \Gamma \vdash e_2 : \tau, c_2}{\Gamma \vdash e_1 e_2 : \tau, c_1 \cup c_2 \cup c}$$

그래서,

$$\vdash (\lambda_0 x. x : 1)(\lambda_1 y. 2) : \iota, ?$$

그런데

$$\vdash (\lambda k. k (\lambda y. 2) \cdots k (\lambda z. 3)) (\lambda x. x 1) : \tau, ?$$



$$\overline{\Gamma \vdash n : \iota, \emptyset} \quad \overline{\Gamma \vdash x : \tau, \emptyset} \quad x : \tau \in \Gamma$$

$$\frac{\Gamma \vdash e_1 : \iota, c_1 \quad \Gamma \vdash e_2 : \iota, c_2}{\Gamma \vdash e_1 + e_2 : \iota, c_1 \cup c_2} \quad \frac{\Gamma + x : \tau \vdash e : \tau', c}{\Gamma \vdash \lambda_l x. e : \tau \xrightarrow{c \cup \{l\}} \tau', \emptyset}$$

$$\frac{\Gamma \vdash e_1 : \tau \xrightarrow{c} \tau', c_1 \quad \Gamma \vdash e_2 : \tau, c_2}{\Gamma \vdash e_1 e_2 : \tau, c_1 \cup c_2 \cup c}$$

$$\frac{\Gamma \vdash e : c}{\Gamma \vdash e : c'} \quad c \subseteq c'$$

그래서,

$$\vdash (\lambda k. k (\lambda y. 2) \cdots k (\lambda z. 3)) (\lambda x. x 1) : \tau, ?$$

좁은 보폭으로 실행의미를 정의하면:

- $\vdash e : \tau, c$ 이고  $e \rightarrow e'$  이면  $\vdash e' : \tau, c$ ?
- $\vdash e : \tau, c$ 이고  $e \xrightarrow{c_1} e'$  이면  $\vdash e' : \tau, c_2$  이고  $c_1 \cup c_2 \subseteq c$ !
- 증명 가능, 물론

# 추론 규칙의 안전성 증명

이번엔, 큰 보폭으로 실행의미를 정의해서 해 보자

- 실용적: 텍스트와 의미가 별개 (C, ML, Java, C#의 의미 정의)
- 타입 시스템의 안전성 증명기법?
- 타입스타일로 정의한 분석기의 안전성 증명기법?
- 언제나 귀납법
  - 프로그램 구조에 대한 (“by structural induction on  $e$ ”) 또는
  - 증명나무에 대한 (“by induction on derivation”) 또는
  - 타입구조에 대한 (“by logical relation”)

# 큰 보폭으로 정의한 실행의미

$$v \in \text{Value} = \mathbb{Z} + \text{Closure}$$

$$\sigma \in \text{Env} = \text{Id} \xrightarrow{\text{fin}} \text{Value}$$

$$\text{Closure} = \text{Expr} \times \text{Env}$$

$$\frac{}{\sigma \vdash n \Rightarrow n} \quad \frac{}{\sigma \vdash x \Rightarrow v} \quad \sigma(x) = v$$

$$\frac{}{\sigma \vdash \lambda x.e \Rightarrow \langle \lambda x.e, \sigma \rangle}$$

$$\frac{\sigma \vdash e_1 \Rightarrow \langle \lambda x.e', \sigma' \rangle \quad \sigma \vdash e_2 \Rightarrow v \quad \sigma'\{x \mapsto v\} \vdash e' \Rightarrow v}{\sigma \vdash e_1 e_2 \Rightarrow v}$$

Subject Reduction:  $\vdash e : \tau$  이고  $\vdash e \Rightarrow v$  이면  $v : \tau$

“ $v : \tau$ ”를 어떻게 정의하는가? 논리관계(logical relation)로

# 논리관계(Logical Relation)

- 논리관계(Logical Relation)는 타입을 갖춘 언어(*typed language*)에 대해 사용하는 공리 도구(reasoning tool)
- 논리관계(Logical Relation)는 타입구조를 타고 귀납적으로 정의된다

예를들어, 값과 타입 사이의 관계:

$$\begin{aligned}\tau &\rightarrow \iota \mid \tau \rightarrow \tau \\ v &\in \text{Value} = \mathbb{Z} + \text{Closure} \\ \sigma &\in \text{Env} = \text{Id} \xrightarrow{\text{fin}} \text{Value} \\ &\text{Closure} = \text{Expr} \times \text{Env}\end{aligned}$$

$$\begin{aligned}n &: \iota && \text{iff true} \\ \langle \lambda x.e, \sigma \rangle &: \tau_1 \rightarrow \tau_2 && \text{iff } \forall v : \tau_1. (\sigma\{v \mapsto x\} \vdash e \Rightarrow v') \text{ then } v' : \tau_2\end{aligned}$$

# 타입 시스템의 안전성 증명을 위한 실행의미를 정의

$$\sigma \vdash e \Rightarrow v, f$$

“식  $e$ 가 실행되고 값이  $v$ 이고 실행중에  $f \in 2^{LamLabel}$ 에 있는 함수들이 호출된다.”

$$\frac{}{\sigma \vdash n \Rightarrow n, \emptyset} \quad \frac{}{\sigma \vdash x \Rightarrow v, \emptyset} \quad \sigma(x) = v$$

$$\frac{}{\sigma \vdash \lambda x.e \Rightarrow \langle \lambda x.e, \sigma \rangle, \emptyset}$$

$$\frac{\sigma \vdash e_1 \Rightarrow \langle \lambda_l x.e', \sigma' \rangle, f_1 \quad \sigma \vdash e_2 \Rightarrow v, f_2 \quad \sigma' \{x \mapsto v\} \vdash e' \Rightarrow v, f_3}{\sigma \vdash e_1 e_2 \Rightarrow v, f_1 \cup f_2 \cup f_3 \cup \{l\}}$$

(정의  $\sigma \vdash e \Rightarrow \text{error}$  하고) Subject Reduction

$(\Gamma \vdash e : \tau, c) \wedge (\sigma \models \Gamma) \wedge (\sigma \vdash e \Rightarrow v, f)$  이면  $v : \tau \wedge f \subseteq c$

여기서,  $\sigma \models \Gamma$  (“ $\sigma$ 는  $\Gamma$ 를 존중”)의 정의는

$$\sigma \models \Gamma \text{ iff } \forall x \in \text{Dom } \Gamma. \sigma(x) : \Gamma(x)$$

이고  $v : \tau$ 의 정의는

$$\begin{aligned} n & : \iota && \text{iff true} \\ \langle \lambda x.e, \sigma \rangle & : \tau_1 \xrightarrow{c} \tau_2 && \text{iff } \forall v : \tau_1. (\sigma\{v \mapsto x\} \vdash e \Rightarrow v', f) \\ & && \text{then } v' : \tau_2 \wedge f \subseteq c. \end{aligned}$$

# 방정식과 집합제약식

- 방정식: 타입을 위해서
- 집합제약식(*set constraint*): 부가 분석내용을 위해서

$u$	$\rightarrow u_t \wedge u_c$	두 종류
$u_t$	$\rightarrow \tau \doteq \tau$	타입 방정식
	$  u_t \wedge u_t$	연립
$\tau$	$\rightarrow \alpha$	타입 변수
	$  \iota \mid \tau \xrightarrow{\varphi} \tau$	
$u_c$	$\rightarrow \varphi \supseteq c$	부가분석 집합제약식
	$  u_c \wedge u_c$	연립
$c$	$\rightarrow \{l_1, \dots, l_k\}$	상수
	$  \varphi$	부가분석 변수
	$  c \cup c$	합집합 식



$$V(\Gamma, n, \tau, \varphi) = \tau \doteq \iota \wedge \varphi \supseteq \emptyset$$

$$V(\Gamma, x, \tau, \varphi) = \tau \doteq \tau' \quad \text{if } x : \tau' \in \Gamma \\ \wedge \varphi \supseteq \emptyset$$

$$V(\Gamma, e_1 + e_2, \tau, \varphi) = \tau \doteq \iota \wedge V(\Gamma, e_1, \iota, \varphi_1) \wedge V(\Gamma, e_2, \iota, \varphi_2) \\ \wedge \varphi \supseteq \varphi_1 \cup \varphi_2 \\ \text{new } \varphi_1, \varphi_2$$

$$V(\Gamma, \lambda l x. e, \tau, \varphi) = \tau \doteq \alpha_1 \xrightarrow{\varphi_1} \alpha_2 \wedge V(\Gamma + x : \alpha_1, e, \alpha_2, \varphi') \\ \wedge \varphi \supseteq \emptyset \wedge \varphi_1 \supseteq \varphi' \cup \{l\} \\ \text{new } \alpha_1, \alpha_2, \varphi_1, \varphi'$$

$$V(\Gamma, e_1 e_2, \tau, \varphi) = V(\Gamma, e_1, \alpha \xrightarrow{\varphi'} \tau, \varphi_1) \wedge V(\Gamma, e_2, \alpha, \varphi_2) \\ \wedge \varphi \supseteq \varphi_1 \cup \varphi_2 \cup \varphi' \\ \text{new } \alpha, \varphi_1, \varphi_2, \varphi'$$

# 방정식 세우기 $V(\Gamma, e, \tau, c)$ 는 옳다

즉,

$$S \models V(\Gamma, e, \tau, c) \Leftrightarrow S\Gamma \vdash e : S\tau, Sc$$

증명은  $e$ 의 구조에 대한 귀납법으로.

# 방정식과 집합제약식의 해 구하기

$$V(\Gamma, e, \alpha, \varphi) = u_t \wedge u_c$$

- 타입 연립방정식  $u_t$  는 동일화(*unification*) 알고리즘으로:  
해  $S_1$
- 부가분석 집합제약식  $u_c$  은
  - $S_1$ 을 반영한 집합제약식  $S_1 u_c$ 에서 집합 방정식  $w_c$ 로 변환:  
각각의  $\varphi$ 에 대한 모든 집합제약식들

$$\varphi \supseteq c_1 \wedge \cdots \wedge \varphi \supseteq c_k \text{ 를 } \varphi = c_1 \cup \cdots \cup c_k \text{ 로.}$$

- $w_c$ 은 고정점 계산(*fixpoint iteration*) 방식으로 최소 해를  
계산.