SNU 프로그래밍언어 특강

(2.0)

이 광근

kwangkeunyi.snu.ac.kr

타입$_{\text{type}}$

- 프로그래밍언어를 이해하는/디자인하는 틀
  - 타입마다 값계산을 표현하는 방법들$_{\text{constructs}}$
    = 언어 겉모습$_{\text{syntax}}$
  - 타입에 맞게 값들이 계산되는 소용돌이$_{\text{statics + dynamics}}$
    = 언어 속내용$_{\text{semantics}}$
  - "값계산" = 값 만들기 + 값 사용하기

그래서

- 실행전 의미$_{\text{static semantics, 정적의미}}$: 타입으로 따져보는
- 실행 의미$_{\text{dynamic semantics, 동적의미}}$: 실행으로 따져보는

계획 (1/2)

타입은 프로그래밍을 비추고
프로그래밍은 타입을 이끌고

- 간단한 타입$_{\text{simple type}}$: 기본타입으로 합성하는 상식
- 여러모양 타입$_{\text{polymorphic type}}$: (타입이 뒤따랐던 프로래밍)
- 값에기대는 타입$_{\text{dependent type}}$: (타입이 이끈 프로래밍)

계획 (2/2)

타입은 프로그래밍을 비추고
프로그래밍은 타입을 이끌고

- 커리-하워드 대응$_{\text{curry-howard correspondence}}$:

$$\text{프로그램과 타입} \xleftrightarrow{\text{거울}} \text{증명과 명제}$$

  - 값에기대는 타입$_{\text{dependent type}}$과 함께
    = 증명중심 프로그래밍$_{\text{proof-oriented pgm'ng}}$(Rocq, Lean)을 연 열쇠
- system F, parametric polymorphism, ad-hoc polymorphism, bounded polymorphism, type class, recursive type, abstract data type, subtype, lambda cube, let-polymorphic types, type checking, type inference

간단한 타입simple type

기본값과 기본 합성값(함수, 짝, 갈래)타입

$$
\begin{array}{llll}
\text{Type} \quad \tau & \rightarrow & \iota & (\text{기본타입:int, bool}, \cdots) \\
& \mid & \tau \rightarrow \tau & (\text{함수타입, 함수, 이면}) \\
& \mid & \tau \times \tau & (\text{곱타입, 짝, 그리고}) \\
& \mid & \tau + \tau & (\text{합타입, 갈래, 또는})
\end{array}
$$

간단한 타입$_{\text{simple type}}$ 언어

$$
\begin{array}{llcl}
\text{Exp} & E & \rightarrow & n \\
 & & | & x \\
 & & | & \texttt{-}\ E \\
 & & | & \texttt{fn}\ x\ E \\
 & & | & E\ E
\end{array}
\qquad
\begin{array}{llcl}
\text{Type} & \tau & \rightarrow & \text{int} \\
 & & | & \tau \rightarrow \tau
\end{array}
$$

타입으로 따지기: 실행전의미$_{\text{static semantics}}$

$$\Gamma \vdash E : \tau$$

의 증명. (타입환경$_{\text{type env}}$ $\Gamma \in \text{Var} \xrightarrow{\text{fin}} \text{Type}$)

$$\frac{}{\Gamma \vdash n : \text{int}} \qquad \frac{\Gamma \vdash E : \text{int}}{\Gamma \vdash \texttt{-}\, E : \text{int}}$$

$$\frac{\Gamma(x) = \tau}{\Gamma \vdash x : \tau}$$

$$\frac{\Gamma + x : \tau_1 \vdash E : \tau_2}{\Gamma \vdash \texttt{fn}\; x\; E : \tau_1 \to \tau_2} \qquad \frac{\Gamma \vdash E_1 : \tau_1 \to \tau_2 \quad \Gamma \vdash E_2 : \tau_1}{\Gamma \vdash E_1\; E_2 : \tau_2}$$

실행전의미$_{\text{static semantics}}$가 이치에맞나?

실행의미$_{\text{dynamic semantics}}$와 발맞춰야

- 안전함$_{\text{sound}}$: 실행전의미가 있으면 잘돈다.
- 완전함$_{\text{complete}}$? 잘돌면 실행전의미가 있다.

실행전의미$_{\text{static semantics}}$ $=$ 타입검사$_{\text{type checking}}$(의 기반)

실행전의미$_{\text{static semantics}}$의 안전함 증명

Theorem (안전함$_{\text{soundness, correctness, type safety}}$)

$\forall E. \vdash E : \tau$이면 E는 문제없이 실행되고 끝난다면 결과값은 τ타입의 값이다.

작은보폭($E \to E'$) 실행의미 경우

- 증명

  $\vdash E : \tau$ 이고 $E \notin$ Value 이면 $E \to E'$ “progress lemma”
- 증명

  $\vdash E : \tau$ 이고 $E \to E'$ 이면 $\vdash E' : \tau$ “preservation lemma”

두 증명에 의해 안전함.

실행전의미$_{\text{static semantics}}$의 안전함 증명

Theorem (안전함$_{\text{soundness, correctness, type safety}}$)

$\forall E. \vdash E : \tau$*이면* E*는 문제없이 실행되고 끝난다면 결과값은* τ*타입의 값이다.*

큰보폭($\sigma \vdash E \Rightarrow v$) 실행의미 경우

- 다음 증명이면 충분
  $\vdash E : \tau$ 이면 $\vdash E \Rightarrow r$ 이면 $r : \tau$
  - $r \in \text{Value} + \{\text{error}\}$, $\sigma \vdash E \Rightarrow \text{error}$ 정의필요
- $v : \tau$ 정의 "logical relation"
  - $n : \text{int}$ iff true
  - $(\texttt{fn}\ x\ E, \sigma) : \tau_1 \to \tau_2$ iff
    $\forall v_1 : \tau_1 . \sigma + x : v_1 \vdash E \Rightarrow r$ 이면 $r : \tau_2$

$\tau \times \tau$ 곱타입$_{\text{product type}}$

$$\begin{array}{llll} \text{Exp} & E & \to & \cdots \\ & & | & \texttt{(}E\texttt{,}E\texttt{)} \mid E\texttt{.l} \mid E\texttt{.r} \end{array} \qquad \begin{array}{llll} \text{Type} & \tau & \to & \cdots \\ & & | & \tau \times \tau \end{array}$$

실행전의미$_{\text{static semantics}}$

$$\frac{\Gamma \vdash E_1 : \tau_1 \quad \Gamma \vdash E_1 : \tau_1}{\Gamma \vdash \texttt{(}E_1\texttt{,}E_2\texttt{)} : \tau_1 \times \tau_2}$$

$$\frac{\Gamma \vdash E : \tau_1 \times \tau_2}{\Gamma \vdash E\texttt{.l} : \tau_1} \qquad \frac{\Gamma \vdash E : \tau_1 \times \tau_2}{\Gamma \vdash E\texttt{.r} : \tau_2}$$

참고: 레코드타입$_{\text{record type}}$ = 곱타입 + 부품이름

$$\texttt{(a:}E\texttt{,b:}E\texttt{)} \mid E\texttt{.a} \mid E\texttt{.b}$$

$\tau + \tau$ 합타입$_{\text{sum type}}$

$$\begin{array}{llll} \text{Exp} & E & \rightarrow & \cdots \\ & & \mid & \texttt{inl}\ E \mid \texttt{inr}\ E \\ & & \mid & \texttt{case}\ E\ (\texttt{inl}\ x : E)\ (\texttt{inr}\ x : E) \end{array} \qquad \begin{array}{llll} \text{Type} & \tau & \rightarrow & \cdots \\ & & \mid & \tau + \tau \end{array}$$

실행전의미$_{\text{static semantics}}$

$$\frac{\Gamma \vdash E : \tau_1}{\Gamma \vdash \texttt{inl}\ E : \tau_1 + \tau_2} \qquad \frac{\Gamma \vdash E : \tau_2}{\Gamma \vdash \texttt{inr}\ E : \tau_1 + \tau_2}$$

$$\frac{\Gamma \vdash E : \tau_1 + \tau_2 \quad \Gamma + x : \tau_1 \vdash E_1 : \tau \quad \Gamma + x : \tau_2 \vdash E_2 : \tau}{\Gamma \vdash \texttt{case}\ E\ (\texttt{inl}\ x : E_1)\ (\texttt{inr}\ x : E_2) : \tau}$$

참고: 갈래타입$_{\text{variant type}}$ $=$ 합타입 $+$ 부품(출신)이름

$$\mathsf{a}\ E \mid \mathsf{b}\ E \mid \mathsf{case}\ E\ (\mathsf{a}\ x : E)\ (\mathsf{b}\ x : E)$$

주소 다루는 식의 타입

$$\begin{array}{llll} \text{Exp} & E & \to & \cdots \\ & & \mid & \texttt{ref}\ E \mid E \texttt{ := } E \mid \texttt{!}E \end{array} \qquad \begin{array}{llll} \text{Type} & \tau & \to & \cdots \\ & & \mid & \tau\ \text{loc} \end{array}$$

실행전의미$_{\text{static semantics}}$

$$\frac{\Gamma \vdash E : \tau}{\Gamma \vdash \texttt{ref}\ E : \tau\ \text{loc}}$$

$$\frac{\Gamma \vdash E_1 : \tau\ \text{loc} \qquad \Gamma \vdash E_2 : \tau}{\Gamma \vdash E_1 \texttt{ := } E_2 : \tau}$$

$$\frac{\Gamma \vdash E : \tau\ \text{loc}}{\Gamma \vdash \texttt{!}E : \tau}$$

두드러지게 불완전한 실행전의미: “잘 도는데 타입검사가 안되는” 프로그램들이 뻔함

할일 다루는 식의 타입

$$\begin{array}{lllll} \text{Exp} & E & \to & \cdots & \\ & & | & \texttt{catch}\, x\, E \mid \texttt{throw}\, x\, E & \end{array} \qquad \begin{array}{llll} \text{Type} & \tau & \to & \cdots \end{array}$$

실행의미$_{\text{dynamic semantics}}$

$$\begin{array}{rcl} [\![\texttt{catch}\, x\, E]\!]\, \sigma\, k & = & [\![E]\!]\, \sigma\{x \mapsto k\}\, k \\ [\![\texttt{throw}\, x\,]\!]\, \sigma\, k & = & [\![E]\!]\, \sigma\, (\lambda v.\sigma(x)(v)) \end{array} \qquad \begin{array}{l} C[\texttt{catch}\, x\, E] \to C[\{\lambda v.C[v]/x\}E] \\ C[\texttt{throw}\, (\lambda v.E')\, E] \to (\lambda v.E')E \end{array}$$

실행전의미$_{\text{static semantics}}$

$$\frac{\Gamma + x : \tau \to \tau' \vdash E : \tau}{\Gamma \vdash \texttt{catch}\, x\, E : \tau}$$

$$\frac{\Gamma(x) = \tau \to \tau' \qquad \Gamma \vdash E : \tau}{\Gamma \vdash \texttt{throw}\, x\, E : \tau''}$$

실행의미가 다음과 같으면?

$$\begin{array}{llll} \textsf{Exp} & E & \to & \cdots \\ & & | & \texttt{catch}\,x\,E \mid \texttt{throw}\,x\,E \end{array} \qquad \begin{array}{llll} \textsf{Type} & \tau & \to & \cdots \end{array}$$

실행의미$_{\text{dynamic semantics}}$

$$\begin{array}{rcl} [\![\texttt{catch}\,x\,E]\!]\,\sigma\,k & = & [\![E]\!]\,\sigma\{x \mapsto k\}\,\mathit{id} \\ [\![\texttt{throw}\,x\,]\!]\,\sigma\,k & = & [\![E]\!]\,\sigma\,(\lambda v.\sigma(x)(v)) \end{array} \qquad \begin{array}{l} C[\texttt{catch}\,x\,E] \to \{\lambda v.C[v]/x\}E \\ C[\texttt{throw}\,(\lambda v.E')\,E] \to (\lambda v.E')E \end{array}$$

실행전의미$_{\text{static semantics}}$

$$\frac{\Gamma + x : \tau \to \tau' \vdash E : \tau'}{\Gamma \vdash \texttt{catch}\,x\,E : \tau}$$

$$\frac{\Gamma(x) = \tau \to \tau' \qquad \Gamma \vdash E : \tau}{\Gamma \vdash \texttt{throw}\,x\,E : \tau''}$$

어느 디자인이 “맞을까”?

$$C[\texttt{catch}\, x\, E] \quad \to \quad C[\{\lambda v.C[v]/x\}E]$$

vs

$$C[\texttt{catch}\, x\, E] \quad \to \quad \{\lambda v.C[v]/x\}E$$

- 나침반: 커리-하워드 대응$_{\text{curry-howard correspondence}}$

$$\text{프로램체계} \overset{\text{거울}}{\longleftrightarrow} \text{논리체계}$$

- 두번째 디자인이 “맞아보임”:

$$\frac{\Gamma + x : \tau \to \tau' \vdash E : \tau'}{\Gamma \vdash \texttt{catch}\, x\, E : \tau} \quad \overset{\text{거울}}{\longleftrightarrow} \quad \frac{\neg\neg A}{A}$$

$$\frac{\Gamma(x) = \tau \to \tau' \qquad \Gamma \vdash E : \tau}{\Gamma \vdash \texttt{throw}\, x\, E : \tau''} \quad \overset{\text{거울}}{\longleftrightarrow} \quad \frac{\bot}{A}$$

어느 디자인이 “맞을까”? 되돌아보니

$$C[\texttt{catch}\, x\ E] \;\rightarrow\; C[\{\lambda v.C[v]/x\}E]$$

vs

$$C[\texttt{catch}\, x\ E] \;\rightarrow\; \{\lambda v.C[v]/x\}E$$

- 첫번째 디자인은
  - 제한없는 goto를 반영
  - 넌센스: $C[]$를 예외상황 처리식으로 본다면, E가 정상실행되도 $C[]$를 실행한다니!
- 두번째 디자인은 (논리체계 거울이 안내한)
  - 말이됨: $C[]$를 예외상황 처리식으로 본다면, E가 실행중 예외상황때만 $C[]$를 실행함
  - 특별한 goto(예외상황처리)만 유도

애초부터 감잡았던: “Letters to the editor: goto statement considered harmful” Edgar Dijkstra, *Communications of the ACM*, March, 1968

코드 다루는 식의 타입

$$
\begin{array}{llll}
\text{Exp} & E & \to & \cdots \\
& & | & \texttt{box}\ E \mid \texttt{unbox}\ E \\
& & | & \texttt{run}\ E
\end{array}
\qquad
\begin{array}{llll}
\text{Type} & \tau & \to & \cdots \\
& & | & \Box(\Gamma \triangleright \tau)
\end{array}
$$

실행전의미$_{\text{static semantics}}$ $\Sigma \vdash E : \tau$ $\quad (\Sigma \ \to\ \Gamma \mid \Sigma, \Gamma)$

$$
\frac{\Sigma, \Gamma \vdash E : \tau}{\Sigma \vdash \texttt{box}\ E : \Box(\Gamma \triangleright \tau)}
\qquad
\frac{\Sigma \vdash E : \Box(\Gamma \triangleright \tau)}{\Sigma, \Gamma \vdash \texttt{unbox}\ E : \tau}
$$

$$
\frac{\Sigma \vdash E : \Box(\emptyset \triangleright \tau)}{\Sigma \vdash \texttt{run}\ E : \tau}
$$

$\tau <: \tau'$, 작은타입$_{\text{subtype}}$ 관계가 있는 경우

- 이런 세계에서

$$v : \tau \text{ 이고 } \tau <: \tau' \text{ 이면 } v : \tau'$$

- $\tau <: \tau'$ 정의

$$\frac{}{\tau <: \tau} \qquad \frac{\tau <: \tau' \quad \tau' <: \tau''}{\tau <: \tau''} \qquad \frac{}{\mathsf{nat} <: \mathsf{int}}$$

$$\frac{\tau_1 <: \tau_1' \qquad \tau_2 <: \tau_2'}{\mathsf{a}\tau_1 \times \mathsf{b}\tau_2 \;<:\; \mathsf{a}\tau_1' \times \mathsf{b}\tau_2'} \qquad \frac{}{\mathsf{a}\tau_1 \times \mathsf{b}\tau_2 \;<:\; \mathsf{a}\tau_1} \qquad \frac{}{\mathsf{a}\tau_1 \times \mathsf{b}\tau_2 \;<:\; \mathsf{b}\tau_2} \qquad \text{레코드타입}_{\text{record type}}$$

$$\frac{\tau_1 <: \tau_1' \qquad \tau_2 <: \tau_2'}{\mathsf{a}\tau_1 + \mathsf{b}\tau_2 \;<:\; \mathsf{a}\tau_1' + \mathsf{b}\tau_2'} \qquad \frac{}{\mathsf{a}\tau \;<:\; \mathsf{a}\tau + \mathsf{b}\tau'} \qquad \frac{}{\mathsf{b}\tau \;<:\; \mathsf{a}\tau' + \mathsf{b}\tau} \qquad \text{갈래타입}_{\text{variant type}}$$

$$\frac{\tau_1' <: \tau_1 \qquad \tau_2 <: \tau_2'}{\tau_1 \to \tau_2 \;<:\; \tau_1' \to \tau_2'} \; \text{거스르는}_{\text{contra-variant}}$$

- 실행전의미$_{\text{static semantics}}$

$$\frac{\Gamma \vdash E : \tau \quad \tau <: \tau'}{\Gamma \vdash E : \tau'}$$

$\mu a.\tau$ 재귀타입 recursive type

$$\begin{array}{llll} \text{Type} & \tau & \to & \cdots \\ & & \mid & a \quad \text{타입변수} \\ & & \mid & \mu a.\tau \quad \text{타입방정식 } a = \tau\text{의 해} \end{array}$$

- 타입방정식 $a = \tau$의 답을 $\mu a.\tau$로 표현. 그러므로

$$\mu a.\tau \equiv \{\mu a.\tau / a\}\tau$$

예) 인덕데이터 타입방정식 list $= \mathit{unit} + \mathit{int} \times$ list의 답

$$\text{list} \stackrel{\text{let}}{=} \mu a.\mathit{unit} + \mathit{int} \times a$$

예) 인덕데이터 타입방정식 tree $= \mathit{int} + \mathit{int} \times$ tree $\times$ tree의 답

$$\text{tree} \stackrel{\text{let}}{=} \mu a.\mathit{int} + \mathit{int} \times a \times a$$

예) 타입방정식 $a = \mathit{int} \to a$의 답 $\mu a.(\mathit{int} \to a)$

예) 타입방정식 $a = a \to \mathit{int}$의 답 $\mu a.(a \to \mathit{int})$

$\mu a.\tau$ 재귀타입$_{\text{recursive type}}$으로 따지기

- $\mu a.\tau \equiv \{\mu a.\tau/a\}\tau$ 이므로

$$\frac{\Gamma \vdash E : \{\mu a.\tau/a\}\tau}{\Gamma \vdash E : \mu a.\tau} \qquad \frac{\Gamma \vdash E : \mu a.\tau}{\Gamma \vdash E : \{\mu a.\tau/a\}\tau}$$

예) 갈래타입방정식 $a = \mathsf{N}(unit) + \mathsf{C}(int \times a)$의 답
$\mu a.\mathsf{N}(unit) + \mathsf{C}(int \times a)$

$$\cfrac{\cfrac{\Gamma \vdash \mathsf{N}() : \mathsf{N}(unit)}{\Gamma \vdash \mathsf{N}() : \mathsf{N}(unit) + \mathsf{C}(int \times \mu a.\cdots)}}{\Gamma \vdash \mathsf{N}() : \mu a.\cdots} \qquad \cfrac{\cfrac{\cfrac{\vdots}{\Gamma \vdash \mathsf{C}(E_1, E_2) : \mathsf{C}(int \times \mu a.\cdots)}}{\Gamma \vdash \mathsf{C}(E_1, E_2) : \mathsf{N}(unit) + \mathsf{C}(int \times \mu a.\cdots)}}{\Gamma \vdash \mathsf{C}(E_1, E_2) : \mu a.\cdots}$$

예) 타입방정식 $a = int \to a$의 답 $\mu a.(int \to a)$

$$\cfrac{\cfrac{\mathsf{f} : \mu a.(int \to a) + \mathsf{x} : int \vdash \mathsf{f} : \mu a.(int \to a)}{\vdash \mathtt{rec}\,\mathsf{f}\,\mathsf{x}\,\mathsf{f} : int \to \mu a.(int \to a)}}{\vdash \mathtt{rec}\,\mathsf{f}\,\mathsf{x}\,\mathsf{f} : \mu a.(int \to a)}$$

예) 타입방정식 $a = a \to int$의 답 $\mu a.(a \to int)$

$$\cfrac{\vdots}{\mathsf{x} : \mu a.(a \to int) \vdash \mathsf{x}\,\mathsf{x} : int}$$