SNU 프로그래밍언어 특강

(1.2)

이 광근

kwangkeunyi.snu.ac.kr

개념/용어 리뷰

- 문법$_{\text{syntax}}$: 생김새, 의미$_{\text{semantics}}$: 속내용
- 집합 정의법: 인덕규칙$_{\text{inductive rules}}$, 인덕$_{\text{induction}}$, 코덕$_{\text{coinduction}}$
- 고정점$_{\text{fixpoint}}$: 최소, 최대
- 무엇 의미구조$_{\text{denotational semantics}}$: CPO, 연속함수
- 실행 의미구조$_{\text{operational semantics}}$: 집합, 증명나무$_{\text{proof tree}}$, 실행발자국$_{\text{transition sequence}}$/실행문맥$_{\text{evaluation context}}$, 실행나무$_{\text{itree}}$
- 딱맞는 의미구조$_{\text{full abstraction semantics}}$
- 증명법: 인덕$_{\text{induction}}$, 고정점인덕$_{\text{fixpoint induction}}$, 고정점 정의 이용법, 겉보기 증명$_{\text{extentional proof}}$

사용한/할 의미개념/의미공간

(CPO/연속함수 vs 집합/유한함수, 구분없이)

- 환경environment

$$\begin{array}{rcl} Env & = & Var \to Thing \\ Thing & = & Value, \qquad Thing \quad = \quad Loc \end{array}$$

- 메모리memory

$$Memory = Loc \to Value$$

- 값value

$$\mathbb{V} \quad = \quad \mathbb{N} + (\mathbb{V} \to \mathbb{V}), \qquad \mathbb{V} \quad = \quad \mathbb{N} + (Exp \times Env)$$

- 마저할일continuation

$$\mathbb{K} \quad = \quad \mathbb{V} \to \mathbb{V}$$

계획

- 값중심언어 확장: 메모리 다루기 + 예외상황 다루기
- 마저할일continuation
- 마저할일드러내기continuation-passing-style transformation
- 할일을 값으로control as value

프로그래밍언어가 제공하는 값의 종류

- 기본값primitives: 숫자, 참거짓, 문자열, ...
- 합성값compounds: 함수, 짝, ...

값 종류마다, 만들기 $\rightleftharpoons$ 사용하기 방법을 제공

- 함수: $\texttt{fn}\, x\, E \rightleftharpoons E\, E$
- 짝: $(E, E) \rightleftharpoons E.\texttt{l},\ E.\texttt{r}$

값중심 언어 applicative language

$$
\begin{array}{rrcl}
\text{프로그램식} & E & \to & n \mid x \mid \texttt{fn}\, x\, E \mid \texttt{rec}\, f\, x\, E \mid E\, E \\
& & \mid & \texttt{let}\, x\, E\, E \mid \texttt{if}\, E\, E\, E \\
& & \mid & (E,E) \mid E\texttt{.l} \mid E\texttt{.r} \mid E + E \mid - E \\
\\
\text{실행문맥} & K & \to & [\,] \mid K\, E \mid v\, K \\
\text{값} & v & \to & n \mid \texttt{fn}\, x\, E \mid \texttt{rec}\, f\, x\, E
\end{array}
$$

$$
\frac{E \to E'}{K[E] \to K[E']}
$$

$$
\begin{array}{rcl}
(\texttt{fn}\, x\, E)\, v & \to & \{v/x\}E \\
(\texttt{rec}\, f\, x\, E)\, v & \to & \{(\texttt{rec}\, f\, x\, E)/f, v/x\}E
\end{array}
$$

값중심 언어applicative language 확장

메모리 다루기/메모리주소location를 값으로

$$
\begin{array}{rrcl}
\text{프로그램식} & E & \to & n \mid x \mid \texttt{fn}\, x\, E \mid \texttt{rec}\, f\, x\, E \mid E\, E \\
& & \mid & \texttt{ref}\, E \mid E := E \mid\, !E \\
\\
\text{실행문맥} & K & \to & [\,] \mid K\, E \mid v\, K \mid \cdots \\
\text{값} & v & \to & n \mid \texttt{fn}\, x\, E \mid \texttt{rec}\, f\, x\, E \\
& & \mid & l \\
\text{메모리} & M & \in & \mathit{Loc} \xrightarrow{\text{fin}} \mathit{Value}
\end{array}
$$

$$
\frac{M, E \to M', E'}{M, K[E] \to M', K[E']}
$$

$$
\begin{array}{rcl}
M, (\texttt{fn}\, x\, E)\, v & \to & M, \{v/x\}E \\
M, (\texttt{rec}\, f\, x\, E)\, v & \to & M, \{(\texttt{rec}\, f\, x\, E)/f, v/x\}E \\
M, l := v & \to & M\{l \mapsto v\}, v \\
\cdots & &
\end{array}
$$

값중심 언어applicative language

예외상황 다루기/튀는 실행흐름control/멀리가기non-local goto

$$
\begin{array}{rrcl}
\text{프로그램식} & E & \to & n \mid x \mid \texttt{fn}\, x\, E \mid \texttt{rec}\, f\, x\, E \mid E\, E \\
& & \mid & \texttt{raise}\, L \mid E\, \texttt{handle}\, L\, E
\end{array}
$$

$$
\begin{array}{rrcl}
\text{실행문맥} & K & \to & [\,] \mid K\, E \mid v\, K \mid \cdots \\
\text{값} & v & \to & n \mid \texttt{fn}\, x\, E \mid \texttt{rec}\, f\, x\, E \\
\text{예외상황} & \eta & \to & \underline{L}
\end{array}
$$

$$
\frac{E \to E'}{K[E] \to K[E']}
$$

$$
\begin{array}{rcl}
(\texttt{fn}\, x\, E)\, v & \to & \{v/x\}E \\
(\texttt{rec}\, f\, x\, E)\, v & \to & \{(\texttt{rec}\, f\, x\, E)/f, v/x\}E \\
v\, \texttt{handle}\, L\, E & \to & v \\
\texttt{raise}\, L & \to & \underline{L} \\
\underline{L}\, \texttt{handle}\, L\, E & \to & E \\
\cdots & &
\end{array}
$$

마저할일continuation

"방과후 집에오는 길에 놀이방에 들려 동생을 데려오렴"

- 발견: goto-식의 각잡힌 의미를 정의하려다 발견한 개념
- 사용: 프로그래머가 실행흐름을 자유롭게 다루게 하는데 사용하는 개념
- 안경: 번역기compiler가 중간단계에서 하는 변환을 바라보는 안경

마저할일continuation 예

- E_1 + E_2 에서 E_1 계산을 끝내고 마저 할 일은?
- E_1 + E_2 에서 E_2 계산을 끝내고 마저 할 일은?
- E_1 E_2: E_1 계산을 끝내고 마저 할 일은?
- `(goto L)` + E 에서 `goto`를 만나고 마저 할 일은?
- `(raise L)` + E 에서 `raise`를 만나고 마저 할 일은?

마저할일continuation 예

- E_1 + E_2 에서 E_1 계산을 끝내고 마저 할 일은?

  $\lambda x.x$ + E_2
- E_1 + E_2 에서 E_2 계산을 끝내고 마저 할 일은?

- E_1 E_2: E_1 계산을 끝내고 마저 할 일은?

- `(goto L)` + E 에서 `goto`를 만나고 마저 할 일은?

- `(raise L)` + E 에서 `raise`를 만나고 마저 할 일은?

마저할일continuation 예

- E_1 + E_2 에서 E_1 계산을 끝내고 마저 할 일은?

  $\lambda x.x$ + E_2
- E_1 + E_2 에서 E_2 계산을 끝내고 마저 할 일은?

  $\lambda x.v_1$ + x
- E_1 E_2: E_1 계산을 끝내고 마저 할 일은?

- `(goto L)` + E 에서 `goto`를 만나고 마저 할 일은?

- `(raise L)` + E 에서 `raise`를 만나고 마저 할 일은?

마저할일continuation 예

- E_1 + E_2 에서 E_1 계산을 끝내고 마저 할 일은?

  $\lambda x.x$ + E_2
- E_1 + E_2 에서 E_2 계산을 끝내고 마저 할 일은?

  $\lambda x.v_1$ + x
- E_1 E_2: E_1 계산을 끝내고 마저 할 일은?

  $\lambda x.x$ E_2
- `(goto L)` + E 에서 `goto`를 만나고 마저 할 일은?

- `(raise L)` + E 에서 `raise`를 만나고 마저 할 일은?

마저할일continuation 예

- E_1 + E_2 에서 E_1 계산을 끝내고 마저 할 일은?

  $\lambda x.x$ + E_2
- E_1 + E_2 에서 E_2 계산을 끝내고 마저 할 일은?

  $\lambda x.v_1$ + x
- E_1 E_2: E_1 계산을 끝내고 마저 할 일은?

  $\lambda x.x$ E_2
- `(goto L)` + E 에서 `goto`를 만나고 마저 할 일은?

  `L`이름이 붙은 마저할일로 건너뛰기
- `(raise L)` + E 에서 `raise`를 만나고 마저 할 일은?

마저할일continuation 예

- E_1 + E_2 에서 E_1 계산을 끝내고 마저 할 일은?

  $\lambda x.x$ + E_2
- E_1 + E_2 에서 E_2 계산을 끝내고 마저 할 일은?

  $\lambda x.v_1$ + x
- E_1 E_2: E_1 계산을 끝내고 마저 할 일은?

  $\lambda x.x$ E_2
- `(goto L)` + E 에서 `goto`를 만나고 마저 할 일은?

  `L`이름이 붙은 마저할일로 건너뛰기
- `(raise L)` + E 에서 `raise`를 만나고 마저 할 일은?

  `L` 예외상황때 하기로한 마저할일로 건너뛰기

마저할일continuation

- $K[E]$에서 E계산을 마치고(v) 마저 할 일은?

$$\lambda x.K[x]$$

- 즉, 위의 마저할일continuation을 호출하기

$$(\lambda x.K[x])\, v$$

마저할일continuation 사용한 의미구조: 예

$$E \quad \rightarrow \quad 1 \mid E + E$$

$$\begin{array}{rcl} [\![E]\!] & \in & (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N} \\ [\![1]\!]\, k & = & k(1) \\ [\![E_1 + E_2]\!]\, k & = & [\![E_1]\!]\, (\lambda n_1.[\![E_2]\!]\, (\lambda n_2.k(n_1 + n_2))) \end{array}$$

- $\forall E, k.k(val(E)) = [\![E]\!]\, k$

마저할일continuation 사용한 의미구조: 값중심 언어

$$E \;\to\; n \mid x \mid \texttt{fn}\, x\, E \mid \texttt{rec}\, \texttt{f}\, x\, E \mid E\, E$$

$$\begin{aligned}
[\![E]\!] &\in Env \to \mathbb{K} \to \mathbb{V} \\
\text{환경} \quad \sigma \in Env &= Var_\perp \to \mathbb{V} \\
\text{마저할일} \quad k \in \mathbb{K} &= \mathbb{V} \to \mathbb{V} \\
\text{값} \quad v \in \mathbb{V} &= \mathbb{N}_\perp + (\mathbb{V} \to \mathbb{K} \to \mathbb{V})
\end{aligned}$$

$$\begin{aligned}
[\![n]\!]\, \sigma\, k &= k(n) \\
[\![x]\!]\, \sigma\, k &= k(\sigma(x)) \\
[\![\texttt{fn}\, x\, E]\!]\, \sigma\, k &= k(\lambda v.\lambda k'.[\![E]\!]\, \sigma\{x \mapsto v\}\, k') \\
[\![\texttt{rec}\, f\, x\, E]\!]\, \sigma\, k &= k(lfp(\lambda \bullet .\lambda v.\lambda k'.[\![E]\!]\, \sigma\{x \mapsto v\}\{f \mapsto \bullet\}\, k')) \\
[\![E_1\, E_2]\!]\, \sigma\, k &= [\![E_1]\!]\, \sigma\, (\lambda f.[\![E_2]\!]\, \sigma\, (\lambda v.f\, v\, k))
\end{aligned}$$

마저할일continuation 사용한 의미구조: goto-문

$$
\begin{array}{rcl}
E & \to & n \mid x \mid \texttt{fn}\, x\, E \mid \texttt{rec f}\, x\, E \mid E\, E \\
 & \mid & L : E \quad L : E \mid \texttt{goto}\, L
\end{array}
$$

$$
\begin{array}{rcl}
[\![E]\!] & \in & Env \to \mathbb{K} \to \mathbb{V} \\
\text{환경} \quad \sigma \in Env & = & (Var + Label)_\perp \to \mathbb{V} \\
\text{마저할일} \quad k \in \mathbb{K} & = & \mathbb{V} \to \mathbb{V} \\
\text{값} \quad v \in \mathbb{V} & = & \mathbb{N}_\perp + (\mathbb{V} \to \mathbb{K} \to \mathbb{V}) + \mathbb{K}
\end{array}
$$

$$
\begin{array}{rcl}
[\![L_1 : E_1 \quad L_2 : E_2]\!]\, \sigma\, k & = & [\![E_1]\!]\, \sigma'\, k_2 \quad \text{이고} \\
 & & \sigma' = \sigma\{L_1 \mapsto k_1\}\{L_2 \mapsto k_2\} \\
 & & k_1 = \lambda v.[\![E_1]\!]\, \sigma'\, k_2 \\
 & & k_2 = \lambda v.[\![E_2]\!]\, \sigma'\, k \\
[\![\texttt{goto}\, L]\!]\, \sigma\, k & = & \sigma(L)(0)
\end{array}
$$

마저할일 드러내기 변환cps transformation: 값중심 언어

마저할일 드러내기continuation-passing-style, cps

$$E \;\rightarrow\; n \mid x \mid \texttt{fn}\, x\, E \mid \texttt{rec}\, \texttt{f}\, x\, E \mid E\, E$$

$$\begin{array}{rrcl} \text{변환} & \underline{\cdot} & \in & \mathbb{V}\ Exp \rightarrow (\mathbb{K} \rightarrow \mathbb{V})\ Exp \\ \text{마저할일} & \mathbb{K} & = & \mathbb{V} \rightarrow \mathbb{V} \end{array}$$

$$\begin{array}{rcl} \underline{n} & = & \texttt{fn}\, k\, (k\, n) \\ \underline{x} & = & \texttt{fn}\, k\, (k\, x) \\ \underline{\texttt{fn}\, x\, E} & = & \texttt{fn}\, k\, (k\, (\texttt{fn}\, x\, \underline{E})) \\ \underline{\texttt{rec}\, f\, x\, E} & = & \texttt{fn}\, k\, (k\, (\texttt{rec}\, f\, x\, \underline{E})) \\ \underline{E_1\, E_2} & = & \texttt{fn}\, k\, (\underline{E_1}\, (\texttt{fn}\, f\, (\underline{E_2}\, (\texttt{fn}\, v\, (f\, v\, k))))) \end{array}$$

CPS 변환의 올바름

- 무엇 의미구조denotational semantics:

$$[\![E]\!] = [\![\underline{E}]\!]\ id$$

- 실행 의미구조operational semantics:

$$E \xrightarrow{*} v \quad \text{iff} \quad \underline{E}\ \mathsf{id} \xrightarrow{*} v\_$$

$$\begin{array}{rcl} n\_ & = & n \\ x\_ & = & x \\ (\mathtt{fn}\ x\ E')\_ & = & \mathtt{fn}\ x\ \underline{E'} \\ (\mathtt{rec}\ f\ x\ E')\_ & = & \mathtt{rec}\ f\ x\ \underline{E'} \end{array}$$

- *Call-by-name, call-by-value, and λ-calculus*, Gordon Plotkin, Theoretical Computer Science 1, pp.125–159

CPS변환의 올바름: 겉보기 증명extentional proof

- CPS변환 `cpsExp`:

$$\texttt{cpsExp} : A\ Exp \ \rightarrow \ ((A \rightarrow A) \rightarrow A)\ Exp$$

- 대응하는 의미세계의 함수 cps:

$$cps : (A \rightarrow A) \rightarrow ((A \rightarrow A) \rightarrow A) \rightarrow ((A \rightarrow A) \rightarrow A)$$

다음 성질을 만족한다(가정):

$$\forall f, f', k.\ (cps\ f)\ f'\ k = f\ (f'\ k)$$

- 다음 겉모습을 증명하자: (의미세계에서 CPS변환의 올바름)

$$\forall F.\ lfp F = (lfp(cps\ F))\ id$$

증명목표:

$$lfp(F) = (lfp(cps\ F))\ id$$

고정점 인덕으로 증명: $P(lfp(F), lfp(cps\ F))$

$$P(f, g) \stackrel{\text{let}}{=} (f = g\ id)$$

$(f \in A, \quad g \in (A \to A) \to A)$

- $\bot_A = \bot_{(A\to A)\to A}(id)$? 네.
- $P(f, g)$ 이면 $P(F\ f, (cps\ F)\ g)$? 즉
  $f = g\ id$ 이면 $F\ f = (cps\ F)\ g\ id$? 네, 왜냐면

$$\begin{array}{rcll} (cps\ F)\ g\ id & = & F\ (g\ id) & (cps \text{ 성질}) \\ & = & F\ f. & (\text{인덕가정}) \end{array}$$

마저할일continuation 사용한 의미구조: 예외상황

$$
\begin{array}{rcl}
E & \to & n \mid x \mid \texttt{fn}\, x\, E \mid \texttt{rec f}\, x\, E \mid E\, E \\
 & \mid & \texttt{raise}\, L \mid E\, \texttt{handle}\, L\, E
\end{array}
$$

$$
\begin{array}{rcl}
[\![E]\!] & \in & Env \to \mathbb{K} \to \mathbb{H} \to \mathbb{V} \\
\text{환경} \quad \sigma \in Env & = & Var_\perp \to \mathbb{V} \\
\text{마저할일} \quad k \in \mathbb{K} & = & \mathbb{V} \to \mathbb{V} \\
\text{예외처리} \quad h \in \mathbb{H} & = & Exn_\perp \to \mathbb{V} \qquad \text{예외 } L \in Exn \\
\text{값} \quad v \in \mathbb{V} & = & \mathbb{N}_\perp + (\mathbb{V} \to \mathbb{K} \to \mathbb{H} \to \mathbb{V})
\end{array}
$$

$$
\begin{array}{rcl}
[\![\texttt{raise}\, L]\!]\, \sigma\, k\, h & = & h(L) \\
[\![E_1\, \texttt{handle}\, L\, E_2]\!]\, \sigma\, k\, h & = & [\![E_1]\!]\, \sigma\, k\, (h\{L \mapsto [\![E_2]\!]\, \sigma\, k\, h\}) \\
[\![n]\!]\, \sigma\, k\, h & = & k(n) \\
[\![x]\!]\, \sigma\, k\, h & = & k(\sigma(x)) \\
[\![E_1\, E_2]\!]\, \sigma\, k\, h & = & [\![E_1]\!]\, \sigma\, (\lambda f.[\![E_2]\!]\, \sigma\, (\lambda v.f\, v\, k\, h)\, h)\, h \\
\cdots & &
\end{array}
$$

예외상황 녹이기: 마저할일continuation 이용

$$
\begin{array}{rcl}
E & \to & n \mid x \mid \texttt{fn}\, x\, E \mid \texttt{rec f}\, x\, E \mid E\, E \mid E?\, E : E \\
 & \mid & \texttt{raise}\, L \mid E\, \texttt{handle}\, L\, E
\end{array}
$$

$$
\begin{array}{rrcl}
\text{변환} & \underline{\cdot} & \in & \mathbb{V}\ Exp \to (\mathbb{K} \to \mathbb{H} \to \mathbb{V})\ Exp \\
\text{마저할일} & k \in \mathbb{K} & = & \mathbb{V} \to \mathbb{V} \\
\text{예외처리} & h \in \mathbb{H} & = & Exn \to \mathbb{V} \qquad \text{예외}\ L \in Exn
\end{array}
$$

$$
\begin{array}{rcl}
\underline{\texttt{raise}\, L} & = & \texttt{fn}\, (k, h)\, (h\ L) \\
\underline{E_1\, \texttt{handle}\, L\, E_2} & = & \texttt{fn}\, (k, h)\, \underline{E_1}\, (k, \texttt{fn}\, x\, (x = L?\, (\underline{E_2}\, (k, h)) : h(x))) \\
\underline{n} & = & \texttt{fn}\, (k, h)\, (k\ n) \\
\underline{x} & = & \texttt{fn}\, (k, h)\, (k\ x) \\
\underline{\texttt{fn}\, x\, E} & = & \texttt{fn}\, (k, h)\, (k\, (\texttt{fn}\, x\, \underline{E})) \\
\underline{\texttt{rec}\, f\, x\, E} & = & \texttt{fn}\, (k, h)\, (k\, (\texttt{rec}\, f\, x\, \underline{E})) \\
\underline{E_1\ E_2} & = & \texttt{fn}\, (k, h)\, (\underline{E_1}\, (\texttt{fn}\, f\, (\underline{E_2}\, (\texttt{fn}\, v\, (f\ v\, (k, h)), h)), h)) \\
\cdots &&
\end{array}
$$

마저할일continuation을 값으로control as value

$$
\begin{array}{rcl}
E & \to & n \mid x \mid \texttt{fn}\, x\, E \mid \texttt{rec f}\, x\, E \mid E\, E \\
 & \mid & \texttt{catch}\, x\, E \mid \texttt{throw}\, x\, E
\end{array}
$$

$$
\begin{array}{rcl}
[\![E]\!] & \in & Env \to \mathbb{K} \to \mathbb{V} \\
\text{환경}\quad \sigma \in Env & = & Var_\perp \to \mathbb{V} \\
\text{마저할일}\quad k \in \mathbb{K} & = & \mathbb{V} \to \mathbb{V} \\
\text{값}\quad v \in \mathbb{V} & = & \mathbb{N}_\perp + (\mathbb{V} \to \mathbb{K} \to \mathbb{V}) + \mathbb{K}
\end{array}
$$

$$
\begin{array}{rcl}
[\![\texttt{catch}\, x\, E]\!]\, \sigma\, k & = & [\![E]\!]\, \sigma\{x \mapsto k\}\, k \\
[\![\texttt{throw}\, x\, E]\!]\, \sigma\, k & = & [\![E]\!]\, \sigma\, (\lambda v.\sigma(x)(v)) \\
[\![\texttt{fn}\, x\, E]\!]\, \sigma\, k & = & k(\lambda v.\lambda k'.[\![E]\!]\, \sigma\{x \mapsto v\}\, k') \\
[\![E_1\, E_2]\!]\, \sigma\, k & = & [\![E_1]\!]\, \sigma\, (\lambda f.[\![E_2]\!]\, \sigma\, (\lambda v.f\, v\, k)) \\
\ldots & &
\end{array}
$$

마저할일continuation을 값으로control as value

$$E \;\to\; n \mid x \mid \texttt{fn}\, x\, E \mid \texttt{rec f}\, x\, E \mid E\, E \mid \texttt{catch}\, x\, E \mid \texttt{throw}\, x\, E$$

$$\begin{aligned}
[\![E]\!] &\in Env \to \mathbb{K} \to \mathbb{V} \\
\text{환경}\quad \sigma \in Env &= Var_\perp \to \mathbb{V} \\
\text{마저할일}\quad k \in \mathbb{K} &= \mathbb{V} \to \mathbb{V} \\
\text{값}\quad v \in \mathbb{V} &= \mathbb{N}_\perp + (\mathbb{V} \to \mathbb{K} \to \mathbb{V}) + \mathbb{K}
\end{aligned}$$

$$\begin{aligned}
[\![\texttt{catch}\, x\, E]\!]\, \sigma\, k &= [\![E]\!]\, \sigma\{x \mapsto k\}\, k \\
[\![\texttt{throw}\, x\, E]\!]\, \sigma\, k &= [\![E]\!]\, \sigma\, (\lambda v.\sigma(x)(v)) \\
[\![\texttt{fn}\, x\, E]\!]\, \sigma\, k &= k(\lambda v.\lambda k'.[\![E]\!]\, \sigma\{x \mapsto v\}\, k') \\
[\![E_1\, E_2]\!]\, \sigma\, k &= [\![E_1]\!]\, \sigma\, (\lambda f.[\![E_2]\!]\, \sigma\, (\lambda v.f\, v\, k)) \\
&\cdots
\end{aligned}$$

(C/C++)에서: $\sim \texttt{setjmp}\, x\, E \mid \texttt{longjmp}\, x\, E$

Scheme(Haskell/Scala등)에서: $\sim \texttt{callcc}\, (\texttt{fn}\, x\, E)$

프로그래밍언어가 제공하는 값의 종류

프로래머가 맘대로 다루는as first-class object

- 숫자, 참거짓, 문자열, ...
- 함수, 짝, 메모리주소, 실행순서control, 프로램식, ...

값 종류마다, 만들기 $\rightleftharpoons$ 사용하기 방법을 제공

$$
\begin{array}{rrcl}
\text{함수:} & \texttt{fn}\, x\, E \mid \texttt{rec}\, f\, x\, E & \rightleftharpoons & E\, E \\
\text{짝:} & (E, E) & \rightleftharpoons & E.\texttt{l} \mid E.\texttt{r} \\
\text{메모리주소:} & \texttt{ref}\, E & \rightleftharpoons & E := E \mid\, !E \\
\text{실행순서:} & \texttt{catch}\, x\, E & \rightleftharpoons & \texttt{throw}\, x\, E \\
\text{프로램식:} & \texttt{box}\, E & \rightleftharpoons & \texttt{unbox}\, E \mid \texttt{run}\, E
\end{array}
$$