14 章 Exceptions

cocoa

May 26, 2001

Exercise 14.3.2

ここで紹介する型システムは回答例として挙げられている Leroy の論文[1]を大幅に簡約化したものである。ただし exception の扱いについてはできるだけオリジナルの性質を保持するように努力したつもりである。

$t ::= x$	variable
$\mid \lambda x : T.t$	abstraction
$\mid tt$	application
$\mid \langle l = t \rangle$	tagging
$\mid$ case t of $\langle l = x \rangle \to t \mid x \to t$	case
$\mid$ raise t	raise exception
$\mid$ try t with $x \to t$	exception handler

$T ::= V$	variant type
$\mid T \xrightarrow{V} T$	function type
$V ::= \langle l_i : T_i, \cdots \rangle^{i \in 1..n}$	extensible variant type

Effect system や region analysis になれていないと function type を見て驚くかもしれないが、本質的には Java の型システムと同じである。Java でメソッドを "`int foo( double ) throws exn`" と書く代わりに double $\xrightarrow{\text{exn}}$ int と書いているだけである。

この型システムのポイントは二つある。

- 投げられる可能性のある exception を関数型に付加することで型チェックを容易にしている。
- exception を extensible variant にすることで、ユーザが後から定義した exception を扱えるようになる。

後者について説明するともし extensible でなかったら、ある関数が投げることのできる exception はあらかじめ定義されたものだけになる。そうでなかったら uncaught exception を型システムによって防ぐことができなくなることになってしまう。Java では subtyping を利用してこれを避けているが、ML のような言語では extensible variant を利用している。

[1] Exercise の回答として "See [Leroy 2000]." とだけ書かれている。

Type judgment は $\Gamma \vdash t : T/V$ の形で書かれる。最後の V は t を評価した時に投げられるかもしれない exception の集合 (を variant type として書いたもの) を表している。

$$\frac{x : T \in \Gamma}{\Gamma \vdash x : T/V}$$

$$\frac{\Gamma, x : T \vdash t : T'/V'}{\Gamma \vdash \lambda x : T.t : T \xrightarrow{V'} T'/V}$$

$$\frac{\Gamma \vdash t : T' \xrightarrow{V} T/V \quad \Gamma \vdash t' : T'/V}{\Gamma \vdash tt' : T/V}$$

$$\frac{l = l_i \quad \Gamma \vdash t : T_i/V}{\Gamma \vdash \langle l = t \rangle : \langle l_i : T_i, \cdots \rangle^{i \in 1..n}/V}$$

$$\frac{\Gamma \vdash t : T'/V \quad \vdash T' - l = T''/T_x \quad \Gamma, x : T_x \vdash t' : T/V \quad \Gamma, x : T'' \vdash t'' : T/V}{\Gamma \vdash \text{case } t \text{ of } \langle l = x \rangle \to t' \mid x' \to t'' : T/V}$$

$$\frac{\Gamma \vdash t : V/V}{\Gamma \vdash \text{raise } t : T/V}$$

$$\frac{\Gamma \vdash t : T/V' \quad \Gamma, x : V' \vdash t' : T/V}{\Gamma \vdash \text{try } t \text{ with } x \to t' : T/V}$$

$$\frac{l_i \in \{l_1, ..., l_n\}}{\vdash \langle l_1 : T_1, ..., l_n : T_n, \cdots \rangle - l_i = \langle l_1 : T_1, ..., l_{i-1} : T_{i-1}, l_{i+1} : T_{i+1}, ..., l_n : T_n, \cdots \rangle / T_i}$$

proof of type safety はパス。

Exercise 15.2.2

Subtyping を含む型システムにおいては type derivation が unique に決まらない場合がある。

$$\frac{\dfrac{}{\vdash (\lambda r : \{x : \text{Nat}\}.r.x) : \{x : \text{Nat}\} \to \text{Nat}} \qquad \dfrac{\dfrac{}{\vdash \{x = 0, y = 1\} : \{x : \text{Nat}, y : \text{Nat}\}} \quad \{x : \text{Nat}, y : \text{Nat}\} <: \{x : \text{Nat}\}}{\vdash \{x = 0, y = 1\} : \{x : \text{Nat}\}}}{\vdash (\lambda r : \{x : \text{Nat}\}.r.x)\{x = 0, y = 1\} : \text{Nat}}$$

$$\frac{\dfrac{}{\vdash (\lambda r : \{x : \text{Nat}\}.r.x) : \{x : \text{Nat}\} \to \text{Nat}} \quad \{x : \text{Nat}\} \to \text{Nat} <: \{x : \text{Nat}, y : \text{Nat}\} \to \text{Nat}}{\vdash (\lambda r : \{x : \text{Nat}\}.r.x) : \{x : \text{Nat}, y : \text{Nat}\} \to \text{Nat}} \qquad \dfrac{}{\vdash \{x = 0, y = 1\} : \{x : \text{Nat}, y : \text{Nat}\}}}{\vdash (\lambda r : \{x : \text{Nat}\}.r.x)\{x = 0, y = 1\} : \text{Nat}}$$