

# Operational Semantics of Exception Handling

## 1 Large-Step

$$\frac{\langle e, \sigma \rangle \Downarrow n}{\langle \text{throw } e, \sigma \rangle \Downarrow \sigma \text{ exc } n} \text{ throw}$$

$$\frac{\langle c_1, \sigma \rangle \Downarrow \sigma'}{\langle \text{try } c_1 \text{ catch } x \text{ } c_2, \sigma \rangle \Downarrow \sigma'} \text{ catch1}$$

$$\frac{\langle c_1, \sigma \rangle \Downarrow \sigma' \text{ exc } n \quad \langle x := n; c_2, \sigma' \rangle \Downarrow t}{\langle \text{try } c_1 \text{ catch } x \text{ } c_2, \sigma \rangle \Downarrow t} \text{ catch2}$$

$$\frac{\langle c_1, \sigma \rangle \Downarrow \sigma' \quad \langle c_2, \sigma' \rangle \Downarrow t}{\langle \text{after } c_1 \text{ finally } c_2, \sigma \rangle \Downarrow t} \text{ finally1}$$

$$\frac{\langle c_1, \sigma \rangle \Downarrow \sigma' \text{ exc } n \quad \langle c_2, \sigma' \rangle \Downarrow \sigma''}{\langle \text{after } c_1 \text{ finally } c_2, \sigma \rangle \Downarrow \sigma'' \text{ exc } n} \text{ finally2}$$

$$\frac{\langle c_1, \sigma \rangle \Downarrow \sigma' \text{ exc } n_1 \quad \langle c_2, \sigma' \rangle \Downarrow \sigma'' \text{ exc } n_2}{\langle \text{after } c_1 \text{ finally } c_2, \sigma \rangle \Downarrow \sigma'' \text{ exc } n_2} \text{ finally3}$$

$$\frac{\langle c_1, \sigma \rangle \Downarrow \sigma' \quad \langle c_2, \sigma' \rangle \Downarrow t}{\langle c_1; c_2, \sigma \rangle \Downarrow t} \text{ seq1}$$

$$\frac{\langle c_1, \sigma \rangle \Downarrow \sigma' \text{ exc } n}{\langle c_1; c_2, \sigma \rangle \Downarrow \sigma' \text{ exc } n} \text{ seq2}$$

## 2 Small-Step Contextual

$$\begin{array}{l} \text{Context } H ::= \dots \\ | \text{ throw } H \\ | \text{ try } H \text{ catch } x \text{ } c \\ | \text{ after } H \text{ finally } c \end{array}$$

$$\begin{array}{l} \text{Redex } r ::= \dots \\ | \text{ throw } n; c \\ | \text{ try skip catch } x \text{ } c \\ | \text{ try throw } n \text{ catch } x \text{ } c \\ | \text{ after skip finally } c \\ | \text{ after throw } n \text{ finally } c \end{array}$$

Local Reduction Rules:

$$\langle \text{throw } n; c, \sigma \rangle \rightarrow \langle \text{throw } n, \sigma \rangle$$

$$\langle \text{try skip catch } x \text{ } c, \sigma \rangle \rightarrow \langle \text{skip}, \sigma \rangle$$

$$\langle \text{try throw } n \text{ catch } x \text{ } c, \sigma \rangle \rightarrow \langle x := n; c, \sigma \rangle$$

$$\langle \text{after skip finally } c, \sigma \rangle \rightarrow \langle c, \sigma \rangle$$

$$\langle \text{after throw } n \text{ finally } c, \sigma \rangle \rightarrow \langle c; \text{throw } n, \sigma \rangle$$

Note that `skip` and `throw n` are both terminal programs.