Translation Sketch

1 Expressions

E translates a Chalice expression into an equivalent SIL expression. Given an scoped identifier i, $\rho(i)$ denotes a globally unique identifier. E.g., $\rho(\texttt{someField}) = \texttt{SomeClass::someField}$

$$E\,\llbracket e_1 \,\texttt{?}\, e_2 \,\texttt{:}\, e_3 \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{if } E\,(e_1) \texttt{ then } E\,(e_2) \texttt{ else } E\,(e_3) \rrbracket_{\mathrm{SIL}} \tag{1}$$

$$E\,\llbracket e_1 \texttt{==} e_2 \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{==}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2)) \rrbracket_{\mathrm{SIL}} \tag{2}$$

$$E\,\llbracket e_1 \texttt{!=} e_2 \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{!=}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2)) \rrbracket_{\mathrm{SIL}} \tag{3}$$

$$E\,\llbracket e_1 \texttt{<} e_2 \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{<}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2)) \rrbracket_{\mathrm{SIL}} \tag{4}$$

$$E\,\llbracket e_1 \texttt{<=} e_2 \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{<=}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2)) \rrbracket_{\mathrm{SIL}} \tag{5}$$

$$E\,\llbracket e_1 \texttt{>=} e_2 \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{>=}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2)) \rrbracket_{\mathrm{SIL}} \tag{6}$$

$$E\,\llbracket e_1 \texttt{>} e_2 \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{>}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2)) \rrbracket_{\mathrm{SIL}} \tag{7}$$

$$E\,\llbracket e_1 \texttt{ in } e_2 \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{in}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2)) \rrbracket_{\mathrm{SIL}} \tag{8}$$

$$E\,\llbracket e_1 \texttt{ !in } e_2 \rrbracket_{\mathrm{Ch}} = E\,\llbracket \texttt{!}\,(e_1 \texttt{ in } e_2) \rrbracket_{\mathrm{Ch}} \tag{9}$$

$$E\,\llbracket e_1 \texttt{+} e_2 \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{+}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2)) \rrbracket_{\mathrm{SIL}} \tag{10}$$

$$E\,\llbracket e_1 \texttt{-} e_2 \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{-}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2)) \rrbracket_{\mathrm{SIL}} \tag{11}$$

$$E\,\llbracket e_1 \texttt{*} e_2 \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{*}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2)) \rrbracket_{\mathrm{SIL}} \tag{12}$$

$$E\,\llbracket e_1 \texttt{/} e_2 \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{/}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2)) \rrbracket_{\mathrm{SIL}} \tag{13}$$

$$E\,\llbracket e_1 \texttt{\%} e_2 \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{\%}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2)) \rrbracket_{\mathrm{SIL}} \tag{14}$$

$$E\,\llbracket \texttt{!}\, e \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{!}\,(E\,(e)) \rrbracket_{\mathrm{SIL}} \tag{15}$$

$$E\,\llbracket \texttt{-}\, e \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{-}\,(E\,(e)) \rrbracket_{\mathrm{SIL}} \tag{16}$$

$$E\,\llbracket e_1 \texttt{[}\, e_2 \,\texttt{]} \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{at}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2)) \rrbracket_{\mathrm{SIL}} \tag{17}$$

$$E\,\llbracket e_1 \texttt{[}\, e_2 \texttt{..]} \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{drop}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2)) \rrbracket_{\mathrm{SIL}} \tag{18}$$

$$E\,\llbracket e_1 \texttt{[..}\, e_2 \,\texttt{]} \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{take}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2)) \rrbracket_{\mathrm{SIL}} \tag{19}$$

$$E\,\llbracket e_1 \texttt{.} id \rrbracket_{\mathrm{Ch}} = \llbracket E\,(e_1)\,\texttt{.}\,id \rrbracket_{\mathrm{SIL}} \quad \text{Also assert that } e_1 \text{ is not null} \tag{20}$$

$$E\,\llbracket e_1 \texttt{.} id\,(e \ldots) \rrbracket_{\mathrm{Ch}} = \llbracket \rho(id)(E\,(e_1)\,\texttt{,}\,E\,(e \ldots)) \rrbracket_{\mathrm{SIL}} \quad \text{Also assert that } e_1 \text{ is not null} \tag{21}$$

$$E\,\llbracket \texttt{true} \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{true} \rrbracket_{\mathrm{SIL}} \tag{22}$$

$$E\,\llbracket \texttt{false} \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{false} \rrbracket_{\mathrm{SIL}} \tag{23}$$

$$E\,\llbracket \texttt{null} \rrbracket_{\mathrm{Ch}} = \llbracket \texttt{null} \rrbracket_{\mathrm{SIL}} \tag{24}$$

$$E\,\llbracket \texttt{this} \rrbracket_{\mathrm{Ch}} = \text{First method argument} \tag{25}$$

$$E\,\llbracket x \rrbracket_{\mathrm{Ch}} = \llbracket x \rrbracket_{\mathrm{SIL}} \qquad \text{where } x \text{ is a numeric literal} \tag{26}$$

2 Statements

S translates Chalice statements into equivalent SIL statements/expressions. $\tau_1, \tau_2, \ldots$ are temporary variables unique to each rule instantiation. Similarly, $\eta_1, \eta_2, \ldots$ are labels unique to each rule instantiation.

$$S\,[\![\texttt{assert } e_1]\!]_{\text{Ch}} = [\![\texttt{assert } E\,(e_1)]\!]_{\text{SIL}} \tag{27}$$
$$S\,[\![\texttt{assume } e_1]\!]_{\text{Ch}} = [\![\texttt{assume } E\,(e_1)]\!]_{\text{SIL}} \tag{28}$$
$$S\,[\![\{s \ldots\}]\!]_{\text{Ch}} = [\![S\,(s \ldots)]\!]_{\text{SIL}} \tag{29}$$

Need to flatten nested stmt blocks, since SIL doesn't have local variable scoping.

$$S\,[\![\texttt{ var } id \texttt{ := } e_1]\!]_{\text{Ch}} = [\![id \texttt{ := } E\,(e_1)]\!]_{\text{SIL}} \quad \texttt{Keep const and ghost attributes} \tag{30}$$

$$S\,[\![\texttt{ call } v \ldots \texttt{ := } r\texttt{.}m_1(e \ldots)]\!]_{\text{Ch}} = [\![\texttt{call } v \ldots \texttt{ := } \rho\,(m_1)\,(E\,(r)\,\texttt{,}E\,(e \ldots))]\!]_{\text{SIL}} \tag{31}$$

$$\begin{aligned} S\,[\![\texttt{if}\,(e_1)\,s_1 \texttt{ else } s_2]\!]_{\text{Ch}} = [\![&\texttt{if}\,E\,(e_1) \texttt{ then goto } \eta_1 \texttt{ else } \eta_2 \texttt{ ;} \\ &\eta_1 \texttt{:}\, S\,(s_1) \texttt{ ;} \\ &\quad \texttt{goto } \eta_3 \\ &\eta_2 \texttt{:}\, S\,(s_2) \\ &\eta_3 \texttt{: nop }]\!]_{\text{SIL}} \end{aligned} \tag{32}$$

`lockchange` for `while` loops ignored for now.

$$\begin{aligned} S\,[\![\texttt{while}\,(c) \texttt{ invariant } i \ldots \texttt{ ; } s]\!]_{\text{Ch}} = [\![\; &\eta_1 \texttt{:}\, \tau_1 \texttt{ := } E\,(c) \texttt{ ;} \\ &\quad \texttt{exhale } E\,(i \ldots) \texttt{ ;} \\ &\quad \texttt{if } \tau_1 \texttt{ then goto } \eta_2 \texttt{ else } \eta_3 \texttt{ ;} \\ &\eta_2 \texttt{: inhale}\, E\,(i \ldots) \texttt{ ;} \\ &\quad S\,(s) \texttt{ ;} \\ &\quad \texttt{goto } \eta_1 \texttt{ ;} \\ &\eta_3 \texttt{: inhale}\, E\,(i \ldots)]\!]_{\text{SIL}} \end{aligned} \tag{33}$$

2.1 Predicates

$\sigma\,(o\texttt{.}p)$ looks up the (permission) expression associated with a predicate p on object o.

$$\begin{aligned} S\,\big[\!\!\big[\texttt{fold } e_1\texttt{.}p_1\big]\!\!\big]_{\text{Ch}} = [\![\; &\texttt{exhale } \sigma\big(E\,(e_1)\texttt{.}p_1\big) \texttt{ ;} \\ &\texttt{inhale acc}\big(E\,(e_1)\texttt{.}p_1\big)]\!]_{\text{SIL}} \end{aligned} \tag{34}$$
$$\begin{aligned} S\,\big[\!\!\big[\texttt{unfold } e_1\texttt{.}p_1\big]\!\!\big]_{\text{Ch}} = [\![\; &\texttt{exhale acc } \big(E\,(e_1)\texttt{.}p_1\big) \texttt{ ;} \\ &\texttt{inhale}\,\sigma\big(E\,(e_1)\texttt{.}p_1\big)]\!]_{\text{SIL}} \end{aligned} \tag{35}$$

2.2 Monitors

$\iota\,(e)$ denotes the (monitor) invariant of an object e.

$$S\,[\![\texttt{share } e_1]\!]_{\text{Ch}} = [\![\texttt{exhale}\,\iota\,(E\,(e_1))]\!]_{\text{SIL}} \tag{36}$$
$$S\,[\![\texttt{unshare } e_1]\!]_{\text{Ch}} = [\![\texttt{exhale}\,\iota\,(E\,(e_1))]\!]_{\text{SIL}} \tag{37}$$