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., $\rho(\texttt{someField}) = \texttt{SomeClass::someField}$

$$
\begin{aligned}
E\,[\![e_1\,\texttt{?}\,e_2\,\texttt{:}\,e_3]\!]_{\text{Ch}} &= [\![E\,(e_1)\ \texttt{?}\ E\,(e_2)\ \texttt{:}\ E\,(e_3)]\!]_{\text{SIL}} && (1)\\
E\,[\![e_1\texttt{==}\,e_2]\!]_{\text{Ch}} &= [\![\texttt{==}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2))]\!]_{\text{SIL}} && (2)\\
E\,[\![e_1\,\texttt{!=}\,e_2]\!]_{\text{Ch}} &= [\![\texttt{!=}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2))]\!]_{\text{SIL}} && (3)\\
E\,[\![e_1\texttt{<}\,e_2]\!]_{\text{Ch}} &= [\![\texttt{<}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2))]\!]_{\text{SIL}} && (4)\\
E\,[\![e_1\texttt{<=}\,e_2]\!]_{\text{Ch}} &= [\![\texttt{<=}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2))]\!]_{\text{SIL}} && (5)\\
E\,[\![e_1\texttt{>=}\,e_2]\!]_{\text{Ch}} &= [\![\texttt{>=}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2))]\!]_{\text{SIL}} && (6)\\
E\,[\![e_1\texttt{>}\,e_2]\!]_{\text{Ch}} &= [\![\texttt{>}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2))]\!]_{\text{SIL}} && (7)\\
E\,[\![e_1\text{«}\,e_2]\!]_{\text{Ch}} &= \texttt{/* lock below ??? */} && (8)\\
E\,[\![e_1\ \texttt{in}\ e_2]\!]_{\text{Ch}} &= [\![\texttt{in}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2))]\!]_{\text{SIL}} && (9)\\
E\,[\![e_1\ \texttt{!in}\ e_2]\!]_{\text{Ch}} &= E\,[\![\texttt{!}\,(e_1\ \texttt{in}\ e_2)]\!]_{\text{Ch}} && (10)\\
E\,[\![e_1\texttt{+}\,e_2]\!]_{\text{Ch}} &= [\![\texttt{+}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2))]\!]_{\text{SIL}} && (11)\\
E\,[\![e_1\texttt{-}\,e_2]\!]_{\text{Ch}} &= [\![\texttt{-}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2))]\!]_{\text{SIL}} && (12)\\
E\,[\![e_1\texttt{*}\,e_2]\!]_{\text{Ch}} &= [\![\texttt{*}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2))]\!]_{\text{SIL}} && (13)\\
E\,[\![e_1\texttt{/}\,e_2]\!]_{\text{Ch}} &= [\![\texttt{/}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2))]\!]_{\text{SIL}} && (14)\\
E\,[\![e_1\texttt{\%}\,e_2]\!]_{\text{Ch}} &= [\![\texttt{\%}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2))]\!]_{\text{SIL}} && (15)\\
E\,[\![\texttt{!}\ e]\!]_{\text{Ch}} &= [\![\texttt{!}\ (E\,(e))]\!]_{\text{SIL}} && (16)\\
E\,[\![\texttt{-}\ e]\!]_{\text{Ch}} &= [\![\texttt{-}\ (E\,(e))]\!]_{\text{SIL}} && (17)\\
E\,[\![e_1\,\texttt{[}\ e_2\,\texttt{]}\,]\!]_{\text{Ch}} &= [\![\texttt{at}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2))]\!]_{\text{SIL}} && (18)\\
E\,[\![e_1\,\texttt{[}\ e_2\,\texttt{..]}\,]\!]_{\text{Ch}} &= [\![\texttt{drop}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2))]\!]_{\text{SIL}} && (19)\\
E\,[\![e_1\,\texttt{[..}\ e_2\,\texttt{]}\,]\!]_{\text{Ch}} &= [\![\texttt{take}\,(E\,(e_1)\,\texttt{,}\,E\,(e_2))]\!]_{\text{SIL}} && (20)\\
E\,[\![e_1\,\texttt{.}\,id]\!]_{\text{Ch}} &= [\![E\,(e_1)\ \texttt{.}\,id]\!]_{\text{SIL}} && (21)\\
E\,[\![e_1\,\texttt{.}\,id(e\ \ldots\,)]\!]_{\text{Ch}} &= [\![\rho(id)(E\,(e_1)\,\texttt{,}\,E\,(e\ \ldots))]\!]_{\text{SIL}} && (22)\\
E\,[\![\texttt{true}]\!]_{\text{Ch}} &= [\![\texttt{true}]\!]_{\text{SIL}} && (23)\\
E\,[\![\texttt{false}]\!]_{\text{Ch}} &= [\![\texttt{false}]\!]_{\text{SIL}} && (24)\\
E\,[\![\texttt{null}]\!]_{\text{Ch}} &= [\![\texttt{null}]\!]_{\text{SIL}} && (25)\\
E\,[\![\texttt{this}]\!]_{\text{Ch}} &= \texttt{/* ??? */} && (26)\\
E\,[\![x]\!]_{\text{Ch}} &= [\![x]\!]_{\text{SIL}} \qquad \text{where } x \text{ is a numeric literal} && (27)
\end{aligned}
$$

2 Statements

S translates Chalice statements into equivalent SIL statements/expressions. $\tau_1, \tau_2, \dots$ are temporary variables unique to each rule instantiation.

$$S\,[\![\texttt{assert } e_1]\!]_{\text{Ch}} = [\![\texttt{assert } E\,(e_1)]\!]_{\text{SIL}} \tag{28}$$
$$S\,[\![\texttt{assume } e_1]\!]_{\text{Ch}} = [\![\texttt{assume } E\,(e_1)]\!]_{\text{SIL}} \tag{29}$$
$$S\,[\![\{s \dots\}]\!]_{\text{Ch}} = [\![S\,(s \dots)]\!]_{\text{SIL}} \tag{30}$$

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

$$S\,[\![\ \texttt{spec } id\,[e \dots]]\!]_{\text{Ch}} = \text{What does } \texttt{spec} \text{ do?} \tag{31}$$
$$S\,[\![\ \texttt{var } id\ \texttt{:=}\ e_1]\!]_{\text{Ch}} = [\![id\ \texttt{:=}\ E\,(e_1)]\!]_{\text{SIL}} \tag{32}$$
$$S\,[\![\ \texttt{const } id\ \texttt{:=}\ e_1]\!]_{\text{Ch}} = [\![id\ \texttt{:=}\ E\,(e_1)]\!]_{\text{SIL}} \tag{33}$$
$$S\,[\![\ \texttt{ghost var } id\ \texttt{:=}\ e_1]\!]_{\text{Ch}} = [\![id\ \texttt{:=}\ E\,(e_1)]\!]_{\text{SIL}} \quad \text{Keep track of } \texttt{ghost}\text{?} \tag{34}$$
$$S\,[\![\ \texttt{ghost const } id\ \texttt{:=}\ e_1]\!]_{\text{Ch}} = [\![id\ \texttt{:=}\ E\,(e_1)]\!]_{\text{SIL}} \tag{35}$$
$$S\,[\![\ \texttt{call } v \dots\ \texttt{:=}\ r\texttt{.}m_1(e \dots)]\!]_{\text{Ch}} = [\![\texttt{call } v \dots\ \texttt{:=}\ E\,(r)\ \texttt{.}m_1\,(E\,(e \dots))]\!]$$

If SIL doesn't support multiple return values, the method will have to return a tuple containing the actual return values.

$$S\,[\![\ \texttt{call } v_1, v_2, \dots, v_n\ \texttt{:=}\ r\texttt{.}m_1(e \dots)]\!]_{\text{Ch}} = [\![\texttt{call } \tau_1\ \texttt{:=}\ E\,(r)\ \texttt{.}m_1(E\,(e \dots))\ \texttt{;} \tag{36}$$
$$S\,(v_1\ \texttt{:=}\ \tau_1\texttt{.}elem_1)\ \texttt{;} \tag{37}$$
$$S\,(v_1\ \texttt{:=}\ \tau_1\texttt{.}elem_1)\ \texttt{;} \tag{38}$$
$$\vdots \tag{39}$$
$$S\,(v_n\ \texttt{:=}\ \tau_1\texttt{.}elem_n)\ \texttt{;} \tag{40}$$
$$]\!]_{\text{SIL}}$$