Syntax

Literals	lit	$::= None \mid True \mid False \mid$ "a" $\mid \ldots \mid 1 \mid \ldots$
Expression	Exp	$::= lit@(\overline{A}) \mid Exp.id \mid f(\overline{Exp}) \mid Exp.f(\overline{Exp}) \mid C[\overline{A}](\overline{Exp})$
Typed Expression	$TExp$	$::= lit@(\overline{A}) : \tau \mid \ldots$
Assign Op.	$AsgOp$	$\in \{$`=, +=, -=, *=, /=, %=, //=`$\}$
Binary Op.	$BinOp$	$\in \{$`||, &&, |, &, ==, !=, <, >, <=, >=, +, -, *, /, %, **`$\}$
Statement	Stm	$::=$ **pass** $\mid$ **return** $Exp \mid Exp\,;Stm \mid id = Exp\,;Stm$
		$\mid Exp_1\ AsgOp\ Exp_2\,;Stm \mid$ **if** $Exp : Stm_1\,;$**else** $: Stm_2\,;Stm$
		$\mid$ **try** $: Stm$ **except** $: Exp\,;Stm \mid$ **raise** Exp

Projection To Python

$$(Exp)\quad (\!|lit@(\overline{B}) : \tau|\!)^A = \begin{cases} lit & \text{if } A \in \overline{B} \\ \texttt{Unit.id} & \text{otherwise} \end{cases}$$

$$(\!|Exp.id : \tau|\!)^A = \begin{cases} (\!|Exp|\!)^A.id & \text{if } A \in \text{rolesOf}(Exp.id) \\ \text{absent} & \text{otherwise} \end{cases}$$

$$(\!|f(\overline{Exp}) : \tau|\!)^A = \begin{cases} f(\overline{(\!|Exp|\!)^A}) & \text{if } A \in \text{rolesOf}(f(\overline{Exp})) \\ \texttt{Unit.id}(f(\overline{(\!|Exp|\!)^A})) & \text{if } A \in \text{rolesOf}(\overline{Exp}) \wedge A \notin \text{rolesOf}(f(\overline{Exp})) \\ \texttt{Unit.id}(\overline{(\!|Exp|\!)^A}) & \text{otherwise} \end{cases}$$

$$(\!|Exp.f(\overline{Exp}) : \tau|\!)^A = \begin{cases} (\!|Exp|\!)^A.f(\overline{(\!|Exp|\!)^A}) \\ \qquad \text{if } A \in \text{rolesOf}(Exp) \wedge A \in \text{rolesOf}(\overline{Exp}) \\ \qquad\qquad \wedge A \in \text{rolesOf}(Exp.f(\overline{Exp})) \\ \texttt{Unit.id}((\!|Exp|\!)^A.f(\overline{(\!|Exp|\!)^A})) \\ \qquad \text{if } A \in \text{rolesOf}(Exp) \wedge A \notin \text{rolesOf}(Exp.f(\overline{Exp})) \\ \texttt{Unit.id}((\!|Exp|\!)^A, \overline{(\!|Exp|\!)^A}) \qquad \text{otherwise} \end{cases}$$

$$(\!|C[\overline{B}](\overline{Exp}) : \tau|\!)^A = \begin{cases} (\!|C[\overline{B}]|\!)^A(\overline{(\!|Exp|\!)^A}) & A \in \overline{B} \\ \texttt{Unit.id}(\overline{(\!|Exp|\!)^A}) & \text{otherwise} \end{cases}$$

$$\text{rolesOf}(\_ : \tau@(\overline{B})) = \overline{B}$$

$$\text{rolesOf}(\overline{Exp}) = \bigcup_i \text{rolesOf}(Exp_i)$$

$$(\!|\overline{Exp}|\!)^A = Exp'_1, Exp'_2, \cdots, Exp'_n \quad \text{where} \quad Exp'_i = (\!|Exp_i|\!)^A$$

(Stm) $(\!|\textsf{pass}|\!)^A = \textsf{pass}$

$(\!|\textsf{return } Exp;|\!)^A = \textsf{return } (\!|Exp|\!)^A$

$$(\!|Exp; Stm|\!)^A = \begin{cases} \textsf{match } (\!|Exp|\!)^A : & \text{if } Exp = Exp.f(\overline{Exp}) : \texttt{Enum@}A \text{ and} \\ \quad \textsf{case } id : (\!|Stm|\!)^A; & \quad \text{Name}(f) = \texttt{Selection} \\ \quad \textsf{case } \_ : \text{assert False}; & \\ (\!|Exp|\!)^A; (\!|Stm|\!)^A & \text{otherwise} \end{cases}$$

$$(\!|id : TE = Exp\ ; Stm|\!)^A = \begin{cases} id = (\!|Exp|\!)^A; (\!|Stm|\!)^A & \text{if } A \in \text{rolesOf}(TE) \\ (\!|Exp|\!)^A; (\!|Stm|\!)^A & \text{otherwise} \end{cases}$$

$(\!|Exp_1\ AsgOp\ Exp_2\ ; Stm|\!)^A = (\!|Exp_1|\!)^A\, AsgOp\ (\!|Exp_2|\!)^A\ ; (\!|Stm|\!)^A$

$(\!|\textsf{if } Exp : Stm_1\ ; \textsf{else} : Stm_2\ ; Stm|\!)^A =$

$$\begin{cases} \textsf{if } (\!|Exp|\!)^A : (\!|Stm_1|\!)^A\ ; \textsf{else} : (\!|Stm_2|\!)^A\ ; (\!|Stm|\!)^A & \text{if rolesOf}(Exp) = A \\ (\!|Exp|\!)^A\ ; [\![(\!|Stm_1|\!)^A]\!] \sqcup [\![(\!|Stm_2|\!)^A]\!]\ ; (\!|Stm|\!)^A & \text{otherwise} \end{cases}$$

$(\!|\textsf{try} : Stm\ ;\ \textsf{except}\ Exp\ : Stm\ ; Stm|\!)^A =$

$$\{\textsf{try} : (\!|Stm|\!)^A\ ;\ \overline{\textsf{except } (\!|Exp|\!)^A : (\!|Stm|\!)^A}\ ; (\!|Stm|\!)^A \quad \text{if } A \in \text{rolesOf}(Exp)$$

Merging

Statement

$\textsf{return } Exp \sqcup \textsf{return } Exp' = \textsf{return } Exp \sqcup Exp'$

$(Exp_1\ AsgOp\ Exp_2; Stm) \sqcup (Exp'_1\ AsgOp\ Exp'_2; Stm')$
$= (Exp_1 \sqcup Exp'_1)\ AsgOp\ (Exp_2 \sqcup Exp'_2); (Stm \sqcup Stm')$

$(Exp; Stm) \sqcup (Exp'; Stm') = (Exp \sqcup Exp'); (Stm \sqcup Stm')$

$(\textsf{if } Exp : Stm_1\ ; \textsf{else} : Stm_2\ ; Stm) \sqcup (\textsf{if } Exp' : Stm'_1\ ; \textsf{else} : Stm'_2\ ; Stm')$
$= \textsf{if } (Exp \sqcup Exp') : (Stm_1 \sqcup Stm'_1)\ ; \textsf{else} : (Stm_2 \sqcup Stm'_2)\ ; (Stm \sqcup Stm')$

$\textsf{match } Exp :$		$\textsf{match } Exp' :$		$\textsf{match } Exp \sqcup Exp' :$
$\quad \textsf{case } id_a : Stm'_a;$		$\quad \textsf{case } id_a : Stm''_a;$		$\quad \textsf{case } id_a : Stm'_a \sqcup Stm''_a;$
$\quad \ldots$		$\quad \ldots$		$\quad \ldots$
$\quad \textsf{case } id_x : Stm'_x;$	$\sqcup$	$\quad \textsf{case } id_x : Stm''_x;$	$=$	$\quad \textsf{case } id_x : Stm'_x \sqcup Stm''_x;$
$\quad \textsf{case } id_y : Stm'_y;$				$\quad \textsf{case } id_y : Stm'_y;$
		$\quad \textsf{case } id_z : Stm'_z;$		$\quad \textsf{case } id_z : Stm'_z;$
$\quad \textsf{case } \_ : Stm'_{ex};$		$\quad \textsf{case } \_ : Stm''_{ex};$		$\quad \textsf{case } \_ : Stm'_{ex} \sqcup Stm''_{ex};$
Stm		Stm'		$Stm \sqcup Stm'$

$tryexcept$

Expression

$f(\overline{Exp}) \sqcup f(\overline{Exp'}) = f(\overline{Exp \sqcup Exp'})$

$Exp.f(\overline{Exp}) \sqcup Exp'.f(\overline{Exp'}) = (Exp \sqcup Exp').f(\overline{Exp \sqcup Exp'})$

Normaliser

Statements

$[\![\textsf{pass}]\!] = \textsf{pass}$

$[\![\textsf{return } Exp]\!] = \textsf{return } [\![Exp]\!]$

$$\textsc{noop}(Exp) = \begin{cases} [blank] & \text{if } Exp \in \{\texttt{Unit.id, None}\} \\ Exp & \text{otherwise} \end{cases}$$

$$[\![Exp_1 \ AsgOp \ Exp_2; Stm]\!] = \begin{cases} [\![Exp_1]\!]; [\![Stm]\!] & \text{if } \textsc{noop}([\![Exp_2]\!]) = [blank] \\ [\![Exp_2]\!]; [\![Stm]\!] & \text{if } \textsc{noop}([\![Exp_1]\!]) = [blank] \\ [\![Stm]\!] & \text{if } \textsc{noop}([\![Exp_1]\!], [\![Exp_2]\!]) = [blank] \\ [\![Exp_1]\!] AsgOp [\![Exp_2]\!]; [\![Stm]\!] & \text{otherwise} \end{cases}$$

$$[\![Exp; Stm]\!] = \begin{cases} [\![Stm]\!] & \text{if } \textsc{noop}([\![Exp]\!]) = [blank] \\ [\![Exp]\!]; [\![Stm]\!] & \text{otherwise} \end{cases}$$

$[\![\textsf{if } Exp : Stm_1 \ ; \ \textsf{else} : Stm_2 \ ; Stm]\!] = \textsf{if } [\![Exp]\!] : [\![Stm_1]\!] \ ; \ \textsf{else} : [\![Stm_2]\!] \ ; [\![Stm]\!]$

$[\![\textsf{match } Exp : \overline{\textsf{case } id : Stm'}; \textsf{case } \_ : Stm''; Stm]\!] = \textsf{match} [\![Exp]\!] : \overline{\textsf{case } id : [\![Stm']\!]}; \ \textsf{else} : [\![Stm'']\!]; [\![Stm]\!]$