EXP-UNTYPED-ENVIRONMENT

MODULE EXP-UNTYPED-ENVIRONMENT-SYNTAX

SYNTAX *Exp* ::= (*Exp*)
| `if` *Exp* `then` *Exp* `else` *Exp* [strict(1)]
| *Exp* *Exp* [strict]
| μ*Id*.*Exp*
| λ*Id*.*Exp*
| `let` *Id* `=` *Exp* `in` *Exp*
| `letrec` *Id* *Id* `=` *Exp* `in` *Exp*
| *Int*
| *Bool*
| *Id*
| *Exp* `+` *Exp* [strict]
| *Exp* `-` *Exp* [strict]
| *Exp* `*` *Exp* [strict]
| *Exp* `/` *Exp* [strict]
| *Exp* `%` *Exp* [strict]
| `-` *Exp* [strict]
| *Exp* `<` *Exp* [strict]
| *Exp* `<=` *Exp* [strict]
| *Exp* `>` *Exp* [strict]
| *Exp* `>=` *Exp* [strict]
| *Exp* `==` *Exp* [strict]
| *Exp* `!=` *Exp* [strict]
| *Exp* `and` *Exp* [strict]
| *Exp* `or` *Exp* [strict]
| `not` *Exp* [strict]

RULE $\dfrac{\texttt{let}\ X\ \texttt{=}\ E\ \texttt{in}\ E'}{\lambda X.E'\ \ E}$ [macro, anywhere]

RULE $\dfrac{\texttt{letrec}\ F\ \ X\ \texttt{=}\ E\ \texttt{in}\ E'}{\texttt{let}\ F\ \texttt{=}\ \mu F.\lambda X.E\ \texttt{in}\ E'}$ [macro, anywhere]

END MODULE

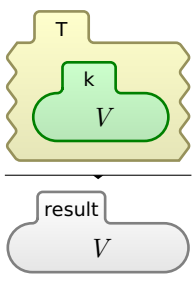
MODULE EXP-UNTYPED-ENVIRONMENT

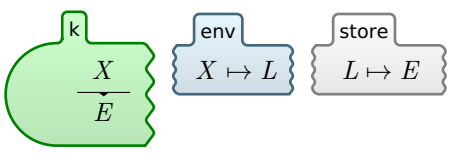
SYNTAX *Val* ::= *Int*
| *Bool*

SYNTAX *Exp* ::= *Val*

SYNTAX *KResult* ::= *Val*

CONFIGURATION:

RULE 

RULE 

RULE $\dfrac{I1\ \texttt{+}\ I2}{I1\ +_{Int}\ I2}$

RULE $\dfrac{I1\ \texttt{-}\ I2}{I1\ -_{Int}\ I2}$

RULE $\dfrac{I1\ \texttt{*}\ I2}{I1\ *_{Int}\ I2}$

RULE $\dfrac{I1\ \texttt{/}\ I2}{I1\ \div_{Int}\ I2}$ when $I2$ =/=Int 0

RULE $\dfrac{I1\ \texttt{\%}\ I2}{I1\ \%_{Int}\ I2}$ when $I2$ =/=Int 0

RULE $\dfrac{\texttt{-}\ I}{\texttt{0}\ -_{Int}\ I}$

RULE $\dfrac{I1\ \texttt{<}\ I2}{I1\ <_{Int}\ I2}$

RULE $\dfrac{I1\ \texttt{<=}\ I2}{I1\ \leq_{Int}\ I2}$

RULE $\dfrac{I1\ \texttt{>}\ I2}{I1\ >_{Int}\ I2}$

RULE $\dfrac{I1\ \texttt{>=}\ I2}{I1\ \geq_{Int}\ I2}$

RULE $\dfrac{V1\ \texttt{==}\ V2}{V1\ \texttt{==K}\ V2}$

RULE $\dfrac{V1\ \texttt{!=}\ V2}{V1\ \texttt{=/=K}\ V2}$

RULE $\dfrac{T1\ \texttt{and}\ T2}{T1\ \wedge_{Bool}\ T2}$

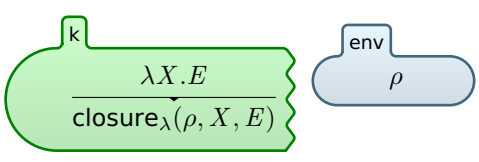
RULE $\dfrac{T1\ \texttt{or}\ T2}{T1\ \vee_{Bool}\ T2}$

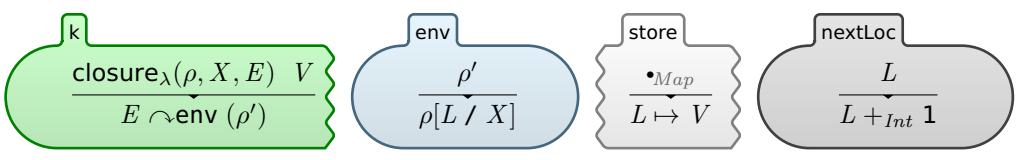
RULE $\dfrac{\texttt{not}\ T}{\neg_{Bool}\ T}$

RULE $\dfrac{\texttt{if true then}\ E\ \texttt{else}\ \_}{E}$

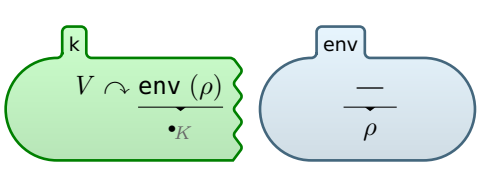
RULE $\dfrac{\texttt{if false then}\ \_\ \texttt{else}\ E}{E}$

SYNTAX *Val* ::= $\texttt{closure}_{\lambda}(Map, Id, Exp)$

RULE 

RULE 

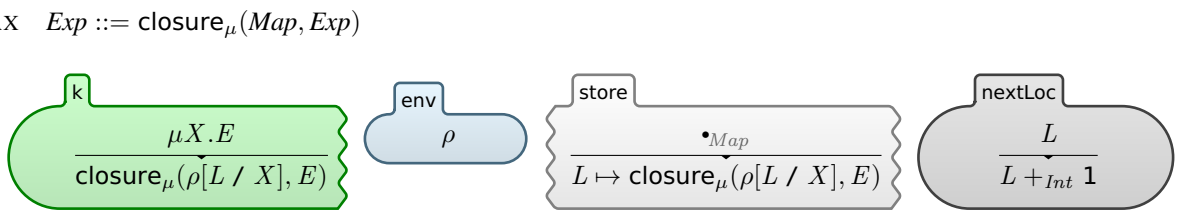
SYNTAX *K* ::= $\texttt{env}\ (Map)$

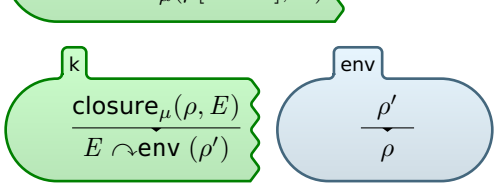
RULE 

[structural]

RULE $\dfrac{\texttt{env}\ (\_) \curvearrowright \texttt{env}\ (\_)}{\bullet_K}$ [structural]

SYNTAX *Exp* ::= $\texttt{closure}_{\mu}(Map, Exp)$

RULE 

RULE 

END MODULE