A formal specification of the Albert language (proposal)

Bruno Bernardo, Raphaël Cauderlier, Julien Tesson

June 2019

Contents

1 Introduction

The Michelson language for the smart contracts on the Tezos blockchain has been designed with software verification in mind. It is currently possible to formally verify the functional correctness of simple Michelson contracts such as the multisig contract using the Mi-Cho-Coq framework[8]. Concretely, we define in the Coq proof assistant a purely functional program, *the functional model* of the contract and we prove in Coq that the Michelson contract and its functional model are semantically equivalent.

The functional model defined in Coq need not be efficient. We can use very naive implementations for which it is easier to get convinced that the underlying logic is the one the programmer expected. For example, in the multisig contract[5, 6], the Michelson code uses the main loop to count the number of provided signatures and to check their validity; the functional model however split these two aspects in two recursive functions, one for counting the number of non-`None` elements in a list of options and one for checking the signatures without counting them.

While formally specifying Michelson's semantics in Coq and proving a few simple smart contracts was relatively straightforward, Michelson has not been designed to be massively adopted. In order to attract programmers familiar with popular high-level languages such as Python, Javascript, ML, and Pascal, new languages such as Ligo[2], SmartPy[4], FI[3], Morley[7], and Archetype[1] are being developed. Our ultimate goal is to certify compilers for these languages to Michelson in the Coq proof assistant so that formal verification of smart-contracts can be conducted directly in the high level languages. To achieve this, we need to share as much as possible between these certified compiler. Hence the need for a common intermediate compilation target that should:

- remain agnostic with respect to the overlying language;
- abstract away aspects of Michelson that are abstract in all the high-level languages such as the Michelson stack in order to share optimisations.

We are currently designing such an intermediate language called Albert. This document is a first proposal of a formal specification of the Albert language that is intended as a basis for discussing the features that Albert should have.

2 Albert specification

We detail here the formal specification of the Albert language. This formal specification is written using Ott from which a Coq definition of the Albert language is generated and type-checked.

To simplify the presentation, we have split the Albert language into a collection of small language fragments. The complete Albert language resulting from the merging of all fragments is given in Appendix.

2.1 Records and linear typing

Albert is a linearly-typed programming language. Resources in Albert are tracked by the type system and duplicating or destroying a resource are explicit operations.

Records in Albert generalize both the Michelson stack types and the Michelson pair type their syntax is given in fig. 1.

label, *l*, *var*, *x*	::=		Label / variable
	| *id*	S	
ty, *a*, *b*, *c*	::=		Type
	| *rty*		Record type
	| (*ty*)	S	Parenthesised type
rty	::=		Record type
	| $\{l_1 : ty_1 ; .. ; l_n : ty_n\}$		

Figure 1: Syntax of the record types

Γ	::=	Typing context (for inlined definitions)
	| .	

Figure 2: Syntax of the typing contextx

In the record type $\{l_1 : ty_1; ..; l_n : ty_n\}$, we assume the labels to be sorted in lexicographic order, so as to simplify future formal specifications and proofs.

The label sort constraint is formalized by well-formedness judgement $\Gamma| - ty$ defined in fig. 3 in which the typing context Γ follows the trivial gammar of fig. 2 (but other cases for typing contexts will be added in other language fragments). The `sorted` predicate correspond to the `StronglySorted` Coq predicates used with a strict inferior inequality, which ensures the labels are also distinct.

$$\frac{\begin{array}{l}\texttt{sorted}\,\{l_1 ; .. ; l_n\} \\ \Gamma \vdash ty_1 \quad .. \quad \Gamma \vdash ty_n \\ \Gamma \vdash \end{array}}{\Gamma \vdash \{l_1 : ty_1 ; .. ; l_n : ty_n\}} \text{ TYPING\_TWF\_RECORD}$$

Because the stack is abstracted, we cannot choose how the elements are ordered on the stack (which is unrelated to the lexicographic ordering used in the formal specification); we have no equivalent in Albert to the `SWAP` and `DIP` instructions of Michelson. Moreover, we have to name everything because we cannot point to an element on the stack by its level.

$\boxed{\Gamma \vdash ty}$ Type well-formedness

$$\frac{\begin{array}{l}\texttt{sorted}\,\{l_1\,;\;..\,;\;l_n\} \\ \Gamma \vdash ty_1 \quad .. \quad \Gamma \vdash ty_n \\ \Gamma \vdash \end{array}}{\Gamma \vdash \{l_1 \;:\; ty_1\,;\;..\,;\;l_n \;:\; ty_n\}} \;\textsc{typing\_twf\_record}$$

Figure 3: Type well-formedness judgment

$\boxed{\Gamma \vdash ty_1 \equiv ty_2}$ Type equality

$$\frac{}{\Gamma \vdash ty \equiv ty} \;\textsc{typing\_Teq\_refl}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_2}{\Gamma \vdash ty_2 \equiv ty_1} \;\textsc{typing\_Teq\_sym}$$

$$\frac{\begin{array}{l}\Gamma \vdash ty_1 \equiv ty_2 \\ \Gamma \vdash ty_2 \equiv ty_3\end{array}}{\Gamma \vdash ty_1 \equiv ty_3} \;\textsc{typing\_Teq\_trans}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_1' \quad .. \quad \Gamma \vdash ty_n \equiv ty_n'}{\Gamma \vdash \{l_1 \;:\; ty_1\,;\;..\,;\;l_n \;:\; ty_n\} \equiv \{l_1 \;:\; ty_1'\,;\;..\,;\;l_n \;:\; ty_n'\}} \;\textsc{typing\_Teq\_congr}$$

Figure 4: Type equality judgment

instruction, I, *ins*	::=		Instruction
	|	`noop`	No operation
	|	$instruction_1$ `;` $instruction_2$	Sequencing
	|	*lhs*`=`*rhs*	Assignment
lhs	::=		Left-hand side of assignement
	|	*var*	
	|	`{`l_1`=`var_1 `;` .. `;` l_n`=`var_n`}`	
rhs	::=		Right-hand side of assignments
	|	*arg*	
	|	*f arg*	
	|	*var*.*l*	
	|	`{`*var* `with` l_1`=`var_1 `;` ... `;` l_n`=`var_n`}`	
f	::=		Function symbol
arg	::=		Function argument
	|	*var*	
	|	*value*	
	|	`{`l_1`=`var_1 `;` ... `;` l_n`=`var_n`}`	

However, like in Michelson, dropping or duplicating a resource are explicit operations in Albert. Resources are tracked using a linear type system. Each (sequence of) instruction in Albert is typed by a pair of record types that play a similar role to the pair of stack types in Michelson: the first indicates the type of the input of the instruction and the second one indicates the type of the output.

$\boxed{\Gamma \vdash instruction : ty \Rightarrow ty'}$ Instruction typing

$$\frac{\begin{array}{l}\Gamma \vdash rty_1 \\ \Gamma \vdash rty_2 \\ rty \odot rty'' = rty_1 \\ rty' \odot rty'' = rty_2 \\ \Gamma \vdash instruction : rty \Rightarrow rty'\end{array}}{\Gamma \vdash instruction : rty_1 \Rightarrow rty_2}\ \textsc{typing\_t\_frame}$$

$$\frac{}{\Gamma \vdash \texttt{noop} : \texttt{\{\}} \Rightarrow \texttt{\{\}}}\ \textsc{typing\_t\_noop}$$

$$\frac{\begin{array}{l}\Gamma \vdash instruction : ty_1 \Rightarrow ty_2 \\ \Gamma \vdash instruction' : ty_2 \Rightarrow ty_3\end{array}}{\Gamma \vdash instruction\,;\ instruction' : ty_1 \Rightarrow ty_3}\ \textsc{typing\_t\_seq}$$

$$\frac{\Gamma \vdash rhs \ : \ a \Rightarrow b \qquad \Gamma \vdash lhs \ : \ b \Rightarrow c}{\Gamma \vdash lhs{=}rhs \ : \ a \Rightarrow c} \ \textsc{typing\_T\_assign}$$

$\boxed{\Gamma \vdash lhs \ : \ ty \Rightarrow ty'}$ Left-hand sides typing

$$\frac{}{\Gamma \vdash var \ : \ ty \Rightarrow \{var \ : \ ty\}} \ \textsc{typing\_Tlhs\_var}$$

$$\frac{}{\Gamma \vdash \{l_1{=}x_1 \, ; \ .. \ ; \ l_n{=}x_n\} \ : \ \{l_1 \ : \ ty_1 \, ; \ .. \ ; \ l_n \ : \ ty_n\} \Rightarrow \{x_1 \ : \ ty_1 \, ; \ .. \ ; \ x_n \ : \ ty_n\}} \ \textsc{typing\_Tlhs\_record}$$

$\boxed{\Gamma \vdash rhs \ : \ ty \Rightarrow ty'}$ Right-hand side typing

$$\frac{\Gamma \vdash_a arg \ : \ ty \Rightarrow ty'}{\Gamma \vdash arg \ : \ ty \Rightarrow ty'} \ \textsc{typing\_Trhs\_arg}$$

$$\frac{\Gamma \vdash_a arg \ : \ ty \Rightarrow ty' \qquad \Gamma \vdash f \ : \ ty' \Rightarrow ty''}{\Gamma \vdash f \ arg \ : \ ty \Rightarrow ty''} \ \textsc{typing\_Trhs\_f}$$

$$\frac{\{l \ : \ ty\} \odot rty{=}rty' \qquad \Gamma \vdash rty'}{\Gamma \vdash var.l \ : \ rty' \Rightarrow ty} \ \textsc{typing\_Trhs\_projection}$$

$$\frac{\Gamma \vdash rty' \qquad \{l_1 \ : \ ty_1 \, ; \ .. \ ; \ l_n \ : \ ty_n\} \odot rty{=}rty'}{\Gamma \vdash \{var \ \texttt{with} \ l_1{=}var_1 \, ; \ .. \ ; \ l_n{=}var_n\} \ : \ \{var \ : \ rty' ; \ var_1 \ : \ ty_1 \, ; \ .. \ ; \ var_n \ : \ ty_n\} \Rightarrow rty'} \ \textsc{typing\_Trhs\_upd}$$

$\boxed{\Gamma \vdash f \ : \ ty \Rightarrow ty'}$ Function symbol typing

$\boxed{\Gamma \vdash_a arg \ : \ ty \Rightarrow ty'}$ Argument typing

$$\frac{}{\Gamma \vdash_a var \ : \ \{var \ : \ ty\} \Rightarrow ty} \ \textsc{typing\_Targ\_var}$$

$$\frac{\Gamma \vdash value \ : \ ty}{\Gamma \vdash_a value \ : \ \{\} \Rightarrow ty} \ \textsc{typing\_Targ\_value}$$

$$\frac{}{\Gamma \vdash_a \{l_1{=}x_1 \, ; \ .. \ ; \ l_n{=}x_n\} \ : \ \{x_1 \ : \ ty_1 \, ; \ .. \ ; \ x_n \ : \ ty_n\} \Rightarrow \{l_1 \ : \ ty_1 \, ; \ .. \ ; \ l_n \ : \ ty_n\}} \ \textsc{typing\_Targ\_record}$$

The pair type $a * b$ that is equivalent to Michelson's `pair a b` and is used by the duplication function `dup` is defined as a record with two fields called `car` and `cdr`.

n, m

id

rty	::=		Record type
	|	$\{l_1 : ty_1 ; \ldots ; l_n : ty_n\}$	
instruction, I, *ins*	::=		Instruction
	|	`noop`	No operation
	|	$instruction_1 ; instruction_2$	Sequencing
	|	$lhs = rhs$	Assignment
rhs	::=		Right-hand side of assignments
	|	*arg*	
	|	$f\ arg$	
	|	$var.l$	
	|	$\{var\ \texttt{with}\ l_1 = var_1 ; \ldots ; l_n = var_n\}$	
f	::=		Function symbol
arg	::=		Function argument
	|	*var*	
	|	*value*	
	|	$\{l_1 = var_1 ; \ldots ; l_n = var_n\}$	
rval	::=		Record value
	|	$\{l_1 = value_1 ; \ldots ; l_n = value_n\}$	Record
value, *val*	::=		Value
	|	*rval*	Record
	|	`(`*value*`)`	S
Γ	::=		Typing context (for inlined definitions)
	|	.	
products	::=		Products of types
	|	$\{ty_1 * ty_1' ; \ldots ; ty_n * ty_n'\}$	
label, l, *var*, x	::=		Label / variable
	|	*id*	S

$ty,\ a,\ b,\ c$	::=			Type
	|	rty		Record type
	|	(ty)	S	Parenthesised type
	|	ty `*` ty'		Binary product type

lhs	::=			Left-hand side of assignement
	|	var		
	|	$\{l_1{=}var_1\ ;\ ..\ ;\ l_n{=}var_n\}$		
	|	(var_1, var_2)	S	

$\boxed{\mathtt{E} \vdash lhs/val \texttt{ -> } val'}$ Left-hand side evaluation

$$\frac{}{\mathtt{E} \vdash var/val \texttt{ -> } \{var{=}val\}}\ \textsc{eval\_lhs\_var}$$

$$\frac{}{\mathtt{E} \vdash \{l_1{=}x_1\ ;\ ..\ ;\ l_n{=}x_n\}/\{l_1{=}val_1\ ;\ ..\ ;\ l_n{=}val_n\} \texttt{ -> } \{x_1{=}val_1\ ;\ ..\ ;\ x_n{=}val_n\}}\ \textsc{eval\_lhs\_record}$$

$\boxed{\mathtt{E} \vdash arg/aval \texttt{ -> } val'}$ Argument evaluation

$$\frac{}{\mathtt{E} \vdash var/a\{var{=}val\} \texttt{ -> } val}\ \textsc{eval\_arg\_var}$$

$$\frac{}{\mathtt{E} \vdash val/a\{\} \texttt{ -> } val}\ \textsc{eval\_arg\_val}$$

$$\frac{}{\mathtt{E} \vdash \{l_1{=}x_1\ ;\ ..\ ;\ l_n{=}x_n\}/a\{x_1{=}val_1\ ;\ ..\ ;\ x_n{=}val_n\} \texttt{ -> } \{l_1{=}val_1\ ;\ ..\ ;\ l_n{=}val_n\}}\ \textsc{eval\_arg\_record}$$

$\boxed{\mathtt{E} \vdash f/val \texttt{ -> } val'}$ Function symbol evaluation

$\boxed{\mathtt{E} \vdash rhs/val \texttt{ -> } val'}$ Right-hand side evaluation

$$\frac{\mathtt{E} \vdash arg/aval \texttt{ -> } val'}{\mathtt{E} \vdash arg/val \texttt{ -> } val'}\ \textsc{eval\_rhs\_arg}$$

$$\frac{\begin{array}{l}\mathtt{E} \vdash arg/aval \texttt{ -> } val' \\ \mathtt{E} \vdash f/val' \texttt{ -> } val''\end{array}}{\mathtt{E} \vdash f\ arg/val \texttt{ -> } val''}\ \textsc{eval\_rhs\_f}$$

$$\frac{\{l{=}val\} \odot rval{=}rval'}{\mathtt{E} \vdash var.l/rval' \texttt{ -> } val}\ \textsc{eval\_rhs\_projection}$$

$$\frac{\begin{array}{l}\{l_1{=}val'_1\ ;\ ..\ ;\ l_n{=}val'_n\} \odot rval{=}rval' \\ \{l_1{=}val_1\ ;\ ..\ ;\ l_n{=}val_n\} \odot rval{=}rval''\end{array}}{\mathtt{E} \vdash \{var\ \texttt{with}\ l_1{=}var_1\ ;\ ..\ ;\ l_n{=}var_n\}/\{var{=}rval'\ ;\ var_1{=}val_1\ ;\ ..\ ;\ var_n{=}val_n\} \texttt{ -> } rval''}\ \textsc{eval\_rhs\_update}$$

$\boxed{\mathtt{E} \vdash instruction/val \texttt{ -> } val'}$ Instruction evaluation

$$\frac{\texttt{E} \vdash \mathit{instruction}/\mathit{rval} \texttt{ -> } \mathit{rval}' \quad \mathit{rval} \odot \mathit{rval}''{=}\mathit{rval}_1 \quad \mathit{rval}' \odot \mathit{rval}''{=}\mathit{rval}_2}{\texttt{E} \vdash \mathit{instruction}/\mathit{rval}_1 \texttt{ -> } \mathit{rval}_2}\ \textsc{eval\_instr\_frame}$$

$$\frac{}{\texttt{E} \vdash \texttt{noop/\{\} -> \{\}}}\ \textsc{eval\_instr\_noop}$$

$$\frac{\texttt{E} \vdash I_1/\mathit{val} \texttt{ -> } \mathit{val}' \quad \texttt{E} \vdash I_2/\mathit{val}' \texttt{ -> } \mathit{val}''}{\texttt{E} \vdash I_1\texttt{;}\ I_2/\mathit{val} \texttt{ -> } \mathit{val}''}\ \textsc{eval\_instr\_seq}$$

$$\frac{\texttt{E} \vdash \mathit{rhs}/\mathit{val} \texttt{ -> } \mathit{val}' \quad \texttt{E} \vdash \mathit{lhs}/\mathit{val}' \texttt{ -> } \mathit{val}''}{\texttt{E} \vdash \mathit{lhs}{=}\mathit{rhs}/\mathit{val} \texttt{ -> } \mathit{val}''}\ \textsc{eval\_instr\_assign}$$

$\boxed{\Gamma \vdash}$ Context well-fornedness

$$\frac{}{. \vdash}\ \textsc{typing\_Gwf\_empty}$$

$\boxed{\Gamma \vdash \mathit{instruction} : \mathit{ty} \Rightarrow \mathit{ty}'}$ Instruction typing

$$\frac{\Gamma \vdash \mathit{rty}_1 \quad \Gamma \vdash \mathit{rty}_2 \quad \mathit{rty} \odot \mathit{rty}''{=}\mathit{rty}_1 \quad \mathit{rty}' \odot \mathit{rty}''{=}\mathit{rty}_2 \quad \Gamma \vdash \mathit{instruction} : \mathit{rty} \Rightarrow \mathit{rty}'}{\Gamma \vdash \mathit{instruction} : \mathit{rty}_1 \Rightarrow \mathit{rty}_2}\ \textsc{typing\_T\_frame}$$

$$\frac{}{\Gamma \vdash \texttt{noop} : \texttt{\{\}} \Rightarrow \texttt{\{\}}}\ \textsc{typing\_T\_noop}$$

$$\frac{\Gamma \vdash \mathit{instruction} : \mathit{ty}_1 \Rightarrow \mathit{ty}_2 \quad \Gamma \vdash \mathit{instruction}' : \mathit{ty}_2 \Rightarrow \mathit{ty}_3}{\Gamma \vdash \mathit{instruction}\texttt{;}\ \mathit{instruction}' : \mathit{ty}_1 \Rightarrow \mathit{ty}_3}\ \textsc{typing\_T\_seq}$$

$$\frac{\Gamma \vdash \mathit{rhs} : a \Rightarrow b \quad \Gamma \vdash \mathit{lhs} : b \Rightarrow c}{\Gamma \vdash \mathit{lhs}{=}\mathit{rhs} : a \Rightarrow c}\ \textsc{typing\_T\_assign}$$

$\boxed{\Gamma \vdash \mathit{lhs} : \mathit{ty} \Rightarrow \mathit{ty}'}$ Left-hand sides typing

$$\frac{}{\Gamma \vdash \mathit{var} : \mathit{ty} \Rightarrow \{\mathit{var} : \mathit{ty}\}}\ \textsc{typing\_Tlhs\_var}$$

$$\frac{}{\Gamma \vdash \{l_1{=}x_1\texttt{;}\ \ldots\texttt{;}\ l_n{=}x_n\} : \{l_1 : \mathit{ty}_1\texttt{;}\ \ldots\texttt{;}\ l_n : \mathit{ty}_n\} \Rightarrow \{x_1 : \mathit{ty}_1\texttt{;}\ \ldots\texttt{;}\ x_n : \mathit{ty}_n\}}\ \textsc{typing\_Tlhs\_record}$$

$\boxed{\Gamma \vdash_a arg \ : \ ty \Rightarrow ty'}$ Argument typing

$$\frac{}{\Gamma \vdash_a var \ : \ \{var \ : \ ty\} \Rightarrow ty} \ \textsc{typing\_Targ\_var}$$

$$\frac{\Gamma \vdash value \ : \ ty}{\Gamma \vdash_a value \ : \ \{\,\} \Rightarrow ty} \ \textsc{typing\_Targ\_value}$$

$$\frac{}{\Gamma \vdash_a \{l_1{=}x_1 \,; \ .. \,; \ l_n{=}x_n\} \ : \ \{x_1 \ : \ ty_1 \,; \ .. \,; \ x_n \ : \ ty_n\} \Rightarrow \{l_1 \ : \ ty_1 \,; \ .. \,; \ l_n \ : \ ty_n\}} \ \textsc{typing\_Targ\_record}$$

$\boxed{\Gamma \vdash rhs \ : \ ty \Rightarrow ty'}$ Right-hand side typing

$$\frac{\Gamma \vdash_a arg \ : \ ty \Rightarrow ty'}{\Gamma \vdash arg \ : \ ty \Rightarrow ty'} \ \textsc{typing\_Trhs\_arg}$$

$$\frac{\Gamma \vdash_a arg \ : \ ty \Rightarrow ty' \quad \Gamma \vdash f \ : \ ty' \Rightarrow ty''}{\Gamma \vdash f \ arg \ : \ ty \Rightarrow ty''} \ \textsc{typing\_Trhs\_f}$$

$$\frac{\{l \ : \ ty\} \odot rty{=}rty' \quad \Gamma \vdash rty'}{\Gamma \vdash var.l \ : \ rty' \Rightarrow ty} \ \textsc{typing\_Trhs\_projection}$$

$$\frac{\Gamma \vdash rty' \quad \{l_1 \ : \ ty_1 \,; \ .. \,; \ l_n \ : \ ty_n\} \odot rty{=}rty'}{\Gamma \vdash \{var \ \texttt{with} \ l_1{=}var_1 \,; \ .. \,; \ l_n{=}var_n\} \ : \ \{var \ : \ rty' \,; \ var_1 \ : \ ty_1 \,; \ .. \,; \ var_n \ : \ ty_n\} \Rightarrow rty'} \ \textsc{typing\_Trhs\_upd}$$

$\boxed{\Gamma \vdash f \ : \ ty \Rightarrow ty'}$ Function symbol typing

$\boxed{\Gamma \vdash value \ : \ ty}$ Value typing

$$\frac{\Gamma \vdash val_1 \ : \ ty_1 \quad .. \quad \Gamma \vdash val_n \ : \ ty_n}{\Gamma \vdash \{l_1{=}val_1 \,; \ .. \,; \ l_n{=}val_n\} \ : \ \{l_1 \ : \ ty_1 \,; \ .. \,; \ l_n \ : \ ty_n\}} \ \textsc{typing\_Tval\_record}$$

$\boxed{\Gamma \vdash ty_1 \equiv ty_2}$ Type equality

$$\frac{}{\Gamma \vdash ty \equiv ty} \ \textsc{typing\_Teq\_refl}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_2}{\Gamma \vdash ty_2 \equiv ty_1} \ \textsc{typing\_Teq\_sym}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_2 \quad \Gamma \vdash ty_2 \equiv ty_3}{\Gamma \vdash ty_1 \equiv ty_3} \ \textsc{typing\_Teq\_trans}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_1' \quad .. \quad \Gamma \vdash ty_n \equiv ty_n'}{\Gamma \vdash \{l_1 \ : \ ty_1 \,; \ .. \,; \ l_n \ : \ ty_n\} \equiv \{l_1 \ : \ ty_1' \,; \ .. \,; \ l_n \ : \ ty_n'\}} \ \textsc{typing\_Teq\_congr}$$

$$\frac{}{\Gamma \vdash ty_1 \texttt{ * } ty_2 \equiv \texttt{\{car : } ty_1\texttt{; cdr : } ty_2\texttt{\}}} \quad \textsc{typing\_Teq\_pair}$$

$\boxed{\Gamma \vdash ty}$ Type well-formedness

$$\frac{\begin{array}{l}\texttt{sorted\{}l_1\texttt{; .. ; }l_n\texttt{\}} \\ \Gamma \vdash ty_1 \quad .. \quad \Gamma \vdash ty_n \\ \Gamma \vdash \end{array}}{\Gamma \vdash \texttt{\{}l_1 \texttt{ : } ty_1\texttt{; .. ; }l_n \texttt{ : } ty_n\texttt{\}}} \quad \textsc{typing\_twf\_record}$$

$$\frac{\begin{array}{l}\Gamma \vdash ty \\ \Gamma \vdash ty'\end{array}}{\Gamma \vdash ty \texttt{ * } ty'} \quad \textsc{typing\_twf\_pair}$$

$\boxed{\texttt{tvar} \notin \mathsf{FV}(ty)}$ Type variable freshness

$$\frac{\texttt{tvar} \notin \mathsf{FV}(ty_1) \quad .. \quad \texttt{tvar} \notin \mathsf{FV}(ty_n)}{\texttt{tvar} \notin \mathsf{FV}(\texttt{\{}l_1 \texttt{ : } ty_1\texttt{; .. ; }l_n \texttt{ : } ty_n\texttt{\}})} \quad \textsc{typing\_Tfresh\_tvarrecord}$$

$$\frac{\begin{array}{l}\texttt{tvar} \notin \mathsf{FV}(ty) \\ \texttt{tvar} \notin \mathsf{FV}(ty')\end{array}}{\texttt{tvar} \notin \mathsf{FV}(ty \texttt{ * } ty')} \quad \textsc{typing\_Tfresh\_tvarpair}$$

n, m			
id			
rty	::=		Record type
	|	$\texttt{\{}l_1 \texttt{ : } ty_1\texttt{; .. ; }l_n \texttt{ : } ty_n\texttt{\}}$	
$instruction$, I, ins	::=		Instruction
	|	$\texttt{noop}$	No operation
	|	$instruction_1\texttt{; }instruction_2$	Sequencing
	|	$lhs\texttt{=}rhs$	Assignment
rhs	::=		Right-hand side of assignments
	|	arg	
	|	$f\ arg$	
	|	$var.l$	
	|	$\texttt{\{}var\ \texttt{with}\ l_1\texttt{=}var_1\texttt{; ... ; }l_n\texttt{=}var_n\texttt{\}}$	
arg	::=		Function argument
	|	var	
	|	$value$	

	|	$\{l_1{=}var_1\,;\ \ldots\ ;\ l_n{=}var_n\}$	
rval	::=		Record value
	|	$\{l_1{=}value_1\,;\ ..\ ;\ l_n{=}value_n\}$	Record
value, *val*	::=		Value
	|	*rval*	Record
	|	(*value*)	S
Γ	::=		Typing context (for inlined definitions)
	|	.	
products	::=		Products of types
	|	$\{ty_1 * ty_1'\,;\ ..\ ;\ ty_n * ty_n'\}$	
label, *l*, *var*, *x*	::=		Label / variable
	|	*id*	S
ty, *a*, *b*, *c*	::=		Type
	|	*rty*	Record type
	|	(*ty*)	S Parenthesised type
	|	$ty * ty'$	Binary product type
lhs	::=		Left-hand side of assignement
	|	*var*	
	|	$\{l_1{=}var_1\,;\ ..\ ;\ l_n{=}var_n\}$	
	|	(var_1, var_2)	S
f	::=		Function symbol
	|	`dup`	

$\boxed{\Gamma \vdash}$ Context well-fornedness

$$\frac{}{\cdot \vdash}\ \textsc{typing\_Gwf\_empty}$$

$\boxed{\Gamma \vdash \mathit{instruction} : ty \Rightarrow ty'}$ Instruction typing

$$\frac{\begin{array}{l}\Gamma \vdash rty_1 \\ \Gamma \vdash rty_2 \\ rty \odot rty''{=}rty_1 \\ rty' \odot rty''{=}rty_2 \\ \Gamma \vdash instruction \ : \ rty \Rightarrow rty'\end{array}}{\Gamma \vdash instruction \ : \ rty_1 \Rightarrow rty_2} \quad \textsc{typing\_T\_frame}$$

$$\frac{}{\Gamma \vdash \texttt{noop} \ : \ \{\,\} \Rightarrow \{\,\}} \quad \textsc{typing\_T\_noop}$$

$$\frac{\begin{array}{c}\Gamma \vdash instruction \ : \ ty_1 \Rightarrow ty_2 \\ \Gamma \vdash instruction' \ : \ ty_2 \Rightarrow ty_3\end{array}}{\Gamma \vdash instruction ; \ instruction' \ : \ ty_1 \Rightarrow ty_3} \quad \textsc{typing\_T\_seq}$$

$$\frac{\begin{array}{c}\Gamma \vdash rhs \ : \ a \Rightarrow b \\ \Gamma \vdash lhs \ : \ b \Rightarrow c\end{array}}{\Gamma \vdash lhs{=}rhs \ : \ a \Rightarrow c} \quad \textsc{typing\_T\_assign}$$

$\boxed{\Gamma \vdash lhs \ : \ ty \Rightarrow ty'}$ Left-hand sides typing

$$\frac{}{\Gamma \vdash var \ : \ ty \Rightarrow \{var \ : \ ty\}} \quad \textsc{typing\_Tlhs\_var}$$

$$\frac{}{\Gamma \vdash \{l_1{=}x_1 ; \ .. \ ; \ l_n{=}x_n\} \ : \ \{l_1 \ : \ ty_1 ; \ .. \ ; \ l_n \ : \ ty_n\} \Rightarrow \{x_1 \ : \ ty_1 ; \ .. \ ; \ x_n \ : \ ty_n\}} \quad \textsc{typing\_Tlhs\_record}$$

$\boxed{\Gamma \vdash_a arg \ : \ ty \Rightarrow ty'}$ Argument typing

$$\frac{}{\Gamma \vdash_a var \ : \ \{var \ : \ ty\} \Rightarrow ty} \quad \textsc{typing\_Targ\_var}$$

$$\frac{\Gamma \vdash value \ : \ ty}{\Gamma \vdash_a value \ : \ \{\,\} \Rightarrow ty} \quad \textsc{typing\_Targ\_value}$$

$$\frac{}{\Gamma \vdash_a \{l_1{=}x_1 ; \ .. \ ; \ l_n{=}x_n\} \ : \ \{x_1 \ : \ ty_1 ; \ .. \ ; \ x_n \ : \ ty_n\} \Rightarrow \{l_1 \ : \ ty_1 ; \ .. \ ; \ l_n \ : \ ty_n\}} \quad \textsc{typing\_Targ\_record}$$

$\boxed{\Gamma \vdash rhs \ : \ ty \Rightarrow ty'}$ Right-hand side typing

$$\frac{\Gamma \vdash_a arg \ : \ ty \Rightarrow ty'}{\Gamma \vdash arg \ : \ ty \Rightarrow ty'} \quad \textsc{typing\_Trhs\_arg}$$

$$\frac{\begin{array}{c}\Gamma \vdash_a arg \ : \ ty \Rightarrow ty' \\ \Gamma \vdash f \ : \ ty' \Rightarrow ty''\end{array}}{\Gamma \vdash f \ arg \ : \ ty \Rightarrow ty''} \quad \textsc{typing\_Trhs\_f}$$

$$\frac{\begin{array}{l}\{l \ : \ ty\} \odot rty{=}rty' \\ \Gamma \vdash rty'\end{array}}{\Gamma \vdash var.l \ : \ rty' \Rightarrow ty} \quad \textsc{typing\_Trhs\_projection}$$

$$\frac{\begin{array}{l}\Gamma \vdash rty' \\ \{l_1 : ty_1;\ ..;\ l_n : ty_n\} \odot rty{=}rty'\end{array}}{\Gamma \vdash \{var\ \texttt{with}\ l_1{=}var_1;\ ..;\ l_n{=}var_n\} : \{var : rty';\ var_1 : ty_1;\ ..;\ var_n : ty_n\} \Rightarrow rty'}\ \textsc{typing\_Trhs\_upd}$$

$\boxed{\Gamma \vdash value : ty}$ Value typing

$$\frac{\Gamma \vdash val_1 : ty_1 \quad .. \quad \Gamma \vdash val_n : ty_n}{\Gamma \vdash \{l_1{=}val_1;\ ..;\ l_n{=}val_n\} : \{l_1 : ty_1;\ ..;\ l_n : ty_n\}}\ \textsc{typing\_Tval\_record}$$

$\boxed{\Gamma \vdash ty_1 \equiv ty_2}$ Type equality

$$\frac{}{\Gamma \vdash ty \equiv ty}\ \textsc{typing\_Teq\_refl}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_2}{\Gamma \vdash ty_2 \equiv ty_1}\ \textsc{typing\_Teq\_sym}$$

$$\frac{\begin{array}{l}\Gamma \vdash ty_1 \equiv ty_2 \\ \Gamma \vdash ty_2 \equiv ty_3\end{array}}{\Gamma \vdash ty_1 \equiv ty_3}\ \textsc{typing\_Teq\_trans}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty'_1 \quad .. \quad \Gamma \vdash ty_n \equiv ty'_n}{\Gamma \vdash \{l_1 : ty_1;\ ..;\ l_n : ty_n\} \equiv \{l_1 : ty'_1;\ ..;\ l_n : ty'_n\}}\ \textsc{typing\_Teq\_congr}$$

$$\frac{}{\Gamma \vdash ty_1 * ty_2 \equiv \{\texttt{car} : ty_1;\ \texttt{cdr} : ty_2\}}\ \textsc{typing\_Teq\_pair}$$

$\boxed{\Gamma \vdash ty}$ Type well-formedness

$$\frac{\begin{array}{l}\texttt{sorted}\,\{l_1;\ ..;\ l_n\} \\ \Gamma \vdash ty_1 \quad .. \quad \Gamma \vdash ty_n \\ \Gamma \vdash\end{array}}{\Gamma \vdash \{l_1 : ty_1;\ ..;\ l_n : ty_n\}}\ \textsc{typing\_twf\_record}$$

$$\frac{\begin{array}{l}\Gamma \vdash ty \\ \Gamma \vdash ty'\end{array}}{\Gamma \vdash ty * ty'}\ \textsc{typing\_twf\_pair}$$

$\boxed{\texttt{tvar} \notin \mathsf{FV}(ty)}$ Type variable freshness

$$\frac{\texttt{tvar} \notin \mathsf{FV}(ty_1) \quad .. \quad \texttt{tvar} \notin \mathsf{FV}(ty_n)}{\texttt{tvar} \notin \mathsf{FV}(\{l_1 : ty_1;\ ..;\ l_n : ty_n\})}\ \textsc{typing\_Tfresh\_tvarrecord}$$

$$\frac{\begin{array}{l}\texttt{tvar} \notin \mathsf{FV}(ty) \\ \texttt{tvar} \notin \mathsf{FV}(ty')\end{array}}{\texttt{tvar} \notin \mathsf{FV}(ty * ty')}\ \textsc{typing\_Tfresh\_tvarpair}$$

$\boxed{\Gamma \vdash f \ : \ ty \Rightarrow ty'}$ Function symbol typing

$$\frac{}{\Gamma \vdash \texttt{dup} \ : \ ty \Rightarrow ty \ \texttt{*} \ ty} \ \text{TYPING\_TF\_DUP}$$

$\boxed{\texttt{E} \vdash lhs/val \ \texttt{->} \ val'}$ Left-hand side evaluation

$$\frac{}{\texttt{E} \vdash var/val \ \texttt{->} \ \{var\texttt{=}val\}} \ \text{EVAL\_LHS\_VAR}$$

$$\frac{}{\texttt{E} \vdash \{l_1\texttt{=}x_1\texttt{;} \ \ldots\texttt{;} \ l_n\texttt{=}x_n\}/\{l_1\texttt{=}val_1\texttt{;} \ \ldots\texttt{;} \ l_n\texttt{=}val_n\} \ \texttt{->} \ \{x_1\texttt{=}val_1\texttt{;} \ \ldots\texttt{;} \ x_n\texttt{=}val_n\}} \ \text{EVAL\_LHS\_RECORD}$$

$\boxed{\texttt{E} \vdash arg/aval \ \texttt{->} \ val'}$ Argument evaluation

$$\frac{}{\texttt{E} \vdash var/a\{var\texttt{=}val\} \ \texttt{->} \ val} \ \text{EVAL\_ARG\_VAR}$$

$$\frac{}{\texttt{E} \vdash val/a\{\,\} \ \texttt{->} \ val} \ \text{EVAL\_ARG\_VAL}$$

$$\frac{}{\texttt{E} \vdash \{l_1\texttt{=}x_1\texttt{;} \ \ldots\texttt{;} \ l_n\texttt{=}x_n\}/a\{x_1\texttt{=}val_1\texttt{;} \ \ldots\texttt{;} \ x_n\texttt{=}val_n\} \ \texttt{->} \ \{l_1\texttt{=}val_1\texttt{;} \ \ldots\texttt{;} \ l_n\texttt{=}val_n\}} \ \text{EVAL\_ARG\_RECORD}$$

$\boxed{\texttt{E} \vdash rhs/val \ \texttt{->} \ val'}$ Right-hand side evaluation

$$\frac{\texttt{E} \vdash arg/aval \ \texttt{->} \ val'}{\texttt{E} \vdash arg/val \ \texttt{->} \ val'} \ \text{EVAL\_RHS\_ARG}$$

$$\frac{\begin{array}{l}\texttt{E} \vdash arg/aval \ \texttt{->} \ val' \\ \texttt{E} \vdash f/val' \ \texttt{->} \ val''\end{array}}{\texttt{E} \vdash f \ arg/val \ \texttt{->} \ val''} \ \text{EVAL\_RHS\_F}$$

$$\frac{\{l\texttt{=}val\} \odot rval\texttt{=}rval'}{\texttt{E} \vdash var.l/rval' \ \texttt{->} \ val} \ \text{EVAL\_RHS\_PROJECTION}$$

$$\frac{\begin{array}{l}\{l_1\texttt{=}val'_1\texttt{;} \ \ldots\texttt{;} \ l_n\texttt{=}val'_n\} \odot rval\texttt{=}rval' \\ \{l_1\texttt{=}val_1\texttt{;} \ \ldots\texttt{;} \ l_n\texttt{=}val_n\} \odot rval\texttt{=}rval''\end{array}}{\texttt{E} \vdash \{var \ \texttt{with} \ l_1\texttt{=}var_1\texttt{;} \ \ldots\texttt{;} \ l_n\texttt{=}var_n\}/\{var\texttt{=}rval'\texttt{;} \ var_1\texttt{=}val_1\texttt{;} \ \ldots\texttt{;} \ var_n\texttt{=}val_n\} \ \texttt{->} \ rval''} \ \text{EVAL\_RHS\_UPDATE}$$

$\boxed{\texttt{E} \vdash instruction/val \ \texttt{->} \ val'}$ Instruction evaluation

$$\frac{\begin{array}{l}\texttt{E} \vdash instruction/rval \ \texttt{->} \ rval' \\ rval \odot rval''\texttt{=}rval_1 \\ rval' \odot rval''\texttt{=}rval_2\end{array}}{\texttt{E} \vdash instruction/rval_1 \ \texttt{->} \ rval_2} \ \text{EVAL\_INSTR\_FRAME}$$

$$\frac{}{\texttt{E} \vdash \texttt{noop}/\{\,\} \ \texttt{->} \ \{\,\}} \ \text{EVAL\_INSTR\_NOOP}$$

$$\frac{\texttt{E} \vdash I_1/val \texttt{ -> } val' \quad \texttt{E} \vdash I_2/val' \texttt{ -> } val''}{\texttt{E} \vdash I_1\texttt{;}\ I_2/val \texttt{ -> } val''}\ \text{EVAL\_INSTR\_SEQ}$$

$$\frac{\texttt{E} \vdash rhs/val \texttt{ -> } val' \quad \texttt{E} \vdash lhs/val' \texttt{ -> } val''}{\texttt{E} \vdash lhs\texttt{=}rhs/val \texttt{ -> } val''}\ \text{EVAL\_INSTR\_ASSIGN}$$

$\boxed{\texttt{E} \vdash f/val \texttt{ -> } val'}$ Function symbol evaluation

$$\frac{}{\texttt{E} \vdash \texttt{dup}/val \texttt{ -> \{car=}val\texttt{; cdr=}val\texttt{\}}}\ \text{EVAL\_F\_DUP}$$

Similarly, the type `unit` is equivalent to Michelson's `unit`, it is used by the dropping instruction `drop` and is defined as a record with no field.

n, m

id

label, *l*, *var*, *x*	::=		Label / variable
	|	*id* S	
rty	::=		Record type
	|	{l_1 : ty_1 ; .. ; l_n : ty_n}	
instruction, *I*, *ins*	::=		Instruction
	|	`noop`	No operation
	|	*instruction*$_1$; *instruction*$_2$	Sequencing
	|	*lhs*=*rhs*	Assignment
lhs	::=		Left-hand side of assignement
	|	*var*	
	|	{l_1=var_1 ; .. ; l_n=var_n}	
rhs	::=		Right-hand side of assignments
	|	*arg*	
	|	*f arg*	
	|	*var*.*l*	
	|	{*var* `with` l_1=var_1 ; ... ; l_n=var_n}	
f	::=		Function symbol
arg	::=		Function argument

	|	*var*	
	|	*value*	
	|	{l_1=*var*$_1$; .. ; l_n=*var*$_n$}	
rval	::=		Record value
	|	{l_1=*value*$_1$; .. ; l_n=*value*$_n$}	Record
value, *val*	::=		Value
	|	*rval*	Record
	|	(*value*)	S
Γ	::=		Typing context (for inlined definitions)
	|	.	
products	::=		Products of types
	|	{ty_1 * ty'_1 ; .. ; ty_n * ty'_n}	
ty, *a*, *b*, *c*	::=		Type
	|	*rty*	Record type
	|	(*ty*)	S Parenthesised type
	|	unit	

$\boxed{\texttt{E} \vdash lhs/val \texttt{ -> } val'}$ Left-hand side evaluation

$$\frac{}{\texttt{E} \vdash var/val \texttt{ -> } \{var=val\}}\ \text{EVAL\_LHS\_VAR}$$

$$\frac{}{\texttt{E} \vdash \{l_1=x_1\texttt{ ; }..\texttt{ ; }l_n=x_n\}/\{l_1=val_1\texttt{ ; }..\texttt{ ; }l_n=val_n\} \texttt{ -> } \{x_1=val_1\texttt{ ; }..\texttt{ ; }x_n=val_n\}}\ \text{EVAL\_LHS\_RECORD}$$

$\boxed{\texttt{E} \vdash arg/aval \texttt{ -> } val'}$ Argument evaluation

$$\frac{}{\texttt{E} \vdash var/a\{var=val\} \texttt{ -> } val}\ \text{EVAL\_ARG\_VAR}$$

$$\frac{}{\texttt{E} \vdash val/a\{\} \texttt{ -> } val}\ \text{EVAL\_ARG\_VAL}$$

$$\frac{}{\texttt{E} \vdash \{l_1=x_1\texttt{ ; }..\texttt{ ; }l_n=x_n\}/a\{x_1=val_1\texttt{ ; }..\texttt{ ; }x_n=val_n\} \texttt{ -> } \{l_1=val_1\texttt{ ; }..\texttt{ ; }l_n=val_n\}}\ \text{EVAL\_ARG\_RECORD}$$

$\boxed{\texttt{E} \vdash f/val \texttt{ -> } val'}$ Function symbol evaluation

$\boxed{\texttt{E} \vdash rhs/val \texttt{ -> } val'}$ Right-hand side evaluation

$$\frac{\texttt{E} \vdash arg/aval \texttt{ -> } val'}{\texttt{E} \vdash arg/val \texttt{ -> } val'} \quad \textsc{eval\_rhs\_arg}$$

$$\frac{\begin{array}{c}\texttt{E} \vdash arg/aval \texttt{ -> } val' \\ \texttt{E} \vdash f/val' \texttt{ -> } val''\end{array}}{\texttt{E} \vdash f\ arg/val \texttt{ -> } val''} \quad \textsc{eval\_rhs\_f}$$

$$\frac{\{l{=}val\} \odot rval{=}rval'}{\texttt{E} \vdash var.l/rval' \texttt{ -> } val} \quad \textsc{eval\_rhs\_projection}$$

$$\frac{\begin{array}{c}\{l_1{=}val'_1\texttt{;}\ \ldots\texttt{;}\ l_n{=}val'_n\} \odot rval{=}rval' \\ \{l_1{=}val_1\texttt{;}\ \ldots\texttt{;}\ l_n{=}val_n\} \odot rval{=}rval''\end{array}}{\texttt{E} \vdash \{var\ \texttt{with}\ l_1{=}var_1\texttt{;}\ \ldots\texttt{;}\ l_n{=}var_n\}/\{var{=}rval'\texttt{;}\ var_1{=}val_1\texttt{;}\ \ldots\texttt{;}\ var_n{=}val_n\} \texttt{ -> } rval''} \quad \textsc{eval\_rhs\_update}$$

$\boxed{\texttt{E} \vdash instruction/val \texttt{ -> } val'}$ Instruction evaluation

$$\frac{\begin{array}{l}\texttt{E} \vdash instruction/rval \texttt{ -> } rval' \\ rval \odot rval''{=}rval_1 \\ rval' \odot rval''{=}rval_2\end{array}}{\texttt{E} \vdash instruction/rval_1 \texttt{ -> } rval_2} \quad \textsc{eval\_instr\_frame}$$

$$\frac{}{\texttt{E} \vdash \texttt{noop/\{\} -> \{\}}} \quad \textsc{eval\_instr\_noop}$$

$$\frac{\begin{array}{c}\texttt{E} \vdash I_1/val \texttt{ -> } val' \\ \texttt{E} \vdash I_2/val' \texttt{ -> } val''\end{array}}{\texttt{E} \vdash I_1\texttt{;}\ I_2/val \texttt{ -> } val''} \quad \textsc{eval\_instr\_seq}$$

$$\frac{\begin{array}{c}\texttt{E} \vdash rhs/val \texttt{ -> } val' \\ \texttt{E} \vdash lhs/val' \texttt{ -> } val''\end{array}}{\texttt{E} \vdash lhs{=}rhs/val \texttt{ -> } val''} \quad \textsc{eval\_instr\_assign}$$

$\boxed{\Gamma \vdash}$ Context well-fornedness

$$\frac{}{.\vdash} \quad \textsc{typing\_Gwf\_empty}$$

$\boxed{\Gamma \vdash instruction\ :\ ty \Rightarrow ty'}$ Instruction typing

$$\frac{\begin{array}{l}\Gamma \vdash rty_1 \\ \Gamma \vdash rty_2 \\ rty \odot rty''{=}rty_1 \\ rty' \odot rty''{=}rty_2 \\ \Gamma \vdash instruction\ :\ rty \Rightarrow rty'\end{array}}{\Gamma \vdash instruction\ :\ rty_1 \Rightarrow rty_2} \quad \textsc{typing\_T\_frame}$$

$$\frac{}{\Gamma \vdash \texttt{noop} : \{\,\} \Rightarrow \{\,\}} \quad \textsc{typing\_T\_noop}$$

$$\frac{\Gamma \vdash \mathit{instruction} : \mathit{ty}_1 \Rightarrow \mathit{ty}_2 \quad \Gamma \vdash \mathit{instruction}' : \mathit{ty}_2 \Rightarrow \mathit{ty}_3}{\Gamma \vdash \mathit{instruction}\,;\ \mathit{instruction}' : \mathit{ty}_1 \Rightarrow \mathit{ty}_3} \quad \textsc{typing\_T\_seq}$$

$$\frac{\Gamma \vdash \mathit{rhs} : a \Rightarrow b \quad \Gamma \vdash \mathit{lhs} : b \Rightarrow c}{\Gamma \vdash \mathit{lhs}\texttt{=}\mathit{rhs} : a \Rightarrow c} \quad \textsc{typing\_T\_assign}$$

$\boxed{\Gamma \vdash \mathit{lhs} : \mathit{ty} \Rightarrow \mathit{ty}'}$ Left-hand sides typing

$$\frac{}{\Gamma \vdash \mathit{var} : \mathit{ty} \Rightarrow \{\mathit{var} : \mathit{ty}\}} \quad \textsc{typing\_Tlhs\_var}$$

$$\frac{}{\Gamma \vdash \{l_1\texttt{=}x_1\,;\ ..\ ;\ l_n\texttt{=}x_n\} : \{l_1 : \mathit{ty}_1\,;\ ..\ ;\ l_n : \mathit{ty}_n\} \Rightarrow \{x_1 : \mathit{ty}_1\,;\ ..\ ;\ x_n : \mathit{ty}_n\}} \quad \textsc{typing\_Tlhs\_record}$$

$\boxed{\Gamma \vdash_a \mathit{arg} : \mathit{ty} \Rightarrow \mathit{ty}'}$ Argument typing

$$\frac{}{\Gamma \vdash_a \mathit{var} : \{\mathit{var} : \mathit{ty}\} \Rightarrow \mathit{ty}} \quad \textsc{typing\_Targ\_var}$$

$$\frac{\Gamma \vdash \mathit{value} : \mathit{ty}}{\Gamma \vdash_a \mathit{value} : \{\,\} \Rightarrow \mathit{ty}} \quad \textsc{typing\_Targ\_value}$$

$$\frac{}{\Gamma \vdash_a \{l_1\texttt{=}x_1\,;\ ..\ ;\ l_n\texttt{=}x_n\} : \{x_1 : \mathit{ty}_1\,;\ ..\ ;\ x_n : \mathit{ty}_n\} \Rightarrow \{l_1 : \mathit{ty}_1\,;\ ..\ ;\ l_n : \mathit{ty}_n\}} \quad \textsc{typing\_Targ\_record}$$

$\boxed{\Gamma \vdash \mathit{rhs} : \mathit{ty} \Rightarrow \mathit{ty}'}$ Right-hand side typing

$$\frac{\Gamma \vdash_a \mathit{arg} : \mathit{ty} \Rightarrow \mathit{ty}'}{\Gamma \vdash \mathit{arg} : \mathit{ty} \Rightarrow \mathit{ty}'} \quad \textsc{typing\_Trhs\_arg}$$

$$\frac{\Gamma \vdash_a \mathit{arg} : \mathit{ty} \Rightarrow \mathit{ty}' \quad \Gamma \vdash f : \mathit{ty}' \Rightarrow \mathit{ty}''}{\Gamma \vdash f\ \mathit{arg} : \mathit{ty} \Rightarrow \mathit{ty}''} \quad \textsc{typing\_Trhs\_f}$$

$$\frac{\{l : \mathit{ty}\} \odot \mathit{rty}\texttt{=}\mathit{rty}' \quad \Gamma \vdash \mathit{rty}'}{\Gamma \vdash \mathit{var}.l : \mathit{rty}' \Rightarrow \mathit{ty}} \quad \textsc{typing\_Trhs\_projection}$$

$$\frac{\Gamma \vdash \mathit{rty}' \quad \{l_1 : \mathit{ty}_1\,;\ ..\ ;\ l_n : \mathit{ty}_n\} \odot \mathit{rty}\texttt{=}\mathit{rty}'}{\Gamma \vdash \{\mathit{var}\ \texttt{with}\ l_1\texttt{=}\mathit{var}_1\,;\ ..\ ;\ l_n\texttt{=}\mathit{var}_n\} : \{\mathit{var} : \mathit{rty}'\,;\ \mathit{var}_1 : \mathit{ty}_1\,;\ ..\ ;\ \mathit{var}_n : \mathit{ty}_n\} \Rightarrow \mathit{rty}'} \quad \textsc{typing\_Trhs\_upd}$$

$\boxed{\Gamma \vdash f : \mathit{ty} \Rightarrow \mathit{ty}'}$ Function symbol typing

$\boxed{\Gamma \vdash \mathit{value} : \mathit{ty}}$ Value typing

$$\frac{\Gamma \vdash val_1 \ : \ ty_1 \quad .. \quad \Gamma \vdash val_n \ : \ ty_n}{\Gamma \vdash \{l_1{=}val_1 \ ; \ .. \ ; \ l_n{=}val_n\} \ : \ \{l_1 \ : \ ty_1 \ ; \ .. \ ; \ l_n \ : \ ty_n\}} \quad \textsc{typing\_Tval\_record}$$

$\boxed{\Gamma \vdash ty_1 \equiv ty_2}$ Type equality

$$\frac{}{\Gamma \vdash ty \equiv ty} \quad \textsc{typing\_Teq\_refl}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_2}{\Gamma \vdash ty_2 \equiv ty_1} \quad \textsc{typing\_Teq\_sym}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_2 \quad \Gamma \vdash ty_2 \equiv ty_3}{\Gamma \vdash ty_1 \equiv ty_3} \quad \textsc{typing\_Teq\_trans}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_1' \quad .. \quad \Gamma \vdash ty_n \equiv ty_n'}{\Gamma \vdash \{l_1 \ : \ ty_1 \ ; \ .. \ ; \ l_n \ : \ ty_n\} \equiv \{l_1 \ : \ ty_1' \ ; \ .. \ ; \ l_n \ : \ ty_n'\}} \quad \textsc{typing\_Teq\_congr}$$

$$\frac{}{\Gamma \vdash \texttt{unit} \equiv \{\}} \quad \textsc{typing\_Teq\_unit}$$

$\boxed{\Gamma \vdash ty}$ Type well-formedness

$$\frac{\begin{array}{l}\texttt{sorted}\,\{l_1 \ ; \ .. \ ; \ l_n\} \\ \Gamma \vdash ty_1 \quad .. \quad \Gamma \vdash ty_n \\ \Gamma \vdash \end{array}}{\Gamma \vdash \{l_1 \ : \ ty_1 \ ; \ .. \ ; \ l_n \ : \ ty_n\}} \quad \textsc{typing\_Twf\_record}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \texttt{unit}} \quad \textsc{typing\_Twf\_unit}$$

$\boxed{\texttt{tvar} \notin \mathsf{FV}(ty)}$ Type variable freshness

$$\frac{\texttt{tvar} \notin \mathsf{FV}(ty_1) \quad .. \quad \texttt{tvar} \notin \mathsf{FV}(ty_n)}{\texttt{tvar} \notin \mathsf{FV}(\{l_1 \ : \ ty_1 \ ; \ .. \ ; \ l_n \ : \ ty_n\})} \quad \textsc{typing\_Tfresh\_tvarrecord}$$

$$\frac{}{\texttt{tvar} \notin \mathsf{FV}(\texttt{unit})} \quad \textsc{typing\_Tfresh\_tvarunit}$$

n, m			
id			
$label$, l, var, x	::=		Label / variable
	|	id	S
rty	::=		Record type
	|	$\{l_1 \ : \ ty_1 \ ; \ .. \ ; \ l_n \ : \ ty_n\}$	

lhs	::=		Left-hand side of assignement
	|	*var*	
	|	{l_1=var_1 ; .. ; l_n=var_n}	
rhs	::=		Right-hand side of assignments
	|	*arg*	
	|	*f arg*	
	|	*var*.*l*	
	|	{*var* with l_1=var_1 ; ... ; l_n=var_n}	
f	::=		Function symbol
arg	::=		Function argument
	|	*var*	
	|	*value*	
	|	{l_1=var_1 ; ... ; l_n=var_n}	
rval	::=		Record value
	|	{l_1=$value_1$; .. ; l_n=$value_n$}	Record
value, *val*	::=		Value
	|	*rval*	Record
	|	(*value*) S	
Γ	::=		Typing context (for inlined definitions)
	|	.	
products	::=		Products of types
	|	{ty_1 * ty'_1 ; .. ; ty_n * ty'_n}	
ty, *a*, *b*, *c*	::=		Type
	|	*rty*	Record type
	|	(*ty*) S	Parenthesised type
	|	unit	
instruction, *I*, *ins*	::=		Instruction
	|	noop	No operation

	$\mathit{instruction}_1$; $\mathit{instruction}_2$	Sequencing
|	lhs=rhs	Assignment
|	`drop` var	Resource dropping

$\boxed{\Gamma \vdash}$ Context well-fornedness

$$\frac{}{. \vdash} \quad \textsc{typing\_Gwf\_empty}$$

$\boxed{\Gamma \vdash \mathit{lhs} : \mathit{ty} \Rightarrow \mathit{ty}'}$ Left-hand sides typing

$$\frac{}{\Gamma \vdash \mathit{var} : \mathit{ty} \Rightarrow \{\mathit{var} : \mathit{ty}\}} \quad \textsc{typing\_Tlhs\_var}$$

$$\frac{}{\Gamma \vdash \{l_1{=}x_1 ; \ .. \ ; \ l_n{=}x_n\} : \{l_1 : \mathit{ty}_1 ; \ .. \ ; \ l_n : \mathit{ty}_n\} \Rightarrow \{x_1 : \mathit{ty}_1 ; \ .. \ ; \ x_n : \mathit{ty}_n\}} \quad \textsc{typing\_Tlhs\_record}$$

$\boxed{\Gamma \vdash_a \mathit{arg} : \mathit{ty} \Rightarrow \mathit{ty}'}$ Argument typing

$$\frac{}{\Gamma \vdash_a \mathit{var} : \{\mathit{var} : \mathit{ty}\} \Rightarrow \mathit{ty}} \quad \textsc{typing\_Targ\_var}$$

$$\frac{\Gamma \vdash \mathit{value} : \mathit{ty}}{\Gamma \vdash_a \mathit{value} : \{\,\} \Rightarrow \mathit{ty}} \quad \textsc{typing\_Targ\_value}$$

$$\frac{}{\Gamma \vdash_a \{l_1{=}x_1 ; \ .. \ ; \ l_n{=}x_n\} : \{x_1 : \mathit{ty}_1 ; \ .. \ ; \ x_n : \mathit{ty}_n\} \Rightarrow \{l_1 : \mathit{ty}_1 ; \ .. \ ; \ l_n : \mathit{ty}_n\}} \quad \textsc{typing\_Targ\_record}$$

$\boxed{\Gamma \vdash \mathit{rhs} : \mathit{ty} \Rightarrow \mathit{ty}'}$ Right-hand side typing

$$\frac{\Gamma \vdash_a \mathit{arg} : \mathit{ty} \Rightarrow \mathit{ty}'}{\Gamma \vdash \mathit{arg} : \mathit{ty} \Rightarrow \mathit{ty}'} \quad \textsc{typing\_Trhs\_arg}$$

$$\frac{\Gamma \vdash_a \mathit{arg} : \mathit{ty} \Rightarrow \mathit{ty}' \quad \Gamma \vdash f : \mathit{ty}' \Rightarrow \mathit{ty}''}{\Gamma \vdash f \ \mathit{arg} : \mathit{ty} \Rightarrow \mathit{ty}''} \quad \textsc{typing\_Trhs\_f}$$

$$\frac{\{l : \mathit{ty}\} \odot \mathit{rty}{=}\mathit{rty}' \quad \Gamma \vdash \mathit{rty}'}{\Gamma \vdash \mathit{var}.l : \mathit{rty}' \Rightarrow \mathit{ty}} \quad \textsc{typing\_Trhs\_projection}$$

$$\frac{\Gamma \vdash \mathit{rty}' \quad \{l_1 : \mathit{ty}_1 ; \ .. \ ; \ l_n : \mathit{ty}_n\} \odot \mathit{rty}{=}\mathit{rty}'}{\Gamma \vdash \{\mathit{var}\ \texttt{with}\ l_1{=}\mathit{var}_1 ; \ .. \ ; \ l_n{=}\mathit{var}_n\} : \{\mathit{var} : \mathit{rty}' ; \ \mathit{var}_1 : \mathit{ty}_1 ; \ .. \ ; \ \mathit{var}_n : \mathit{ty}_n\} \Rightarrow \mathit{rty}'} \quad \textsc{typing\_Trhs\_upd}$$

$\boxed{\Gamma \vdash f : \mathit{ty} \Rightarrow \mathit{ty}'}$ Function symbol typing

$\boxed{\Gamma \vdash \mathit{value} : \mathit{ty}}$ Value typing

$$\frac{\Gamma \vdash val_1 \ : \ ty_1 \quad .. \quad \Gamma \vdash val_n \ : \ ty_n}{\Gamma \vdash \{l_1{=}val_1 \, ; \ .. \ ; \ l_n{=}val_n\} \ : \ \{l_1 \ : \ ty_1 \, ; \ .. \ ; \ l_n \ : \ ty_n\}} \quad \textsc{typing\_Tval\_record}$$

$\boxed{\Gamma \vdash ty_1 \equiv ty_2}$ Type equality

$$\frac{}{\Gamma \vdash ty \equiv ty} \quad \textsc{typing\_Teq\_refl}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_2}{\Gamma \vdash ty_2 \equiv ty_1} \quad \textsc{typing\_Teq\_sym}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_2 \quad \Gamma \vdash ty_2 \equiv ty_3}{\Gamma \vdash ty_1 \equiv ty_3} \quad \textsc{typing\_Teq\_trans}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_1' \quad .. \quad \Gamma \vdash ty_n \equiv ty_n'}{\Gamma \vdash \{l_1 \ : \ ty_1 \, ; \ .. \ ; \ l_n \ : \ ty_n\} \equiv \{l_1 \ : \ ty_1' \, ; \ .. \ ; \ l_n \ : \ ty_n'\}} \quad \textsc{typing\_Teq\_congr}$$

$$\frac{}{\Gamma \vdash \mathtt{unit} \equiv \{\}} \quad \textsc{typing\_Teq\_unit}$$

$\boxed{\Gamma \vdash ty}$ Type well-formedness

$$\frac{\begin{array}{l}\mathtt{sorted}\,\{l_1 \, ; \ .. \ ; \ l_n\} \\ \Gamma \vdash ty_1 \quad .. \quad \Gamma \vdash ty_n \\ \Gamma \vdash \end{array}}{\Gamma \vdash \{l_1 \ : \ ty_1 \, ; \ .. \ ; \ l_n \ : \ ty_n\}} \quad \textsc{typing\_twf\_record}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \mathtt{unit}} \quad \textsc{typing\_twf\_unit}$$

$\boxed{\mathtt{tvar} \notin \mathsf{FV}(ty)}$ Type variable freshness

$$\frac{\mathtt{tvar} \notin \mathsf{FV}(ty_1) \quad .. \quad \mathtt{tvar} \notin \mathsf{FV}(ty_n)}{\mathtt{tvar} \notin \mathsf{FV}(\{l_1 \ : \ ty_1 \, ; \ .. \ ; \ l_n \ : \ ty_n\})} \quad \textsc{typing\_Tfresh\_tvarrecord}$$

$$\frac{}{\mathtt{tvar} \notin \mathsf{FV}(\mathtt{unit})} \quad \textsc{typing\_Tfresh\_tvarunit}$$

$\boxed{\Gamma \vdash instruction \ : \ ty \Rightarrow ty'}$ Instruction typing

$$\frac{\begin{array}{l}\Gamma \vdash rty_1 \\ \Gamma \vdash rty_2 \\ rty \odot rty''{=}rty_1 \\ rty' \odot rty''{=}rty_2 \\ \Gamma \vdash instruction \ : \ rty \Rightarrow rty'\end{array}}{\Gamma \vdash instruction \ : \ rty_1 \Rightarrow rty_2} \quad \textsc{typing\_T\_frame}$$

$$\frac{}{\Gamma \vdash \texttt{noop} : \{\,\} \Rightarrow \{\,\}}\ \textsc{typing\_t\_noop}$$

$$\frac{\Gamma \vdash \mathit{instruction} : \mathit{ty}_1 \Rightarrow \mathit{ty}_2 \quad \Gamma \vdash \mathit{instruction}' : \mathit{ty}_2 \Rightarrow \mathit{ty}_3}{\Gamma \vdash \mathit{instruction}\,;\ \mathit{instruction}' : \mathit{ty}_1 \Rightarrow \mathit{ty}_3}\ \textsc{typing\_t\_seq}$$

$$\frac{\Gamma \vdash \mathit{rhs} : a \Rightarrow b \quad \Gamma \vdash \mathit{lhs} : b \Rightarrow c}{\Gamma \vdash \mathit{lhs}{=}\mathit{rhs} : a \Rightarrow c}\ \textsc{typing\_t\_assign}$$

$$\frac{}{\Gamma \vdash \texttt{drop}\,\mathit{var} : \{\mathit{var} : \mathit{ty}\} \Rightarrow \texttt{unit}}\ \textsc{typing\_t\_drop}$$

$\boxed{\texttt{E} \vdash \mathit{lhs}/\mathit{val}\ \texttt{->}\ \mathit{val}'}$ Left-hand side evaluation

$$\frac{}{\texttt{E} \vdash \mathit{var}/\mathit{val}\ \texttt{->}\ \{\mathit{var}{=}\mathit{val}\}}\ \textsc{eval\_lhs\_var}$$

$$\frac{}{\texttt{E} \vdash \{l_1{=}x_1\,;\ ..\,;\ l_n{=}x_n\}/\{l_1{=}\mathit{val}_1\,;\ ..\,;\ l_n{=}\mathit{val}_n\}\ \texttt{->}\ \{x_1{=}\mathit{val}_1\,;\ ..\,;\ x_n{=}\mathit{val}_n\}}\ \textsc{eval\_lhs\_record}$$

$\boxed{\texttt{E} \vdash \mathit{arg}/\mathit{aval}\ \texttt{->}\ \mathit{val}'}$ Argument evaluation

$$\frac{}{\texttt{E} \vdash \mathit{var}/a\{\mathit{var}{=}\mathit{val}\}\ \texttt{->}\ \mathit{val}}\ \textsc{eval\_arg\_var}$$

$$\frac{}{\texttt{E} \vdash \mathit{val}/a\{\,\}\ \texttt{->}\ \mathit{val}}\ \textsc{eval\_arg\_val}$$

$$\frac{}{\texttt{E} \vdash \{l_1{=}x_1\,;\ ..\,;\ l_n{=}x_n\}/a\{x_1{=}\mathit{val}_1\,;\ ..\,;\ x_n{=}\mathit{val}_n\}\ \texttt{->}\ \{l_1{=}\mathit{val}_1\,;\ ..\,;\ l_n{=}\mathit{val}_n\}}\ \textsc{eval\_arg\_record}$$

$\boxed{\texttt{E} \vdash f/\mathit{val}\ \texttt{->}\ \mathit{val}'}$ Function symbol evaluation

$\boxed{\texttt{E} \vdash \mathit{rhs}/\mathit{val}\ \texttt{->}\ \mathit{val}'}$ Right-hand side evaluation

$$\frac{\texttt{E} \vdash \mathit{arg}/\mathit{aval}\ \texttt{->}\ \mathit{val}'}{\texttt{E} \vdash \mathit{arg}/\mathit{val}\ \texttt{->}\ \mathit{val}'}\ \textsc{eval\_rhs\_arg}$$

$$\frac{\texttt{E} \vdash \mathit{arg}/\mathit{aval}\ \texttt{->}\ \mathit{val}' \quad \texttt{E} \vdash f/\mathit{val}'\ \texttt{->}\ \mathit{val}''}{\texttt{E} \vdash f\ \mathit{arg}/\mathit{val}\ \texttt{->}\ \mathit{val}''}\ \textsc{eval\_rhs\_f}$$

$$\frac{\{l{=}\mathit{val}\} \odot \mathit{rval}{=}\mathit{rval}'}{\texttt{E} \vdash \mathit{var}.l/\mathit{rval}'\ \texttt{->}\ \mathit{val}}\ \textsc{eval\_rhs\_projection}$$

$$\frac{\{l_1{=}\mathit{val}'_1\,;\ ..\,;\ l_n{=}\mathit{val}'_n\} \odot \mathit{rval}{=}\mathit{rval}' \quad \{l_1{=}\mathit{val}_1\,;\ ..\,;\ l_n{=}\mathit{val}_n\} \odot \mathit{rval}{=}\mathit{rval}''}{\texttt{E} \vdash \{\mathit{var}\ \texttt{with}\ l_1{=}\mathit{var}_1\,;\ ..\,;\ l_n{=}\mathit{var}_n\}/\{\mathit{var}{=}\mathit{rval}'\,;\ \mathit{var}_1{=}\mathit{val}_1\,;\ ..\,;\ \mathit{var}_n{=}\mathit{val}_n\}\ \texttt{->}\ \mathit{rval}''}\ \textsc{eval\_rhs\_update}$$

$\boxed{\texttt{E} \vdash \mathit{instruction}/\mathit{val}\ \texttt{->}\ \mathit{val}'}$ Instruction evaluation

$$\frac{\begin{array}{l}\texttt{E} \vdash \mathit{instruction}/\mathit{rval}\ \texttt{->}\ \mathit{rval}' \\ \mathit{rval} \odot \mathit{rval}''\texttt{=}\mathit{rval}_1 \\ \mathit{rval}' \odot \mathit{rval}''\texttt{=}\mathit{rval}_2\end{array}}{\texttt{E} \vdash \mathit{instruction}/\mathit{rval}_1\ \texttt{->}\ \mathit{rval}_2}\ \textsc{eval\_instr\_frame}$$

$$\frac{}{\texttt{E} \vdash \texttt{noop}/\texttt{\{\}}\ \texttt{->}\ \texttt{\{\}}}\ \textsc{eval\_instr\_noop}$$

$$\frac{\begin{array}{l}\texttt{E} \vdash I_1/\mathit{val}\ \texttt{->}\ \mathit{val}' \\ \texttt{E} \vdash I_2/\mathit{val}'\ \texttt{->}\ \mathit{val}''\end{array}}{\texttt{E} \vdash I_1\texttt{;}\ I_2/\mathit{val}\ \texttt{->}\ \mathit{val}''}\ \textsc{eval\_instr\_seq}$$

$$\frac{\begin{array}{l}\texttt{E} \vdash \mathit{rhs}/\mathit{val}\ \texttt{->}\ \mathit{val}' \\ \texttt{E} \vdash \mathit{lhs}/\mathit{val}'\ \texttt{->}\ \mathit{val}''\end{array}}{\texttt{E} \vdash \mathit{lhs}\texttt{=}\mathit{rhs}/\mathit{val}\ \texttt{->}\ \mathit{val}''}\ \textsc{eval\_instr\_assign}$$

$$\frac{}{\texttt{E} \vdash \texttt{drop}\ \mathit{var}/\texttt{\{}\mathit{var}\texttt{=}\mathit{val}\texttt{\}}\ \texttt{->}\ \texttt{\{\}}}\ \textsc{eval\_instr\_drop}$$

2.2 Variants

Variants are the dual of records and generalize Michelson's `or` type to arbitrary arities with named constructors.

n, m			
id			
k			
rty	::=		Record type
	|	$\texttt{\{}l_1\ \texttt{:}\ \mathit{ty}_1\texttt{;}\ \ldots\texttt{;}\ l_n\ \texttt{:}\ \mathit{ty}_n\texttt{\}}$	
lhs	::=		Left-hand side of assignement
	|	var	
	|	$\texttt{\{}l_1\texttt{=}\mathit{var}_1\texttt{;}\ \ldots\texttt{;}\ l_n\texttt{=}\mathit{var}_n\texttt{\}}$	
arg	::=		Function argument
	|	var	
	|	value	
	|	$\texttt{\{}l_1\texttt{=}\mathit{var}_1\texttt{;}\ \ldots\texttt{;}\ l_n\texttt{=}\mathit{var}_n\texttt{\}}$	
rval	::=		Record value
	|	$\texttt{\{}l_1\texttt{=}\mathit{value}_1\texttt{;}\ \ldots\texttt{;}\ l_n\texttt{=}\mathit{value}_n\texttt{\}}$	Record
Γ	::=		Typing context (for inlined definitions)

	|	.	
products	::=		Products of ty
	|	{ty_1 * ty'_1; .. ; ty_n * ty'_n}	
constructor	::=		
	|	*id*	S
label, *l*, *var*, *x*	::=		Label / variab
	|	*id*	S
ty, *a*, *b*, *c*	::=		Type
	|	*rty*	Record type
	|	(*ty*)	S Parenthesis
	|	*vty*	
vty	::=		
	|	[$constructor_1$: ty_1 | ... | $constructor_k$: ty_k]	
f	::=		Function symb
	|	(*constructor* : *vty*)	
branches_rhs	::=		
	|	$constructor_1$ var_1 -> rhs_1 | ... | $constructor_k$ var_k -> rhs_k	
rhs	::=		Right-hand sic
	|	*arg*	
	|	*f arg*	
	|	*var.l*	
	|	{*var* with l_1=var_1; ... ; l_n=var_n}	
	|	match *var* with *branches_rhs* end	
branches_ins	::=		
	|	$constructor_1$ var_1 -> ins_1 | ... | $constructor_k$ var_k -> ins_k	
instruction, *I*, *ins*	::=		Instruction
	|	noop	No operatio

	$instruction_1$; $instruction_2$	Sequencing
|	*lhs*=*rhs*	Assignment
|	`match` *var* `with` *branches_ins* `end`	

value, *val*	::=			Value
	|	*rval*		Record
	|	(*value*)	S	
	|	(*constructor value* : *vty*)		

$\boxed{\Gamma \vdash}$ Context well-fornedness

$$\frac{}{. \vdash} \quad \textsc{typing\_Gwf\_empty}$$

$\boxed{\Gamma \vdash lhs : ty \Rightarrow ty'}$ Left-hand sides typing

$$\frac{}{\Gamma \vdash var : ty \Rightarrow \{var : ty\}} \quad \textsc{typing\_Tlhs\_var}$$

$$\frac{}{\Gamma \vdash \{l_1{=}x_1 ; \ .. \ ; \ l_n{=}x_n\} : \{l_1 : ty_1 ; \ .. \ ; \ l_n : ty_n\} \Rightarrow \{x_1 : ty_1 ; \ .. \ ; \ x_n : ty_n\}} \quad \textsc{typing\_Tlhs\_record}$$

$\boxed{\Gamma \vdash_a arg : ty \Rightarrow ty'}$ Argument typing

$$\frac{}{\Gamma \vdash_a var : \{var : ty\} \Rightarrow ty} \quad \textsc{typing\_Targ\_var}$$

$$\frac{\Gamma \vdash value : ty}{\Gamma \vdash_a value : \{\} \Rightarrow ty} \quad \textsc{typing\_Targ\_value}$$

$$\frac{}{\Gamma \vdash_a \{l_1{=}x_1 ; \ .. \ ; \ l_n{=}x_n\} : \{x_1 : ty_1 ; \ .. \ ; \ x_n : ty_n\} \Rightarrow \{l_1 : ty_1 ; \ .. \ ; \ l_n : ty_n\}} \quad \textsc{typing\_Targ\_record}$$

$\boxed{\Gamma \vdash ty_1 \equiv ty_2}$ Type equality

$$\frac{}{\Gamma \vdash ty \equiv ty} \quad \textsc{typing\_Teq\_refl}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_2}{\Gamma \vdash ty_2 \equiv ty_1} \quad \textsc{typing\_Teq\_sym}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_2 \quad \Gamma \vdash ty_2 \equiv ty_3}{\Gamma \vdash ty_1 \equiv ty_3} \quad \textsc{typing\_Teq\_trans}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_1' \quad .. \quad \Gamma \vdash ty_n \equiv ty_n'}{\Gamma \vdash \{l_1 : ty_1 ; \ .. \ ; \ l_n : ty_n\} \equiv \{l_1 : ty_1' ; \ .. \ ; \ l_n : ty_n'\}} \quad \textsc{typing\_Teq\_congr}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_1' \quad .. \quad \Gamma \vdash ty_k \equiv ty_k'}{\Gamma \vdash [constructor_1 : ty_1 \mid .. \mid constructor_k : ty_k] \equiv [constructor_1 : ty_1' \mid .. \mid constructor_k : ty_k']} \text{ TYPING\_}$$

$\boxed{\Gamma \vdash ty}$ Type well-formedness

$$\frac{\begin{array}{l}\texttt{sorted}\,\{l_1\,;\;..\,;\;l_n\} \\ \Gamma \vdash ty_1 \quad .. \quad \Gamma \vdash ty_n \\ \Gamma \vdash \end{array}}{\Gamma \vdash \{l_1 : ty_1\,;\;..\,;\;l_n : ty_n\}} \text{ TYPING\_TWF\_RECORD}$$

$$\frac{\begin{array}{l}\texttt{sorted}\,\{constructor_1\,;\;..\,;\;constructor_k\} \\ \Gamma \vdash ty_1 \quad .. \quad \Gamma \vdash ty_k \\ \Gamma \vdash \end{array}}{\Gamma \vdash [constructor_1 : ty_1 \mid .. \mid constructor_k : ty_k]} \text{ TYPING\_TWF\_VARIANT}$$

$\boxed{\texttt{tvar} \notin \mathsf{FV}(ty)}$ Type variable freshness

$$\frac{\texttt{tvar} \notin \mathsf{FV}(ty_1) \quad .. \quad \texttt{tvar} \notin \mathsf{FV}(ty_n)}{\texttt{tvar} \notin \mathsf{FV}(\{l_1 : ty_1\,;\;..\,;\;l_n : ty_n\})} \text{ TYPING\_TFRESH\_TVARRECORD}$$

$$\frac{\texttt{tvar} \notin \mathsf{FV}(ty_1) \quad .. \quad \texttt{tvar} \notin \mathsf{FV}(ty_k)}{\texttt{tvar} \notin \mathsf{FV}([constructor_1 : ty_1 \mid .. \mid constructor_k : ty_k])} \text{ TYPING\_TFRESH\_TVARVARIANT}$$

$\boxed{\Gamma \vdash rhs : ty \Rightarrow ty'}$ Right-hand side typing

$$\frac{\Gamma \vdash_a arg : ty \Rightarrow ty'}{\Gamma \vdash arg : ty \Rightarrow ty'} \text{ TYPING\_TRHS\_ARG}$$

$$\frac{\begin{array}{l}\Gamma \vdash_a arg : ty \Rightarrow ty' \\ \Gamma \vdash f : ty' \Rightarrow ty'' \end{array}}{\Gamma \vdash f\ arg : ty \Rightarrow ty''} \text{ TYPING\_TRHS\_F}$$

$$\frac{\begin{array}{l}\{l : ty\} \odot rty{=}rty' \\ \Gamma \vdash rty' \end{array}}{\Gamma \vdash var.l : rty' \Rightarrow ty} \text{ TYPING\_TRHS\_PROJECTION}$$

$$\frac{\begin{array}{l}\Gamma \vdash rty' \\ \{l_1 : ty_1\,;\;..\,;\;l_n : ty_n\} \odot rty{=}rty' \end{array}}{\Gamma \vdash \{var\ \texttt{with}\ l_1{=}var_1\,;\;..\,;\;l_n{=}var_n\} : \{var : rty'\,;\;var_1 : ty_1\,;\;..\,;\;var_n : ty_n\} \Rightarrow rty'} \text{ TYPING\_TRHS\_UPD}$$

$$\frac{\begin{array}{l}\{var : [constructor_1 : ty_1 \mid .. \mid constructor_k : ty_k]\} \odot rty{=}rty' \\ \{var_1 : ty_1\} \odot rty{=}rty_1 \quad .. \quad \{var_k : ty_k\} \odot rty{=}rty_k \\ \Gamma \vdash rhs_1 : rty_1 \Rightarrow ty \quad .. \quad \Gamma \vdash rhs_k : rty_k \Rightarrow ty \end{array}}{\Gamma \vdash \texttt{match}\ var\ \texttt{with}\ constructor_1\ var_1\ \texttt{->}\ rhs_1 \mid .. \mid constructor_k\ var_k\ \texttt{->}\ rhs_k\ \texttt{end} : rty' \Rightarrow ty} \text{ TYPING\_TRH}$$

$\boxed{\Gamma \vdash instruction : ty \Rightarrow ty'}$ Instruction typing

$$\frac{\begin{array}{l}\Gamma \vdash rty_1 \\ \Gamma \vdash rty_2 \\ rty \odot rty''{=}rty_1 \\ rty' \odot rty''{=}rty_2 \\ \Gamma \vdash instruction \ : \ rty \Rightarrow rty'\end{array}}{\Gamma \vdash instruction \ : \ rty_1 \Rightarrow rty_2} \ \textsc{typing\_T\_frame}$$

$$\frac{}{\Gamma \vdash \texttt{noop} \ : \ \{\,\} \Rightarrow \{\,\}} \ \textsc{typing\_T\_noop}$$

$$\frac{\begin{array}{c}\Gamma \vdash instruction \ : \ ty_1 \Rightarrow ty_2 \\ \Gamma \vdash instruction' \ : \ ty_2 \Rightarrow ty_3\end{array}}{\Gamma \vdash instruction\,;\ instruction' \ : \ ty_1 \Rightarrow ty_3} \ \textsc{typing\_T\_seq}$$

$$\frac{\begin{array}{c}\Gamma \vdash rhs \ : \ a \Rightarrow b \\ \Gamma \vdash lhs \ : \ b \Rightarrow c\end{array}}{\Gamma \vdash lhs{=}rhs \ : \ a \Rightarrow c} \ \textsc{typing\_T\_assign}$$

$$\frac{\begin{array}{l}\{var \ : \ [constructor_1 \ : \ ty_1 \ \texttt{|} \ .. \ \texttt{|} \ constructor_k \ : \ ty_k]\} \odot rty{=}rty' \\ \{var_1 \ : \ ty_1\} \odot rty{=}rty_1 \quad .. \quad \{var_k \ : \ ty_k\} \odot rty{=}rty_k \\ \Gamma \vdash I_1 \ : \ rty_1 \Rightarrow ty \quad .. \quad \Gamma \vdash I_k \ : \ rty_k \Rightarrow ty\end{array}}{\Gamma \vdash \texttt{match}\ var\ \texttt{with}\ constructor_1\ var_1\ \texttt{->}\ I_1\ \texttt{|}\ ..\ \texttt{|}\ constructor_k\ var_k\ \texttt{->}\ I_k\ \texttt{end} \ : \ rty' \Rightarrow ty} \ \textsc{typing\_T\_matc}$$

$\boxed{\Gamma \vdash value \ : \ ty}$ Value typing

$$\frac{\Gamma \vdash val_1 \ : \ ty_1 \quad .. \quad \Gamma \vdash val_n \ : \ ty_n}{\Gamma \vdash \{l_1{=}val_1\,;\ ..\,;\ l_n{=}val_n\} \ : \ \{l_1 \ : \ ty_1\,;\ ..\,;\ l_n \ : \ ty_n\}} \ \textsc{typing\_Tval\_record}$$

$$\frac{\begin{array}{c}\Gamma \vdash value \ : \ ty' \\ [constructor \ : \ ty'] \odot vty'{=}vty\end{array}}{\Gamma \vdash (constructor\ value \ : \ vty) \ : \ vty} \ \textsc{typing\_Tval\_constructor}$$

$\boxed{\Gamma \vdash f \ : \ ty \Rightarrow ty'}$ Function symbol typing

$$\frac{[constructor \ : \ ty'] \odot vty'{=}vty}{\Gamma \vdash (constructor \ : \ vty) \ : \ ty' \Rightarrow vty} \ \textsc{typing\_Tf\_constructor}$$

$\boxed{\texttt{E} \vdash lhs/val \ \texttt{->}\ val'}$ Left-hand side evaluation

$$\frac{}{\texttt{E} \vdash var/val \ \texttt{->}\ \{var{=}val\}} \ \textsc{eval\_lhs\_var}$$

$$\frac{}{\texttt{E} \vdash \{l_1{=}x_1\,;\ ..\,;\ l_n{=}x_n\}/\{l_1{=}val_1\,;\ ..\,;\ l_n{=}val_n\} \ \texttt{->}\ \{x_1{=}val_1\,;\ ..\,;\ x_n{=}val_n\}} \ \textsc{eval\_lhs\_record}$$

$\boxed{\texttt{E} \vdash arg/aval \ \texttt{->}\ val'}$ Argument evaluation

$$\frac{}{\texttt{E} \vdash var/a\{var{=}val\} \texttt{ -> } val} \quad \text{EVAL\_ARG\_VAR}$$

$$\frac{}{\texttt{E} \vdash val/a\{\,\} \texttt{ -> } val} \quad \text{EVAL\_ARG\_VAL}$$

$$\frac{}{\texttt{E} \vdash \{l_1{=}x_1\texttt{;}\ \ldots\texttt{;}\ l_n{=}x_n\}/a\{x_1{=}val_1\texttt{;}\ \ldots\texttt{;}\ x_n{=}val_n\} \texttt{ -> } \{l_1{=}val_1\texttt{;}\ \ldots\texttt{;}\ l_n{=}val_n\}} \quad \text{EVAL\_ARG\_RECORD}$$

$\boxed{\texttt{E} \vdash f/val \texttt{ -> } val'}$ Function symbol evaluation

$$\frac{}{\texttt{E} \vdash (constructor\ \texttt{:}\ vty)/val \texttt{ -> } (constructor\ val\ \texttt{:}\ vty)} \quad \text{EVAL\_F\_CONSTRUCTOR}$$

$\boxed{\texttt{E} \vdash rhs/val \texttt{ -> } val'}$ Right-hand side evaluation

$$\frac{\texttt{E} \vdash arg/aval \texttt{ -> } val'}{\texttt{E} \vdash arg/val \texttt{ -> } val'} \quad \text{EVAL\_RHS\_ARG}$$

$$\frac{\begin{array}{l}\texttt{E} \vdash arg/aval \texttt{ -> } val' \\ \texttt{E} \vdash f/val' \texttt{ -> } val''\end{array}}{\texttt{E} \vdash f\ arg/val \texttt{ -> } val''} \quad \text{EVAL\_RHS\_F}$$

$$\frac{\{l{=}val\} \odot rval{=}rval'}{\texttt{E} \vdash var.l/rval' \texttt{ -> } val} \quad \text{EVAL\_RHS\_PROJECTION}$$

$$\frac{\begin{array}{l}\{l_1{=}val'_1\texttt{;}\ \ldots\texttt{;}\ l_n{=}val'_n\} \odot rval{=}rval' \\ \{l_1{=}val_1\texttt{;}\ \ldots\texttt{;}\ l_n{=}val_n\} \odot rval{=}rval''\end{array}}{\texttt{E} \vdash \{var\ \texttt{with}\ l_1{=}var_1\texttt{;}\ \ldots\texttt{;}\ l_n{=}var_n\}/\{var{=}rval'\texttt{;}\ var_1{=}val_1\texttt{;}\ \ldots\texttt{;}\ var_n{=}val_n\} \texttt{ -> } rval''} \quad \text{EVAL\_RHS\_UPDATE}$$

$$\frac{\texttt{E} \vdash rhs/\{var'{=}val'\} \texttt{ -> } val}{\texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ constructor\ var' \texttt{ -> } rhs\ \texttt{|}\ <branches\_rhs>\ \texttt{end}/\{var{=}(constructor\ val'\ \texttt{:}\ vty)\} \texttt{ -> } val}$$

$$\frac{\begin{array}{l}constructor <> constructor' \\ \texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ branches\_rhs\ \texttt{end}/\{var{=}(constructor\ val'\ \texttt{:}\ vty)\} \texttt{ -> } val'\end{array}}{\texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ constructor'\ var' \texttt{ -> } rhs\ \texttt{|}\ <branches\_rhs>\ \texttt{end}/\{var{=}(constructor\ val'\ \texttt{:}\ vty)\} \texttt{ -> } val}$$

$\boxed{\texttt{E} \vdash instruction/val \texttt{ -> } val'}$ Instruction evaluation

$$\frac{\begin{array}{l}\texttt{E} \vdash instruction/rval \texttt{ -> } rval' \\ rval \odot rval''{=}rval_1 \\ rval' \odot rval''{=}rval_2\end{array}}{\texttt{E} \vdash instruction/rval_1 \texttt{ -> } rval_2} \quad \text{EVAL\_INSTR\_FRAME}$$

$$\frac{}{\texttt{E} \vdash \texttt{noop}/\{\,\} \texttt{ -> } \{\,\}} \quad \text{EVAL\_INSTR\_NOOP}$$

$$\frac{\texttt{E} \vdash I_1/val \texttt{ -> } val' \quad \texttt{E} \vdash I_2/val' \texttt{ -> } val''}{\texttt{E} \vdash I_1\texttt{;}\ I_2/val \texttt{ -> } val''}\ \textsc{eval\_instr\_seq}$$

$$\frac{\texttt{E} \vdash rhs/val \texttt{ -> } val' \quad \texttt{E} \vdash lhs/val' \texttt{ -> } val''}{\texttt{E} \vdash lhs\texttt{=}rhs/val \texttt{ -> } val''}\ \textsc{eval\_instr\_assign}$$

$$\frac{\begin{array}{l}\{var\texttt{=(}constructor\ val'\ \texttt{:}\ vty\texttt{)}\} \odot rval\texttt{=}rval' \\ \{var'\texttt{=}val'\} \odot rval\texttt{=}rval' \\ \texttt{E} \vdash I/rval' \texttt{ -> } rval_1\end{array}}{\texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ constructor\ var'\ \texttt{->}\ I\ \texttt{|}\ <branches\_ins>\ \texttt{end}/rval'\ \texttt{->}\ rval_1}\ \textsc{eval\_instr\_match\_f}$$

$$\frac{\begin{array}{l}\{var\texttt{=(}constructor\ val'\ \texttt{:}\ vty\texttt{)}\} \odot rval\texttt{=}rval' \\ constructor <> constructor' \\ \texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ branches\_ins\ \texttt{end}/rval'\ \texttt{->}\ rval_1\end{array}}{\texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ constructor'\ var'\ \texttt{->}\ I\ \texttt{|}\ <branches\_ins>\ \texttt{end}/rval'\ \texttt{->}\ rval_1}\ \textsc{eval\_instr\_match\_n}$$

The type $a + b$ corresponds to Michelson's `or a b` with constructors `Left` and `Right`. The type `option` a corresponds to Michelson's `option a` and has constructors `Some` and `None`. The type `bool` corresponds to Michelson's `bool` type and has constructors `True` and `False`.

n, m			
id			
k			
rty	::=		Record type
	|	$\{l_1 : ty_1\texttt{;}\ ..\texttt{;}\ l_n : ty_n\}$	
lhs	::=		Left-hand side of assignement
	|	var	
	|	$\{l_1\texttt{=}var_1\texttt{;}\ ..\texttt{;}\ l_n\texttt{=}var_n\}$	
arg	::=		Function argument
	|	var	
	|	$value$	
	|	$\{l_1\texttt{=}var_1\texttt{;}\ ...\texttt{;}\ l_n\texttt{=}var_n\}$	
$rval$	::=		Record value
	|	$\{l_1\texttt{=}value_1\texttt{;}\ ..\texttt{;}\ l_n\texttt{=}value_n\}$	Record
Γ	::=		Typing context (for inlined definitions)
	|	.	

products	::=		Products of types
	|	{ty_1 * ty'_1; .. ; ty_n * ty'_n}	
label, *l*, *var*, *x*	::=		Label / variable
	|	*id*	S
vty	::=		
	|	[$constructor_1$: ty_1 | ... | $constructor_k$: ty_k]	
f	::=		Function symbol
	|	(*constructor* : *vty*)	
branches_rhs	::=		
	|	$constructor_1$ var_1 -> rhs_1 | ... | $constructor_k$ var_k -> rhs_k	
rhs	::=		Right-hand side o
	|	*arg*	
	|	*f arg*	
	|	*var*.*l*	
	|	{*var* with l_1=var_1; ... ; l_n=var_n}	
	|	match *var* with *branches_rhs* end	
branches_ins	::=		
	|	$constructor_1$ var_1 -> ins_1 | ... | $constructor_k$ var_k -> ins_k	
value, *val*	::=		Value
	|	*rval*	Record
	|	(*value*)	S
	|	(*constructor value* : *vty*)	
constructor	::=		
	|	*id*	S
ty, *a*, *b*, *c*	::=		Type
	|	*rty*	Record type
	|	(*ty*)	S Parenthesised t
	|	*vty*	

	| a `+` b	
instruction, I, *ins*	::=	Instruction
	| `noop`	No operation
	| *instruction*$_1$ `;` *instruction*$_2$	Sequencing
	| *lhs*`=`*rhs*	Assignment
	| `match` *var* `with` *branches*`_`*ins* `end`	
	| `loop_left` *var* `in` *var*$'$ `do` *instruction* `done`	

$\boxed{\Gamma \vdash}$ Context well-fornedness

$$\frac{}{. \vdash}\ \textsc{typing\_Gwf\_empty}$$

$\boxed{\Gamma \vdash lhs : ty \Rightarrow ty'}$ Left-hand sides typing

$$\frac{}{\Gamma \vdash var : ty \Rightarrow \{var : ty\}}\ \textsc{typing\_Tlhs\_var}$$

$$\frac{}{\Gamma \vdash \{l_1{=}x_1\,;\ ..\ ;\ l_n{=}x_n\} : \{l_1 : ty_1\,;\ ..\ ;\ l_n : ty_n\} \Rightarrow \{x_1 : ty_1\,;\ ..\ ;\ x_n : ty_n\}}\ \textsc{typing\_Tlhs\_record}$$

$\boxed{\Gamma \vdash_a arg : ty \Rightarrow ty'}$ Argument typing

$$\frac{}{\Gamma \vdash_a var : \{var : ty\} \Rightarrow ty}\ \textsc{typing\_Targ\_var}$$

$$\frac{\Gamma \vdash value : ty}{\Gamma \vdash_a value : \{\,\} \Rightarrow ty}\ \textsc{typing\_Targ\_value}$$

$$\frac{}{\Gamma \vdash_a \{l_1{=}x_1\,;\ ..\ ;\ l_n{=}x_n\} : \{x_1 : ty_1\,;\ ..\ ;\ x_n : ty_n\} \Rightarrow \{l_1 : ty_1\,;\ ..\ ;\ l_n : ty_n\}}\ \textsc{typing\_Targ\_record}$$

$\boxed{\Gamma \vdash rhs : ty \Rightarrow ty'}$ Right-hand side typing

$$\frac{\Gamma \vdash_a arg : ty \Rightarrow ty'}{\Gamma \vdash arg : ty \Rightarrow ty'}\ \textsc{typing\_Trhs\_arg}$$

$$\frac{\Gamma \vdash_a arg : ty \Rightarrow ty' \qquad \Gamma \vdash f : ty' \Rightarrow ty''}{\Gamma \vdash f\ arg : ty \Rightarrow ty''}\ \textsc{typing\_Trhs\_f}$$

$$\frac{\{l : ty\} \odot rty{=}rty' \qquad \Gamma \vdash rty'}{\Gamma \vdash var.l : rty' \Rightarrow ty}\ \textsc{typing\_Trhs\_projection}$$

$$\frac{\begin{array}{c}\Gamma \vdash rty' \\ \{l_1 : ty_1;\ ..;\ l_n : ty_n\} \odot rty{=}rty'\end{array}}{\Gamma \vdash \{var\ \texttt{with}\ l_1{=}var_1;\ ..;\ l_n{=}var_n\} : \{var : rty';\ var_1 : ty_1;\ ..;\ var_n : ty_n\} \Rightarrow rty'}\ \textsc{typing\_Trhs\_upd}$$

$$\frac{\begin{array}{c}\{var : [constructor_1 : ty_1 \mid .. \mid constructor_k : ty_k]\} \odot rty{=}rty' \\ \{var_1 : ty_1\} \odot rty{=}rty_1 \quad .. \quad \{var_k : ty_k\} \odot rty{=}rty_k \\ \Gamma \vdash rhs_1 : rty_1 \Rightarrow ty \quad .. \quad \Gamma \vdash rhs_k : rty_k \Rightarrow ty\end{array}}{\Gamma \vdash \texttt{match}\ var\ \texttt{with}\ constructor_1\ var_1\ \texttt{->}\ rhs_1 \mid .. \mid constructor_k\ var_k\ \texttt{->}\ rhs_k\ \texttt{end} : rty' \Rightarrow ty}\ \textsc{typing\_Trh}$$

$\boxed{\Gamma \vdash value : ty}$ Value typing

$$\frac{\Gamma \vdash val_1 : ty_1 \quad .. \quad \Gamma \vdash val_n : ty_n}{\Gamma \vdash \{l_1{=}val_1;\ ..;\ l_n{=}val_n\} : \{l_1 : ty_1;\ ..;\ l_n : ty_n\}}\ \textsc{typing\_Tval\_record}$$

$$\frac{\begin{array}{c}\Gamma \vdash value : ty' \\ [constructor : ty'] \odot vty'{=}vty\end{array}}{\Gamma \vdash (constructor\ value : vty) : vty}\ \textsc{typing\_Tval\_constructor}$$

$\boxed{\Gamma \vdash f : ty \Rightarrow ty'}$ Function symbol typing

$$\frac{[constructor : ty'] \odot vty'{=}vty}{\Gamma \vdash (constructor : vty) : ty' \Rightarrow vty}\ \textsc{typing\_Tf\_constructor}$$

$\boxed{\Gamma \vdash ty_1 \equiv ty_2}$ Type equality

$$\frac{}{\Gamma \vdash ty \equiv ty}\ \textsc{typing\_Teq\_refl}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_2}{\Gamma \vdash ty_2 \equiv ty_1}\ \textsc{typing\_Teq\_sym}$$

$$\frac{\begin{array}{c}\Gamma \vdash ty_1 \equiv ty_2 \\ \Gamma \vdash ty_2 \equiv ty_3\end{array}}{\Gamma \vdash ty_1 \equiv ty_3}\ \textsc{typing\_Teq\_trans}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty'_1 \quad .. \quad \Gamma \vdash ty_n \equiv ty'_n}{\Gamma \vdash \{l_1 : ty_1;\ ..;\ l_n : ty_n\} \equiv \{l_1 : ty'_1;\ ..;\ l_n : ty'_n\}}\ \textsc{typing\_Teq\_congr}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty'_1 \quad .. \quad \Gamma \vdash ty_k \equiv ty'_k}{\Gamma \vdash [constructor_1 : ty_1 \mid .. \mid constructor_k : ty_k] \equiv [constructor_1 : ty'_1 \mid .. \mid constructor_k : ty'_k]}\ \textsc{typing}$$

$$\frac{}{\Gamma \vdash ty + ty' \equiv [\texttt{Left} : ty \mid \texttt{Right} : ty']}\ \textsc{typing\_Teq\_or}$$

$\boxed{\Gamma \vdash ty}$ Type well-formedness

$$\frac{\begin{array}{l}\texttt{sorted}\{l_1\,;\;..\,;\;l_n\}\\ \Gamma \vdash ty_1 \quad .. \quad \Gamma \vdash ty_n\\ \Gamma \vdash\end{array}}{\Gamma \vdash \{l_1 \,:\, ty_1\,;\;..\,;\;l_n \,:\, ty_n\}}\ \textsc{typing\_twf\_record}$$

$$\frac{\begin{array}{l}\texttt{sorted}\{constructor_1\,;\;..\,;\;constructor_k\}\\ \Gamma \vdash ty_1 \quad .. \quad \Gamma \vdash ty_k\\ \Gamma \vdash\end{array}}{\Gamma \vdash [constructor_1 \,:\, ty_1 \mid .. \mid constructor_k \,:\, ty_k]}\ \textsc{typing\_twf\_variant}$$

$$\frac{\begin{array}{l}\Gamma \vdash ty\\ \Gamma \vdash ty'\end{array}}{\Gamma \vdash ty \texttt{ + } ty'}\ \textsc{typing\_twf\_or}$$

$\boxed{\texttt{tvar} \notin \mathsf{FV}(ty)}$ Type variable freshness

$$\frac{\texttt{tvar} \notin \mathsf{FV}(ty_1) \quad .. \quad \texttt{tvar} \notin \mathsf{FV}(ty_n)}{\texttt{tvar} \notin \mathsf{FV}(\{l_1 \,:\, ty_1\,;\;..\,;\;l_n \,:\, ty_n\})}\ \textsc{typing\_Tfresh\_tvarrecord}$$

$$\frac{\texttt{tvar} \notin \mathsf{FV}(ty_1) \quad .. \quad \texttt{tvar} \notin \mathsf{FV}(ty_k)}{\texttt{tvar} \notin \mathsf{FV}([constructor_1 \,:\, ty_1 \mid .. \mid constructor_k \,:\, ty_k])}\ \textsc{typing\_Tfresh\_tvarvariant}$$

$$\frac{\begin{array}{l}\texttt{tvar} \notin \mathsf{FV}(ty)\\ \texttt{tvar} \notin \mathsf{FV}(ty')\end{array}}{\texttt{tvar} \notin \mathsf{FV}(ty \texttt{ + } ty')}\ \textsc{typing\_Tfresh\_tvaror}$$

$\boxed{\Gamma \vdash instruction \,:\, ty \Rightarrow ty'}$ Instruction typing

$$\frac{\begin{array}{l}\Gamma \vdash rty_1\\ \Gamma \vdash rty_2\\ rty \odot rty'' {=} rty_1\\ rty' \odot rty'' {=} rty_2\\ \Gamma \vdash instruction \,:\, rty \Rightarrow rty'\end{array}}{\Gamma \vdash instruction \,:\, rty_1 \Rightarrow rty_2}\ \textsc{typing\_T\_frame}$$

$$\frac{}{\Gamma \vdash \texttt{noop} \,:\, \{\,\} \Rightarrow \{\,\}}\ \textsc{typing\_T\_noop}$$

$$\frac{\begin{array}{l}\Gamma \vdash instruction \,:\, ty_1 \Rightarrow ty_2\\ \Gamma \vdash instruction' \,:\, ty_2 \Rightarrow ty_3\end{array}}{\Gamma \vdash instruction\,;\; instruction' \,:\, ty_1 \Rightarrow ty_3}\ \textsc{typing\_T\_seq}$$

$$\frac{\begin{array}{l}\Gamma \vdash rhs \,:\, a \Rightarrow b\\ \Gamma \vdash lhs \,:\, b \Rightarrow c\end{array}}{\Gamma \vdash lhs{=}rhs \,:\, a \Rightarrow c}\ \textsc{typing\_T\_assign}$$

$$\frac{\begin{array}{l}\{var\ :\ [constructor_1\ :\ ty_1\ |\ ..\ |\ constructor_k\ :\ ty_k]\} \odot rty{=}rty' \\ \{var_1\ :\ ty_1\} \odot rty{=}rty_1 \quad .. \quad \{var_k\ :\ ty_k\} \odot rty{=}rty_k \\ \Gamma \vdash I_1\ :\ rty_1 \Rightarrow ty \quad .. \quad \Gamma \vdash I_k\ :\ rty_k \Rightarrow ty\end{array}}{\Gamma \vdash \texttt{match}\ var\ \texttt{with}\ constructor_1\ var_1\ \texttt{->}\ I_1\ \texttt{|}\ ..\ \texttt{|}\ constructor_k\ var_k\ \texttt{->}\ I_k\ \texttt{end}\ :\ rty' \Rightarrow ty}\ \text{TYPING\_T\_MATC}$$

$$\frac{\begin{array}{l}\{var'\ :\ a\ \texttt{+}\ b\} \odot rty{=}rty' \\ \{var\ :\ a\} \odot rty{=}rty'' \\ \Gamma \vdash I\ :\ rty'' \Rightarrow rty'\end{array}}{\Gamma \vdash \texttt{loop\_left}\ var\ \texttt{in}\ var'\ \texttt{do}\ I\ \texttt{done}\ :\ rty' \Rightarrow rty}\ \text{TYPING\_T\_LOOP\_LEFT}$$

$\boxed{\texttt{E} \vdash lhs/val\ \texttt{->}\ val'}$ Left-hand side evaluation

$$\frac{}{\texttt{E} \vdash var/val\ \texttt{->}\ \{var{=}val\}}\ \text{EVAL\_LHS\_VAR}$$

$$\frac{}{\texttt{E} \vdash \{l_1{=}x_1\,;\ ..\ ;\ l_n{=}x_n\}/\{l_1{=}val_1\,;\ ..\ ;\ l_n{=}val_n\}\ \texttt{->}\ \{x_1{=}val_1\,;\ ..\ ;\ x_n{=}val_n\}}\ \text{EVAL\_LHS\_RECORD}$$

$\boxed{\texttt{E} \vdash arg/aval\ \texttt{->}\ val'}$ Argument evaluation

$$\frac{}{\texttt{E} \vdash var/a\{var{=}val\}\ \texttt{->}\ val}\ \text{EVAL\_ARG\_VAR}$$

$$\frac{}{\texttt{E} \vdash val/a\{\}\ \texttt{->}\ val}\ \text{EVAL\_ARG\_VAL}$$

$$\frac{}{\texttt{E} \vdash \{l_1{=}x_1\,;\ ..\ ;\ l_n{=}x_n\}/a\{x_1{=}val_1\,;\ ..\ ;\ x_n{=}val_n\}\ \texttt{->}\ \{l_1{=}val_1\,;\ ..\ ;\ l_n{=}val_n\}}\ \text{EVAL\_ARG\_RECORD}$$

$\boxed{\texttt{E} \vdash f/val\ \texttt{->}\ val'}$ Function symbol evaluation

$$\frac{}{\texttt{E} \vdash (constructor\ :\ vty)/val\ \texttt{->}\ (constructor\ val\ :\ vty)}\ \text{EVAL\_F\_CONSTRUCTOR}$$

$\boxed{\texttt{E} \vdash rhs/val\ \texttt{->}\ val'}$ Right-hand side evaluation

$$\frac{\texttt{E} \vdash arg/aval\ \texttt{->}\ val'}{\texttt{E} \vdash arg/val\ \texttt{->}\ val'}\ \text{EVAL\_RHS\_ARG}$$

$$\frac{\begin{array}{l}\texttt{E} \vdash arg/aval\ \texttt{->}\ val' \\ \texttt{E} \vdash f/val'\ \texttt{->}\ val''\end{array}}{\texttt{E} \vdash f\ arg/val\ \texttt{->}\ val''}\ \text{EVAL\_RHS\_F}$$

$$\frac{\{l{=}val\} \odot rval{=}rval'}{\texttt{E} \vdash var.l/rval'\ \texttt{->}\ val}\ \text{EVAL\_RHS\_PROJECTION}$$

$$\frac{\begin{array}{l}\{l_1{=}val_1'\,;\ ..\ ;\ l_n{=}val_n'\} \odot rval{=}rval' \\ \{l_1{=}val_1\,;\ ..\ ;\ l_n{=}val_n\} \odot rval{=}rval''\end{array}}{\texttt{E} \vdash \{var\ \texttt{with}\ l_1{=}var_1\,;\ ..\ ;\ l_n{=}var_n\}/\{var{=}rval'\,;\ var_1{=}val_1\,;\ ..\ ;\ var_n{=}val_n\}\ \texttt{->}\ rval''}\ \text{EVAL\_RHS\_UPDATE}$$

$$\frac{\texttt{E} \vdash rhs/\{var'=val'\}\ \texttt{->}\ val}{\texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ constructor\ var'\ \texttt{->}\ rhs\ \texttt{|}\ <\ branches\_rhs\ >\ \texttt{end}/\{var=(constructor\ val'\ :\ vty)\}\ \texttt{->}\ val}$$

$$\frac{\begin{array}{l} constructor <> constructor' \\ \texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ branches\_rhs\ \texttt{end}/\{var=(constructor\ val'\ :\ vty)\}\ \texttt{->}\ val' \end{array}}{\texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ constructor'\ var'\ \texttt{->}\ rhs\ \texttt{|}\ <\ branches\_rhs\ >\ \texttt{end}/\{var=(constructor\ val'\ :\ vty)\}\ \texttt{->}\ val}$$

$\boxed{\texttt{E} \vdash instruction/val\ \texttt{->}\ val'}$ Instruction evaluation

$$\frac{\begin{array}{l} \texttt{E} \vdash instruction/rval\ \texttt{->}\ rval' \\ rval \odot rval''=rval_1 \\ rval' \odot rval''=rval_2 \end{array}}{\texttt{E} \vdash instruction/rval_1\ \texttt{->}\ rval_2}\ \text{EVAL\_INSTR\_FRAME}$$

$$\frac{}{\texttt{E} \vdash \texttt{noop}/\{\}\ \texttt{->}\ \{\}}\ \text{EVAL\_INSTR\_NOOP}$$

$$\frac{\begin{array}{l} \texttt{E} \vdash I_1/val\ \texttt{->}\ val' \\ \texttt{E} \vdash I_2/val'\ \texttt{->}\ val'' \end{array}}{\texttt{E} \vdash I_1\texttt{;}\ I_2/val\ \texttt{->}\ val''}\ \text{EVAL\_INSTR\_SEQ}$$

$$\frac{\begin{array}{l} \texttt{E} \vdash rhs/val\ \texttt{->}\ val' \\ \texttt{E} \vdash lhs/val'\ \texttt{->}\ val'' \end{array}}{\texttt{E} \vdash lhs\texttt{=}rhs/val\ \texttt{->}\ val''}\ \text{EVAL\_INSTR\_ASSIGN}$$

$$\frac{\begin{array}{l} \{var=(constructor\ val'\ :\ vty)\} \odot rval=rval' \\ \{var'=val'\} \odot rval=rval' \\ \texttt{E} \vdash I/rval'\ \texttt{->}\ rval_1 \end{array}}{\texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ constructor\ var'\ \texttt{->}\ I\ \texttt{|}\ <\ branches\_ins\ >\ \texttt{end}/rval'\ \texttt{->}\ rval_1}\ \text{EVAL\_INSTR\_MATCH\_F}$$

$$\frac{\begin{array}{l} \{var=(constructor\ val'\ :\ vty)\} \odot rval=rval' \\ constructor <> constructor' \\ \texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ branches\_ins\ \texttt{end}/rval'\ \texttt{->}\ rval_1 \end{array}}{\texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ constructor'\ var'\ \texttt{->}\ I\ \texttt{|}\ <\ branches\_ins\ >\ \texttt{end}/rval'\ \texttt{->}\ rval_1}\ \text{EVAL\_INSTR\_MATCH\_N}$$

$$\frac{\{var'=(\texttt{Right}\ val\ :\ vty)\} \odot rval=rval'}{\texttt{E} \vdash \texttt{loop\_left}\ var\ \texttt{in}\ var'\ \texttt{do}\ I\ \texttt{done}/rval'\ \texttt{->}\ rval}\ \text{EVAL\_INSTR\_LOOP\_LEFT\_END}$$

$$\frac{\begin{array}{l} \{var'=(\texttt{Left}\ val\ :\ vty)\} \odot rval=rval' \\ \{var=val\} \odot rval=rval'' \\ \texttt{E} \vdash I/rval''\ \texttt{->}\ rval_1 \\ \texttt{E} \vdash \texttt{loop\_left}\ var\ \texttt{in}\ var'\ \texttt{do}\ I\ \texttt{done}/rval_1\ \texttt{->}\ rval_2 \end{array}}{\texttt{E} \vdash \texttt{loop\_left}\ var\ \texttt{in}\ var'\ \texttt{do}\ I\ \texttt{done}/rval'\ \texttt{->}\ rval_2}\ \text{EVAL\_INSTR\_LOOP\_LEFT\_CONTINUE}$$

n, m			
id			
k			
rty	::=		Record type
	|	$\{l_1 : ty_1 ; \ldots ; l_n : ty_n\}$	
lhs	::=		Left-hand side of
	|	var	
	|	$\{l_1{=}var_1 ; \ldots ; l_n{=}var_n\}$	
arg	::=		Function argumen
	|	var	
	|	$value$	
	|	$\{l_1{=}var_1 ; \ldots ; l_n{=}var_n\}$	
$rval$	::=		Record value
	|	$\{l_1{=}value_1 ; \ldots ; l_n{=}value_n\}$	Record
Γ	::=		Typing context (f
	|	.	
$products$	::=		Products of types
	|	$\{ty_1 * ty_1' ; \ldots ; ty_n * ty_n'\}$	
$label$, l, var, x	::=		Label / variable
	|	id	S
vty	::=		
	|	$[constructor_1 : ty_1 \mid \ldots \mid constructor_k : ty_k]$	
f	::=		Function symbol
	|	$(constructor : vty)$	
$branches\_rhs$	::=		
	|	$constructor_1\ var_1 \texttt{ -> } rhs_1 \mid \ldots \mid constructor_k\ var_k \texttt{ -> } rhs_k$	
rhs	::=		Right-hand side o
	|	arg	
	|	$f\ arg$	
	|	$var.l$	

	|	{*var* `with` l_1=var_1 ; ... ; l_n=var_n}	
	|	`match` *var* `with` *branches_rhs* `end`	
branches_ins	::=		
	|	$constructor_1$ var_1 `->` ins_1 `|` ... `|` $constructor_k$ var_k `->` ins_k	
instruction, *I*, *ins*	::=		Instruction
	|	`noop`	No operatio
	|	$instruction_1$; $instruction_2$	Sequencing
	|	*lhs*=*rhs*	Assignment
	|	`match` *var* `with` *branches_ins* `end`	
value, *val*	::=		Value
	|	*rval*	Record
	|	(*value*)	S
	|	(*constructor value* : *vty*)	
constructor	::=		
	|	*id*	S
ty, *a*, *b*, *c*	::=		Type
	|	*rty*	Record type
	|	(*ty*)	S Parenthesise
	|	`unit`	
	|	*vty*	
	|	`option` *ty*	

$\boxed{\texttt{E} \vdash lhs/val \texttt{ -> } val'}$ Left-hand side evaluation

$$\frac{}{\texttt{E} \vdash var/val \texttt{ -> } \{var=val\}}\ \text{EVAL\_LHS\_VAR}$$

$$\frac{}{\texttt{E} \vdash \{l_1=x_1;\ ..;\ l_n=x_n\}/\{l_1=val_1;\ ..;\ l_n=val_n\} \texttt{ -> } \{x_1=val_1;\ ..;\ x_n=val_n\}}\ \text{EVAL\_LHS\_RECORD}$$

$\boxed{\texttt{E} \vdash arg/aval \texttt{ -> } val'}$ Argument evaluation

$$\frac{}{\texttt{E} \vdash var/a\{var=val\} \texttt{ -> } val}\ \text{EVAL\_ARG\_VAR}$$

$$\frac{}{\texttt{E} \vdash val/a\{\} \texttt{ -> } val}\ \text{EVAL\_ARG\_VAL}$$

$$\frac{}{\texttt{E} \vdash \{l_1=x_1;\ ..;\ l_n=x_n\}/a\{x_1=val_1;\ ..;\ x_n=val_n\} \texttt{ -> } \{l_1=val_1;\ ..;\ l_n=val_n\}}\ \text{EVAL\_ARG\_RECORD}$$

$\boxed{\texttt{E} \vdash f/val \texttt{ -> } val'}$ Function symbol evaluation

$$\frac{}{\texttt{E} \vdash (constructor \texttt{ : } vty)/val \texttt{ -> } (constructor\ val \texttt{ : } vty)} \text{ EVAL\_F\_CONSTRUCTOR}$$

$\boxed{\texttt{E} \vdash rhs/val \texttt{ -> } val'}$ Right-hand side evaluation

$$\frac{\texttt{E} \vdash arg/aval \texttt{ -> } val'}{\texttt{E} \vdash arg/val \texttt{ -> } val'} \text{ EVAL\_RHS\_ARG}$$

$$\frac{\begin{array}{l}\texttt{E} \vdash arg/aval \texttt{ -> } val' \\ \texttt{E} \vdash f/val' \texttt{ -> } val''\end{array}}{\texttt{E} \vdash f\ arg/val \texttt{ -> } val''} \text{ EVAL\_RHS\_F}$$

$$\frac{\texttt{\{}l\texttt{=}val\texttt{\}} \odot rval\texttt{=}rval'}{\texttt{E} \vdash var.l/rval' \texttt{ -> } val} \text{ EVAL\_RHS\_PROJECTION}$$

$$\frac{\begin{array}{l}\texttt{\{}l_1\texttt{=}val'_1\texttt{;}\ ..\ \texttt{;}\ l_n\texttt{=}val'_n\texttt{\}} \odot rval\texttt{=}rval' \\ \texttt{\{}l_1\texttt{=}val_1\texttt{;}\ ..\ \texttt{;}\ l_n\texttt{=}val_n\texttt{\}} \odot rval\texttt{=}rval''\end{array}}{\texttt{E} \vdash \texttt{\{}var\ \texttt{with}\ l_1\texttt{=}var_1\texttt{;}\ ..\ \texttt{;}\ l_n\texttt{=}var_n\texttt{\}/\{}var\texttt{=}rval'\texttt{;}\ var_1\texttt{=}val_1\texttt{;}\ ..\ \texttt{;}\ var_n\texttt{=}val_n\texttt{\}} \texttt{ -> } rval''} \text{ EVAL\_RHS\_UPDATE}$$

$$\frac{\texttt{E} \vdash rhs/\texttt{\{}var'\texttt{=}val'\texttt{\}} \texttt{ -> } val}{\texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ constructor\ var' \texttt{ -> } rhs \texttt{ | } < branches\_rhs > \texttt{end/\{}var\texttt{=(}constructor\ val' \texttt{ : } vty\texttt{)\}} \texttt{ -> } val}$$

$$\frac{\begin{array}{l}constructor <> constructor' \\ \texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ branches\_rhs\ \texttt{end/\{}var\texttt{=(}constructor\ val' \texttt{ : } vty\texttt{)\}} \texttt{ -> } val'\end{array}}{\texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ constructor'\ var' \texttt{ -> } rhs \texttt{ | } < branches\_rhs > \texttt{end/\{}var\texttt{=(}constructor\ val' \texttt{ : } vty\texttt{)\}} \texttt{ -> } val}$$

$\boxed{\texttt{E} \vdash instruction/val \texttt{ -> } val'}$ Instruction evaluation

$$\frac{\begin{array}{l}\texttt{E} \vdash instruction/rval \texttt{ -> } rval' \\ rval \odot rval''\texttt{=}rval_1 \\ rval' \odot rval''\texttt{=}rval_2\end{array}}{\texttt{E} \vdash instruction/rval_1 \texttt{ -> } rval_2} \text{ EVAL\_INSTR\_FRAME}$$

$$\frac{}{\texttt{E} \vdash \texttt{noop/\{\} -> \{\}}} \text{ EVAL\_INSTR\_NOOP}$$

$$\frac{\begin{array}{l}\texttt{E} \vdash I_1/val \texttt{ -> } val' \\ \texttt{E} \vdash I_2/val' \texttt{ -> } val''\end{array}}{\texttt{E} \vdash I_1\texttt{;}\ I_2/val \texttt{ -> } val''} \text{ EVAL\_INSTR\_SEQ}$$

$$\frac{\begin{array}{l}\texttt{E} \vdash rhs/val \texttt{ -> } val' \\ \texttt{E} \vdash lhs/val' \texttt{ -> } val''\end{array}}{\texttt{E} \vdash lhs\texttt{=}rhs/val \texttt{ -> } val''} \text{ EVAL\_INSTR\_ASSIGN}$$

$$\frac{\begin{array}{l}\texttt{\{}var\texttt{=(}constructor\ val'\ \texttt{:}\ vty\texttt{)\}} \odot rval\texttt{=}rval' \\ \texttt{\{}var'\texttt{=}val'\texttt{\}} \odot rval\texttt{=}rval' \\ \texttt{E} \vdash I/rval'\ \texttt{->}\ rval_1\end{array}}{\texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ constructor\ var'\ \texttt{->}\ I\ \texttt{|}\ <branches\_ins>\ \texttt{end}/rval'\ \texttt{->}\ rval_1}\ \textsc{eval\_instr\_match\_f}$$

$$\frac{\begin{array}{l}\texttt{\{}var\texttt{=(}constructor\ val'\ \texttt{:}\ vty\texttt{)\}} \odot rval\texttt{=}rval' \\ constructor <> constructor' \\ \texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ branches\_ins\ \texttt{end}/rval'\ \texttt{->}\ rval_1\end{array}}{\texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ constructor'\ var'\ \texttt{->}\ I\ \texttt{|}\ <branches\_ins>\ \texttt{end}/rval'\ \texttt{->}\ rval_1}\ \textsc{eval\_instr\_match\_n}$$

$\boxed{\Gamma \vdash}$ Context well-fornedness

$$\frac{}{. \vdash}\ \textsc{typing\_Gwf\_empty}$$

$\boxed{\Gamma \vdash lhs\ \texttt{:}\ ty \Rightarrow ty'}$ Left-hand sides typing

$$\frac{}{\Gamma \vdash var\ \texttt{:}\ ty \Rightarrow \texttt{\{}var\ \texttt{:}\ ty\texttt{\}}}\ \textsc{typing\_Tlhs\_var}$$

$$\frac{}{\Gamma \vdash \texttt{\{}l_1\texttt{=}x_1\texttt{;}\ ..\ \texttt{;}\ l_n\texttt{=}x_n\texttt{\}}\ \texttt{:}\ \texttt{\{}l_1\ \texttt{:}\ ty_1\texttt{;}\ ..\ \texttt{;}\ l_n\ \texttt{:}\ ty_n\texttt{\}} \Rightarrow \texttt{\{}x_1\ \texttt{:}\ ty_1\texttt{;}\ ..\ \texttt{;}\ x_n\ \texttt{:}\ ty_n\texttt{\}}}\ \textsc{typing\_Tlhs\_record}$$

$\boxed{\Gamma \vdash_a arg\ \texttt{:}\ ty \Rightarrow ty'}$ Argument typing

$$\frac{}{\Gamma \vdash_a var\ \texttt{:}\ \texttt{\{}var\ \texttt{:}\ ty\texttt{\}} \Rightarrow ty}\ \textsc{typing\_Targ\_var}$$

$$\frac{\Gamma \vdash value\ \texttt{:}\ ty}{\Gamma \vdash_a value\ \texttt{:}\ \texttt{\{\}} \Rightarrow ty}\ \textsc{typing\_Targ\_value}$$

$$\frac{}{\Gamma \vdash_a \texttt{\{}l_1\texttt{=}x_1\texttt{;}\ ..\ \texttt{;}\ l_n\texttt{=}x_n\texttt{\}}\ \texttt{:}\ \texttt{\{}x_1\ \texttt{:}\ ty_1\texttt{;}\ ..\ \texttt{;}\ x_n\ \texttt{:}\ ty_n\texttt{\}} \Rightarrow \texttt{\{}l_1\ \texttt{:}\ ty_1\texttt{;}\ ..\ \texttt{;}\ l_n\ \texttt{:}\ ty_n\texttt{\}}}\ \textsc{typing\_Targ\_record}$$

$\boxed{\Gamma \vdash rhs\ \texttt{:}\ ty \Rightarrow ty'}$ Right-hand side typing

$$\frac{\Gamma \vdash_a arg\ \texttt{:}\ ty \Rightarrow ty'}{\Gamma \vdash arg\ \texttt{:}\ ty \Rightarrow ty'}\ \textsc{typing\_Trhs\_arg}$$

$$\frac{\begin{array}{l}\Gamma \vdash_a arg\ \texttt{:}\ ty \Rightarrow ty' \\ \Gamma \vdash f\ \texttt{:}\ ty' \Rightarrow ty''\end{array}}{\Gamma \vdash f\ arg\ \texttt{:}\ ty \Rightarrow ty''}\ \textsc{typing\_Trhs\_f}$$

$$\frac{\begin{array}{l}\texttt{\{}l\ \texttt{:}\ ty\texttt{\}} \odot rty\texttt{=}rty' \\ \Gamma \vdash rty'\end{array}}{\Gamma \vdash var.l\ \texttt{:}\ rty' \Rightarrow ty}\ \textsc{typing\_Trhs\_projection}$$

$$\frac{\begin{array}{l}\Gamma \vdash rty' \\ \{l_1 \,:\, ty_1\,;\ ..\ ;\ l_n \,:\, ty_n\} \odot rty{=}rty'\end{array}}{\Gamma \vdash \{var\ \texttt{with}\ l_1{=}var_1\,;\ ..\ ;\ l_n{=}var_n\} \,:\, \{var \,:\, rty';\ var_1 \,:\, ty_1\,;\ ..\ ;\ var_n \,:\, ty_n\} \Rightarrow rty'}\ \textsc{typing\_Trhs\_upd}$$

$$\frac{\begin{array}{l}\{var \,:\, [constructor_1 \,:\, ty_1 \mid ..\ \mid constructor_k \,:\, ty_k]\} \odot rty{=}rty' \\ \{var_1 \,:\, ty_1\} \odot rty{=}rty_1 \quad .. \quad \{var_k \,:\, ty_k\} \odot rty{=}rty_k \\ \Gamma \vdash rhs_1 \,:\, rty_1 \Rightarrow ty \quad .. \quad \Gamma \vdash rhs_k \,:\, rty_k \Rightarrow ty\end{array}}{\Gamma \vdash \texttt{match}\ var\ \texttt{with}\ constructor_1\ var_1\ \texttt{->}\ rhs_1\ \mid\ ..\ \mid\ constructor_k\ var_k\ \texttt{->}\ rhs_k\ \texttt{end} \,:\, rty' \Rightarrow ty}\ \textsc{typing\_Trh}$$

$\boxed{\Gamma \vdash instruction \,:\, ty \Rightarrow ty'}$ Instruction typing

$$\frac{\begin{array}{l}\Gamma \vdash rty_1 \\ \Gamma \vdash rty_2 \\ rty \odot rty''{=}rty_1 \\ rty' \odot rty''{=}rty_2 \\ \Gamma \vdash instruction \,:\, rty \Rightarrow rty'\end{array}}{\Gamma \vdash instruction \,:\, rty_1 \Rightarrow rty_2}\ \textsc{typing\_T\_frame}$$

$$\frac{}{\Gamma \vdash \texttt{noop} \,:\, \{\,\} \Rightarrow \{\,\}}\ \textsc{typing\_T\_noop}$$

$$\frac{\begin{array}{l}\Gamma \vdash instruction \,:\, ty_1 \Rightarrow ty_2 \\ \Gamma \vdash instruction' \,:\, ty_2 \Rightarrow ty_3\end{array}}{\Gamma \vdash instruction\,;\ instruction' \,:\, ty_1 \Rightarrow ty_3}\ \textsc{typing\_T\_seq}$$

$$\frac{\begin{array}{l}\Gamma \vdash rhs \,:\, a \Rightarrow b \\ \Gamma \vdash lhs \,:\, b \Rightarrow c\end{array}}{\Gamma \vdash lhs{=}rhs \,:\, a \Rightarrow c}\ \textsc{typing\_T\_assign}$$

$$\frac{\begin{array}{l}\{var \,:\, [constructor_1 \,:\, ty_1 \mid ..\ \mid constructor_k \,:\, ty_k]\} \odot rty{=}rty' \\ \{var_1 \,:\, ty_1\} \odot rty{=}rty_1 \quad .. \quad \{var_k \,:\, ty_k\} \odot rty{=}rty_k \\ \Gamma \vdash I_1 \,:\, rty_1 \Rightarrow ty \quad .. \quad \Gamma \vdash I_k \,:\, rty_k \Rightarrow ty\end{array}}{\Gamma \vdash \texttt{match}\ var\ \texttt{with}\ constructor_1\ var_1\ \texttt{->}\ I_1\ \mid\ ..\ \mid\ constructor_k\ var_k\ \texttt{->}\ I_k\ \texttt{end} \,:\, rty' \Rightarrow ty}\ \textsc{typing\_T\_matc}$$

$\boxed{\Gamma \vdash value \,:\, ty}$ Value typing

$$\frac{\Gamma \vdash val_1 \,:\, ty_1 \quad .. \quad \Gamma \vdash val_n \,:\, ty_n}{\Gamma \vdash \{l_1{=}val_1\,;\ ..\ ;\ l_n{=}val_n\} \,:\, \{l_1 \,:\, ty_1\,;\ ..\ ;\ l_n \,:\, ty_n\}}\ \textsc{typing\_Tval\_record}$$

$$\frac{\begin{array}{l}\Gamma \vdash value \,:\, ty' \\ [constructor \,:\, ty'] \odot vty'{=}vty\end{array}}{\Gamma \vdash (constructor\ value \,:\, vty) \,:\, vty}\ \textsc{typing\_Tval\_constructor}$$

$\boxed{\Gamma \vdash f \,:\, ty \Rightarrow ty'}$ Function symbol typing

$$\frac{[\mathit{constructor}\ :\ \mathit{ty}'] \odot \mathit{vty}'{=}\mathit{vty}}{\Gamma \vdash (\mathit{constructor}\ :\ \mathit{vty})\ :\ \mathit{ty}' \Rightarrow \mathit{vty}}\ \textsc{typing\_Tf\_constructor}$$

$\boxed{\Gamma \vdash \mathit{ty}_1 \equiv \mathit{ty}_2}$ Type equality

$$\frac{}{\Gamma \vdash \mathit{ty} \equiv \mathit{ty}}\ \textsc{typing\_Teq\_refl}$$

$$\frac{\Gamma \vdash \mathit{ty}_1 \equiv \mathit{ty}_2}{\Gamma \vdash \mathit{ty}_2 \equiv \mathit{ty}_1}\ \textsc{typing\_Teq\_sym}$$

$$\frac{\Gamma \vdash \mathit{ty}_1 \equiv \mathit{ty}_2 \quad \Gamma \vdash \mathit{ty}_2 \equiv \mathit{ty}_3}{\Gamma \vdash \mathit{ty}_1 \equiv \mathit{ty}_3}\ \textsc{typing\_Teq\_trans}$$

$$\frac{\Gamma \vdash \mathit{ty}_1 \equiv \mathit{ty}'_1 \quad .. \quad \Gamma \vdash \mathit{ty}_n \equiv \mathit{ty}'_n}{\Gamma \vdash \{l_1\ :\ \mathit{ty}_1;\ ..;\ l_n\ :\ \mathit{ty}_n\} \equiv \{l_1\ :\ \mathit{ty}'_1;\ ..;\ l_n\ :\ \mathit{ty}'_n\}}\ \textsc{typing\_Teq\_congr}$$

$$\frac{}{\Gamma \vdash \mathtt{unit} \equiv \{\}}\ \textsc{typing\_Teq\_unit}$$

$$\frac{\Gamma \vdash \mathit{ty}_1 \equiv \mathit{ty}'_1 \quad .. \quad \Gamma \vdash \mathit{ty}_k \equiv \mathit{ty}'_k}{\Gamma \vdash [\mathit{constructor}_1\ :\ \mathit{ty}_1\ |\ ..\ |\ \mathit{constructor}_k\ :\ \mathit{ty}_k] \equiv [\mathit{constructor}_1\ :\ \mathit{ty}'_1\ |\ ..\ |\ \mathit{constructor}_k\ :\ \mathit{ty}'_k]}\ \textsc{typing\_}$$

$$\frac{}{\Gamma \vdash \mathtt{option}\ \mathit{ty} \equiv [\mathtt{None}\ :\ \mathtt{unit}\ |\ \mathtt{Some}\ :\ \mathit{ty}]}\ \textsc{typing\_Teq\_option}$$

$\boxed{\Gamma \vdash \mathit{ty}}$ Type well-formedness

$$\frac{\begin{array}{l}\mathtt{sorted}\{l_1;\ ..;\ l_n\} \\ \Gamma \vdash \mathit{ty}_1 \quad .. \quad \Gamma \vdash \mathit{ty}_n \\ \Gamma \vdash\end{array}}{\Gamma \vdash \{l_1\ :\ \mathit{ty}_1;\ ..;\ l_n\ :\ \mathit{ty}_n\}}\ \textsc{typing\_twf\_record}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \mathtt{unit}}\ \textsc{typing\_twf\_unit}$$

$$\frac{\begin{array}{l}\mathtt{sorted}\{\mathit{constructor}_1;\ ..;\ \mathit{constructor}_k\} \\ \Gamma \vdash \mathit{ty}_1 \quad .. \quad \Gamma \vdash \mathit{ty}_k \\ \Gamma \vdash\end{array}}{\Gamma \vdash [\mathit{constructor}_1\ :\ \mathit{ty}_1\ |\ ..\ |\ \mathit{constructor}_k\ :\ \mathit{ty}_k]}\ \textsc{typing\_twf\_variant}$$

$$\frac{\Gamma \vdash \mathit{ty}}{\Gamma \vdash \mathtt{option}\ \mathit{ty}}\ \textsc{typing\_twf\_option}$$

$\boxed{\mathtt{tvar} \notin \mathsf{FV}(\mathit{ty})}$ Type variable freshness

$$\frac{\texttt{tvar} \notin \mathsf{FV}(ty_1) \quad .. \quad \texttt{tvar} \notin \mathsf{FV}(ty_n)}{\texttt{tvar} \notin \mathsf{FV}(\{l_1 \,:\, ty_1\,;\; ..\;;\; l_n \,:\, ty_n\})} \quad \textsc{typing\_Tfresh\_tvarrecord}$$

$$\frac{}{\texttt{tvar} \notin \mathsf{FV}(\texttt{unit})} \quad \textsc{typing\_Tfresh\_tvarunit}$$

$$\frac{\texttt{tvar} \notin \mathsf{FV}(ty_1) \quad .. \quad \texttt{tvar} \notin \mathsf{FV}(ty_k)}{\texttt{tvar} \notin \mathsf{FV}([constructor_1 \,:\, ty_1 \mid .. \mid constructor_k \,:\, ty_k])} \quad \textsc{typing\_Tfresh\_tvarvariant}$$

$$\frac{\texttt{tvar} \notin \mathsf{FV}(ty)}{\texttt{tvar} \notin \mathsf{FV}(\texttt{option}\, ty)} \quad \textsc{typing\_Tfresh\_tvaroption}$$

n, m			
id			
k			
rty	::=		Record type
	|	$\{l_1 \,:\, ty_1\,;\; ..\;;\; l_n \,:\, ty_n\}$	
lhs	::=		Left-hand side of assignement
	|	var	
	|	$\{l_1{=}var_1\,;\; ..\;;\; l_n{=}var_n\}$	
arg	::=		Function argument
	|	var	
	|	$value$	
	|	$\{l_1{=}var_1\,;\; ...\;;\; l_n{=}var_n\}$	
$rval$	::=		Record value
	|	$\{l_1{=}value_1\,;\; ..\;;\; l_n{=}value_n\}$	Record
Γ	::=		Typing context (for inlined d
	|	.	
$products$	::=		Products of types
	|	$\{ty_1 \,\texttt{*}\, ty_1'\,;\; ..\;;\; ty_n \,\texttt{*}\, ty_n'\}$	
$label$, l, var, x	::=		Label / variable
	|	id	S
vty	::=		
	|	$[constructor_1 \,:\, ty_1 \mid ... \mid constructor_k \,:\, ty_k]$	

f	::=			Function symbol
	|	(*constructor* : *vty*)		
branches_rhs	::=			
	|	$constructor_1\ var_1$ -> rhs_1 | ... | $constructor_k\ var_k$ -> rhs_k		
branches_ins	::=			
	|	$constructor_1\ var_1$ -> ins_1 | ... | $constructor_k\ var_k$ -> ins_k		
constructor	::=			
	|	*id*	S	
ty, *a*, *b*, *c*	::=			Type
	|	*rty*		Record type
	|	(*ty*)	S	Parenthesised ty
	|	unit		
	|	*vty*		
	|	bool		
rhs	::=			Right-hand side of
	|	*arg*		
	|	*f arg*		
	|	*var*.*l*		
	|	{*var* with l_1=var_1 ; ... ; l_n=var_n}		
	|	match *var* with *branches_rhs* end		
	|	x_1		
	|	x_1&&x_2		
	|	x_1||x_2		
	|	x_1↑x_2		
bool_val	::=			
	|	true	S	
	|	false	S	
value, *val*	::=			Value
	|	*rval*		Record
	|	(*value*)	S	

	|	`(`*constructor value* `:` *vty*`)`	
	|	*bool_val*	

instruction, *I*, *ins*	::=		Instruction
	|	`noop`	No operation
	|	*instruction*$_1$`;` *instruction*$_2$	Sequencing
	|	*lhs*`=`*rhs*	Assignment
	|	`match` *var* `with` *branches_ins* `end`	
	|	`loop` *var* `do` *instruction* `done`	

$\boxed{\Gamma \vdash}$ Context well-fornedness

$$\frac{}{. \vdash}\ \textsc{typing\_Gwf\_empty}$$

$\boxed{\Gamma \vdash lhs : ty \Rightarrow ty'}$ Left-hand sides typing

$$\frac{}{\Gamma \vdash var : ty \Rightarrow \{var : ty\}}\ \textsc{typing\_Tlhs\_var}$$

$$\frac{}{\Gamma \vdash \{l_1{=}x_1;\ ..;\ l_n{=}x_n\} : \{l_1 : ty_1;\ ..;\ l_n : ty_n\} \Rightarrow \{x_1 : ty_1;\ ..;\ x_n : ty_n\}}\ \textsc{typing\_Tlhs\_record}$$

$\boxed{\Gamma \vdash_a arg : ty \Rightarrow ty'}$ Argument typing

$$\frac{}{\Gamma \vdash_a var : \{var : ty\} \Rightarrow ty}\ \textsc{typing\_Targ\_var}$$

$$\frac{\Gamma \vdash value : ty}{\Gamma \vdash_a value : \{\} \Rightarrow ty}\ \textsc{typing\_Targ\_value}$$

$$\frac{}{\Gamma \vdash_a \{l_1{=}x_1;\ ..;\ l_n{=}x_n\} : \{x_1 : ty_1;\ ..;\ x_n : ty_n\} \Rightarrow \{l_1 : ty_1;\ ..;\ l_n : ty_n\}}\ \textsc{typing\_Targ\_record}$$

$\boxed{\Gamma \vdash f : ty \Rightarrow ty'}$ Function symbol typing

$$\frac{[constructor : ty'] \odot vty'{=}vty}{\Gamma \vdash (constructor : vty) : ty' \Rightarrow vty}\ \textsc{typing\_Tf\_constructor}$$

$\boxed{\Gamma \vdash ty_1 \equiv ty_2}$ Type equality

$$\frac{}{\Gamma \vdash ty \equiv ty}\ \textsc{typing\_Teq\_refl}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_2}{\Gamma \vdash ty_2 \equiv ty_1}\ \textsc{typing\_Teq\_sym}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_2 \quad \Gamma \vdash ty_2 \equiv ty_3}{\Gamma \vdash ty_1 \equiv ty_3} \quad \textsc{typing\_Teq\_trans}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_1' \quad .. \quad \Gamma \vdash ty_n \equiv ty_n'}{\Gamma \vdash \{l_1 : ty_1;\ ..;\ l_n : ty_n\} \equiv \{l_1 : ty_1';\ ..;\ l_n : ty_n'\}} \quad \textsc{typing\_Teq\_congr}$$

$$\frac{}{\Gamma \vdash \texttt{unit} \equiv \{\}} \quad \textsc{typing\_Teq\_unit}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_1' \quad .. \quad \Gamma \vdash ty_k \equiv ty_k'}{\Gamma \vdash [constructor_1 : ty_1 \mid .. \mid constructor_k : ty_k] \equiv [constructor_1 : ty_1' \mid .. \mid constructor_k : ty_k']} \quad \textsc{typing}$$

$$\frac{}{\Gamma \vdash \texttt{bool} \equiv [\texttt{False} : \texttt{unit} \mid \texttt{True} : \texttt{unit}]} \quad \textsc{typing\_Teq\_bool}$$

$\boxed{\Gamma \vdash ty}$ Type well-formedness

$$\frac{\begin{array}{l}\texttt{sorted}\{l_1;\ ..;\ l_n\} \\ \Gamma \vdash ty_1 \quad .. \quad \Gamma \vdash ty_n \\ \Gamma \vdash \end{array}}{\Gamma \vdash \{l_1 : ty_1;\ ..;\ l_n : ty_n\}} \quad \textsc{typing\_twf\_record}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \texttt{unit}} \quad \textsc{typing\_twf\_unit}$$

$$\frac{\begin{array}{l}\texttt{sorted}\{constructor_1;\ ..;\ constructor_k\} \\ \Gamma \vdash ty_1 \quad .. \quad \Gamma \vdash ty_k \\ \Gamma \vdash \end{array}}{\Gamma \vdash [constructor_1 : ty_1 \mid .. \mid constructor_k : ty_k]} \quad \textsc{typing\_twf\_variant}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \texttt{bool}} \quad \textsc{typing\_twf\_bool}$$

$\boxed{\texttt{tvar} \notin \mathsf{FV}(ty)}$ Type variable freshness

$$\frac{\texttt{tvar} \notin \mathsf{FV}(ty_1) \quad .. \quad \texttt{tvar} \notin \mathsf{FV}(ty_n)}{\texttt{tvar} \notin \mathsf{FV}(\{l_1 : ty_1;\ ..;\ l_n : ty_n\})} \quad \textsc{typing\_Tfresh\_tvarrecord}$$

$$\frac{}{\texttt{tvar} \notin \mathsf{FV}(\texttt{unit})} \quad \textsc{typing\_Tfresh\_tvarunit}$$

$$\frac{\texttt{tvar} \notin \mathsf{FV}(ty_1) \quad .. \quad \texttt{tvar} \notin \mathsf{FV}(ty_k)}{\texttt{tvar} \notin \mathsf{FV}([constructor_1 : ty_1 \mid .. \mid constructor_k : ty_k])} \quad \textsc{typing\_Tfresh\_tvarvariant}$$

$$\frac{}{\texttt{tvar} \notin \mathsf{FV}(\texttt{bool})} \quad \textsc{typing\_Tfresh\_tvarbool}$$

$\boxed{\Gamma \vdash value : ty}$ Value typing

$$\frac{\Gamma \vdash val_1 \ : \ ty_1 \quad .. \quad \Gamma \vdash val_n \ : \ ty_n}{\Gamma \vdash \{l_1{=}val_1 \, ; \ .. \ ; \ l_n{=}val_n\} \ : \ \{l_1 \ : \ ty_1 \, ; \ .. \ ; \ l_n \ : \ ty_n\}} \ \textsc{typing\_Tval\_record}$$

$$\frac{\Gamma \vdash value \ : \ ty' \quad [constructor \ : \ ty'] \odot vty'{=}vty}{\Gamma \vdash (constructor \, value \ : \ vty) \ : \ vty} \ \textsc{typing\_Tval\_constructor}$$

$$\frac{}{\Gamma \vdash bool\_val \ : \ \texttt{bool}} \ \textsc{typing\_Tval\_bool}$$

$\boxed{\Gamma \vdash rhs \ : \ ty \Rightarrow ty'}$ Right-hand side typing

$$\frac{\Gamma \vdash_a arg \ : \ ty \Rightarrow ty'}{\Gamma \vdash arg \ : \ ty \Rightarrow ty'} \ \textsc{typing\_Trhs\_arg}$$

$$\frac{\Gamma \vdash_a arg \ : \ ty \Rightarrow ty' \quad \Gamma \vdash f \ : \ ty' \Rightarrow ty''}{\Gamma \vdash f \ arg \ : \ ty \Rightarrow ty''} \ \textsc{typing\_Trhs\_f}$$

$$\frac{\{l \ : \ ty\} \odot rty{=}rty' \quad \Gamma \vdash rty'}{\Gamma \vdash var.l \ : \ rty' \Rightarrow ty} \ \textsc{typing\_Trhs\_projection}$$

$$\frac{\Gamma \vdash rty' \quad \{l_1 \ : \ ty_1 \, ; \ .. \ ; \ l_n \ : \ ty_n\} \odot rty{=}rty'}{\Gamma \vdash \{var \ \texttt{with} \ l_1{=}var_1 \, ; \ .. \ ; \ l_n{=}var_n\} \ : \ \{var \ : \ rty' ; \ var_1 \ : \ ty_1 \, ; \ .. \ ; \ var_n \ : \ ty_n\} \Rightarrow rty'} \ \textsc{typing\_Trhs\_upd}$$

$$\frac{\{var \ : \ [constructor_1 \ : \ ty_1 \ | \ .. \ | \ constructor_k \ : \ ty_k]\} \odot rty{=}rty' \quad \{var_1 \ : \ ty_1\} \odot rty{=}rty_1 \quad .. \quad \{var_k \ : \ ty_k\} \odot rty{=}rty_k \quad \Gamma \vdash rhs_1 \ : \ rty_1 \Rightarrow ty \quad .. \quad \Gamma \vdash rhs_k \ : \ rty_k \Rightarrow ty}{\Gamma \vdash \texttt{match} \ var \ \texttt{with} \ constructor_1 \ var_1 \ \texttt{->} \ rhs_1 \ \texttt{|} \ .. \ \texttt{|} \ constructor_k \ var_k \ \texttt{->} \ rhs_k \ \texttt{end} \ : \ rty' \Rightarrow ty} \ \textsc{typing\_Trh}$$

$$\frac{}{\Gamma \vdash \ x_1 \ : \ \{x_1 \ : \ \texttt{bool}\} \Rightarrow \texttt{bool}} \ \textsc{typing\_Trhs\_not}$$

$$\frac{}{\Gamma \vdash x_1 \& \& x_2 \ : \ \{x_1 \ : \ \texttt{bool}; \ x_2 \ : \ \texttt{bool}\} \Rightarrow \texttt{bool}} \ \textsc{typing\_Trhs\_and}$$

$$\frac{}{\Gamma \vdash x_1 || x_2 \ : \ \{x_1 \ : \ \texttt{bool}; \ x_2 \ : \ \texttt{bool}\} \Rightarrow \texttt{bool}} \ \textsc{typing\_Trhs\_or}$$

$$\frac{}{\Gamma \vdash x_1 \uparrow x_2 \ : \ \{x_1 \ : \ \texttt{bool}; \ x_2 \ : \ \texttt{bool}\} \Rightarrow \texttt{bool}} \ \textsc{typing\_Trhs\_xor}$$

$\boxed{\Gamma \vdash instruction \ : \ ty \Rightarrow ty'}$ Instruction typing

$$\frac{\begin{array}{l}\Gamma \vdash rty_1 \\ \Gamma \vdash rty_2 \\ rty \odot rty'' = rty_1 \\ rty' \odot rty'' = rty_2 \\ \Gamma \vdash instruction \ : \ rty \Rightarrow rty'\end{array}}{\Gamma \vdash instruction \ : \ rty_1 \Rightarrow rty_2} \ \textsc{typing\_T\_frame}$$

$$\frac{}{\Gamma \vdash \texttt{noop} \ : \ \{\,\} \Rightarrow \{\,\}} \ \textsc{typing\_T\_noop}$$

$$\frac{\begin{array}{l}\Gamma \vdash instruction \ : \ ty_1 \Rightarrow ty_2 \\ \Gamma \vdash instruction' \ : \ ty_2 \Rightarrow ty_3\end{array}}{\Gamma \vdash instruction \texttt{;} \ instruction' \ : \ ty_1 \Rightarrow ty_3} \ \textsc{typing\_T\_seq}$$

$$\frac{\begin{array}{l}\Gamma \vdash rhs \ : \ a \Rightarrow b \\ \Gamma \vdash lhs \ : \ b \Rightarrow c\end{array}}{\Gamma \vdash lhs \texttt{=} rhs \ : \ a \Rightarrow c} \ \textsc{typing\_T\_assign}$$

$$\frac{\begin{array}{l}\{var \ : \ [constructor_1 \ : \ ty_1 \ \texttt{|} \ .. \ \texttt{|} \ constructor_k \ : \ ty_k]\} \odot rty = rty' \\ \{var_1 \ : \ ty_1\} \odot rty = rty_1 \quad .. \quad \{var_k \ : \ ty_k\} \odot rty = rty_k \\ \Gamma \vdash I_1 \ : \ rty_1 \Rightarrow ty \quad .. \quad \Gamma \vdash I_k \ : \ rty_k \Rightarrow ty\end{array}}{\Gamma \vdash \texttt{match} \ var \ \texttt{with} \ constructor_1 \ var_1 \ \texttt{->} \ I_1 \ \texttt{|} \ .. \ \texttt{|} \ constructor_k \ var_k \ \texttt{->} \ I_k \ \texttt{end} \ : \ rty' \Rightarrow ty} \ \textsc{typing\_T\_matc}$$

$$\frac{\begin{array}{l}\{var \ : \ \texttt{bool}\} \odot rty = rty' \\ \Gamma \vdash I \ : \ rty \Rightarrow rty'\end{array}}{\Gamma \vdash \texttt{loop} \ var \ \texttt{do} \ I \ \texttt{done} \ : \ rty' \Rightarrow rty} \ \textsc{typing\_T\_loop}$$

$\boxed{\texttt{E} \vdash lhs/val \ \texttt{->} \ val'}$ Left-hand side evaluation

$$\frac{}{\texttt{E} \vdash var/val \ \texttt{->} \ \{var \texttt{=} val\}} \ \textsc{eval\_lhs\_var}$$

$$\frac{}{\texttt{E} \vdash \{l_1 \texttt{=} x_1 \texttt{;} \ .. \ \texttt{;} \ l_n \texttt{=} x_n\}/\{l_1 \texttt{=} val_1 \texttt{;} \ .. \ \texttt{;} \ l_n \texttt{=} val_n\} \ \texttt{->} \ \{x_1 \texttt{=} val_1 \texttt{;} \ .. \ \texttt{;} \ x_n \texttt{=} val_n\}} \ \textsc{eval\_lhs\_record}$$

$\boxed{\texttt{E} \vdash arg/aval \ \texttt{->} \ val'}$ Argument evaluation

$$\frac{}{\texttt{E} \vdash var/a\{var \texttt{=} val\} \ \texttt{->} \ val} \ \textsc{eval\_arg\_var}$$

$$\frac{}{\texttt{E} \vdash val/a\{\,\} \ \texttt{->} \ val} \ \textsc{eval\_arg\_val}$$

$$\frac{}{\texttt{E} \vdash \{l_1 \texttt{=} x_1 \texttt{;} \ .. \ \texttt{;} \ l_n \texttt{=} x_n\}/a\{x_1 \texttt{=} val_1 \texttt{;} \ .. \ \texttt{;} \ x_n \texttt{=} val_n\} \ \texttt{->} \ \{l_1 \texttt{=} val_1 \texttt{;} \ .. \ \texttt{;} \ l_n \texttt{=} val_n\}} \ \textsc{eval\_arg\_record}$$

$\boxed{\texttt{E} \vdash f/val \ \texttt{->} \ val'}$ Function symbol evaluation

$$\frac{}{\texttt{E} \vdash (constructor\ :\ vty)/val\ \texttt{->}\ (constructor\ val\ :\ vty)}\ \text{EVAL\_F\_CONSTRUCTOR}$$

$\boxed{\texttt{E} \vdash rhs/val\ \texttt{->}\ val'}$ Right-hand side evaluation

$$\frac{\texttt{E} \vdash arg/aval\ \texttt{->}\ val'}{\texttt{E} \vdash arg/val\ \texttt{->}\ val'}\ \text{EVAL\_RHS\_ARG}$$

$$\frac{\begin{array}{l}\texttt{E} \vdash arg/aval\ \texttt{->}\ val' \\ \texttt{E} \vdash f/val'\ \texttt{->}\ val''\end{array}}{\texttt{E} \vdash f\ arg/val\ \texttt{->}\ val''}\ \text{EVAL\_RHS\_F}$$

$$\frac{\texttt{\{}l\texttt{=}val\texttt{\}} \odot rval\texttt{=}rval'}{\texttt{E} \vdash var.l/rval'\ \texttt{->}\ val}\ \text{EVAL\_RHS\_PROJECTION}$$

$$\frac{\begin{array}{l}\texttt{\{}l_1\texttt{=}val'_1\texttt{;}\ ..\ \texttt{;}\ l_n\texttt{=}val'_n\texttt{\}} \odot rval\texttt{=}rval' \\ \texttt{\{}l_1\texttt{=}val_1\texttt{;}\ ..\ \texttt{;}\ l_n\texttt{=}val_n\texttt{\}} \odot rval\texttt{=}rval''\end{array}}{\texttt{E} \vdash \texttt{\{}var\ \texttt{with}\ l_1\texttt{=}var_1\texttt{;}\ ..\ \texttt{;}\ l_n\texttt{=}var_n\texttt{\}/\{}var\texttt{=}rval'\texttt{;}\ var_1\texttt{=}val_1\texttt{;}\ ..\ \texttt{;}\ var_n\texttt{=}val_n\texttt{\}}\ \texttt{->}\ rval''}\ \text{EVAL\_RHS\_UPDATE}$$

$$\frac{\texttt{E} \vdash rhs/\texttt{\{}var'\texttt{=}val'\texttt{\}}\ \texttt{->}\ val}{\texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ constructor\ var'\ \texttt{->}\ rhs\ \texttt{|}\ <branches\_rhs>\ \texttt{end/\{}var\texttt{=(}constructor\ val'\ :\ vty\texttt{)\}}\ \texttt{->}\ val}$$

$$\frac{\begin{array}{l}constructor <> constructor' \\ \texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ branches\_rhs\ \texttt{end/\{}var\texttt{=(}constructor\ val'\ :\ vty\texttt{)\}}\ \texttt{->}\ val'\end{array}}{\texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ constructor'\ var'\ \texttt{->}\ rhs\ \texttt{|}\ <branches\_rhs>\ \texttt{end/\{}var\texttt{=(}constructor\ val'\ :\ vty\texttt{)\}}\ \texttt{->}\ val}$$

$$\frac{}{\texttt{E} \vdash\ x_1\texttt{/\{}x_1\texttt{=}bool\_val_1\texttt{\}}\ \texttt{->}\ \ bool\_val_1}\ \text{EVAL\_RHS\_NOT}$$

$$\frac{}{\texttt{E} \vdash x_1\texttt{\&\&}x_2\texttt{/\{}x_1\texttt{=}bool\_val_1\texttt{;}\ x_2\texttt{=}bool\_val_2\texttt{\}}\ \texttt{->}\ bool\_val_1\texttt{\&\&}bool\_val_2}\ \text{EVAL\_RHS\_AND}$$

$$\frac{}{\texttt{E} \vdash x_1\texttt{||}x_2\texttt{/\{}x_1\texttt{=}bool\_val_1\texttt{;}\ x_2\texttt{=}bool\_val_2\texttt{\}}\ \texttt{->}\ bool\_val_1\texttt{||}bool\_val_2}\ \text{EVAL\_RHS\_OR}$$

$$\frac{}{\texttt{E} \vdash x_1\texttt{\textuparrow}x_2\texttt{/\{}x_1\texttt{=}bool\_val_1\texttt{;}\ x_2\texttt{=}bool\_val_2\texttt{\}}\ \texttt{->}\ bool\_val_1\texttt{\textuparrow}bool\_val_2}\ \text{EVAL\_RHS\_XOR}$$

$\boxed{\texttt{E} \vdash instruction/val\ \texttt{->}\ val'}$ Instruction evaluation

$$\frac{\begin{array}{l}\texttt{E} \vdash instruction/rval\ \texttt{->}\ rval' \\ rval \odot rval''\texttt{=}rval_1 \\ rval' \odot rval''\texttt{=}rval_2\end{array}}{\texttt{E} \vdash instruction/rval_1\ \texttt{->}\ rval_2}\ \text{EVAL\_INSTR\_FRAME}$$

$$\frac{}{\texttt{E} \vdash \texttt{noop/\{\}}\ \texttt{->}\ \texttt{\{\}}}\ \text{EVAL\_INSTR\_NOOP}$$

$$\frac{\texttt{E} \vdash I_1/val \texttt{ -> } val' \quad \texttt{E} \vdash I_2/val' \texttt{ -> } val''}{\texttt{E} \vdash I_1\texttt{;}\ I_2/val \texttt{ -> } val''}\ \text{EVAL\_INSTR\_SEQ}$$

$$\frac{\texttt{E} \vdash rhs/val \texttt{ -> } val' \quad \texttt{E} \vdash lhs/val' \texttt{ -> } val''}{\texttt{E} \vdash lhs\texttt{=}rhs/val \texttt{ -> } val''}\ \text{EVAL\_INSTR\_ASSIGN}$$

$$\frac{\begin{array}{l}\{var\texttt{=(}constructor\ val'\ \texttt{:}\ vty\texttt{)}\} \odot rval\texttt{=}rval' \\ \{var'\texttt{=}val'\texttt{}\} \odot rval\texttt{=}rval' \\ \texttt{E} \vdash I/rval' \texttt{ -> } rval_1\end{array}}{\texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ constructor\ var'\ \texttt{->}\ I\ \texttt{|}\ <branches\_ins>\ \texttt{end}/rval' \texttt{ -> } rval_1}\ \text{EVAL\_INSTR\_MATCH\_F}$$

$$\frac{\begin{array}{l}\{var\texttt{=(}constructor\ val'\ \texttt{:}\ vty\texttt{)}\} \odot rval\texttt{=}rval' \\ constructor <> constructor' \\ \texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ branches\_ins\ \texttt{end}/rval' \texttt{ -> } rval_1\end{array}}{\texttt{E} \vdash \texttt{match}\ var\ \texttt{with}\ constructor'\ var'\ \texttt{->}\ I\ \texttt{|}\ <branches\_ins>\ \texttt{end}/rval' \texttt{ -> } rval_1}\ \text{EVAL\_INSTR\_MATCH\_N}$$

$$\frac{\{var'\texttt{=(False\{\} :}\ vty\texttt{)}\} \odot rval\texttt{=}rval'}{\texttt{E} \vdash \texttt{loop}\ var\ \texttt{do}\ I\ \texttt{done}/rval' \texttt{ -> } rval}\ \text{EVAL\_INSTR\_LOOP\_END}$$

$$\frac{\begin{array}{l}\{var'\texttt{=(True\{\} :}\ vty\texttt{)}\} \odot rval\texttt{=}rval' \\ \texttt{E} \vdash I/rval \texttt{ -> } rval_1 \\ \texttt{E} \vdash \texttt{loop}\ var\ \texttt{do}\ I\ \texttt{done}/rval_1 \texttt{ -> } rval_2\end{array}}{\texttt{E} \vdash \texttt{loop}\ var\ \texttt{do}\ I\ \texttt{done}/rval' \texttt{ -> } rval_2}\ \text{EVAL\_INSTR\_LOOP\_CONTINUE}$$

2.3 Functions

In Albert, a strong distinction is made between toplevel and anonymous functions. Toplevel functions defined using the `def` keyword can be used directly zero, one, or several times in the rest of the Albert file. Anonymous functions introduced by the `lambda`) keyword however are Albert terms of type $ty- > ty'$ that are subject of the linear type system so they are ressources that can be duplicated or destroyed using `dup` and `drop`.

2.4 Domain specific operations

This section lists the domain specific operations of Albert. These operations are similar to the ones of Michelson and are listed here only for completeness and to specify their Albert syntax.

3 Examples

```
type storage_ty = { threshold : mutez; votes: map string nat }

def vote :
   { store : storage_ty ; param : string } ->
   { operations : list opertation ; out_storage : storage_ty } =

    (store0, store1) = dup store;
    threshold = store1.threshold;
    (threshold, threshold_copy) = dup threshold;
    ok = amount < threshold;
    match ok with
       True -> state = store0.votes ;
               (state0, state1) = dup state;
               (param0, param1) = dup param;
               prevote_option =  get state1 param1;
               prevote = assert_some prevote_option ;
               postvote = prevote + 1 ;
               final_state = update state0 param0 (Some prevote);
               out_storage = {votes = final_state; threshold = threshold_copy};
               operations = []
     | False ->
               failwith "you're so cheap!"
    end
```

4 Conclusion

The Albert language is an intermediate language between Michelson and the existing high-level languages for writing smart contracts on the Tezos blockchain. Its set of features

5 References

References

[1] Guillaume Duhamel Benoit Rognier Pierr-Yves Sturb edukera. Archetype - a Tezos smart contract development solution dedicated to contract quality insurance. `https://docs.archetype-lang.org`. Accessed: 2019-06-08.

[2] Gabriel Alfour Nomadic Labs. The LIGO smart-contract language. `https://ligolang.org`. Accessed: 2019-06-08.

[3] Stephen Andrews Richard Ayotte. fi - Smart coding for smart contracts. https://fi-code.com. Accessed: 2019-06-08.

[4] François Maurel Smart Chain Arena. SmartPy. https://smartpy.io. Accessed: 2019-06-08.

[5] Arthur Breitman. Multisig contract in Michelson. https://github.com/murbard/smart-contracts/blob/master/multisig/michelson/multisig.tz. Accessed: 2019-06-04.

[6] Raphaël Cauderlier. Certification of Breitman's Multisig implementation in Mi-Cho-Coq. https://gitlab.com/nomadic-labs/mi-cho-coq/blob/master/src/contracts_coq/multisig.v. Accessed: 2019-06-04.

[7] Morley contributors. Morley: Developer tools for the Michelson Language. https://gitlab.com/morley-framework/morley. Accessed: 2019-06-08.

[8] Nomadic Labs. Mi-Cho-Coq public repository. https://gitlab.com/nomadic-labs/mi-cho-coq. Accessed: 2019-06-04.

A Albert grammar

rty	::=		Record type
	|	$\{l_1 : ty_1 ; \ldots ; l_n : ty_n\}$	
arg	::=		Function argument
	|	*var*	
	|	*value*	
	|	$\{l_1 = var_1 ; \ldots ; l_n = var_n\}$	
rval	::=		Record value
	|	$\{l_1 = value_1 ; \ldots ; l_n = value_n\}$	Record
products	::=		Products of types
	|	$\{ty_1 * ty_1' ; \ldots ; ty_n * ty_n'\}$	
tvar	::=		
	|	*id* S	
fvar	::=		
	|	*id* S	
definition	::=		Toplevel definition
	|	`type` *tvar*=*ty*	
	|	`def` *fvar* : *ty* `->` *ty*′=*instruction*	
prog	::=		Toplevel definition
	|	*definition*	
	|	*definition prog*	
Γ	::=		Typing context (for inlined definitions)
	|	.	
	|	$\Gamma, fvar : ty \Rightarrow ty'$	
	|	$\Gamma, tvar = ty$	
E	::=		Semantic context (for function definitions)
	|	$E, fvar = instruction$	

nat_constant	::=		
	|	*num_litteral*	S
int_constant	::=		
	|	*int_litteral*	S
timestamp_constant	::=		
	|	*num_litteral* `ts`	S
mutez_constant	::=		
	|	*num_litteral* `utz`	S
lhs	::=		Left-hand si
	|	*var*	
	|	{l_1=var_1 ; .. ; l_n=var_n}	
	|	(var_1, var_2)	S
vty	::=		
	|	[$constructor_1$: ty_1 | ... | $constructor_k$: ty_k]	
branches_rhs	::=		
	|	$constructor_1$ var_1 -> rhs_1 | ... | $constructor_k$ var_k -> rhs_k	
branches_ins	::=		
	|	$constructor_1$ var_1 -> ins_1 | ... | $constructor_k$ var_k -> ins_k	
constructor	::=		
	|	*id*	S
bool_val	::=		
	|	`true`	S
	|	`false`	S
string_val, *sval*	::=		
	|	*string_litteral*	S
bytes_val, *bval*	::=		

	|	*bytes_litteral*	S
lval, *tl*	::=	
	|	`([`*value*$_1$`;` .. `;` *value*$_n$`] :` `list` *ty*`)`
instruction, *I*, *ins*	::=	Instr
	|	`noop` N
	|	*instruction*$_1$`;` *instruction*$_2$ Se
	|	*lhs*`=`*rhs* A
	|	`failwith` *arg* P
	|	`drop` *var* R
	|	`match` *var* `with` *branches_ins* `end`
	|	`loop_left` *var* `in` *var*$'$ `do` *instruction* `done`
	|	`loop` *var* `do` *instruction* `done`
	|	`for` *var* `in` *var*$'$ `do` *instruction*$_1$ `done`
	|	`match` *var* `with []` `->` *instruction*$_1$ `|` *var*$_1$ `::` *var*$_2$ `->` *instruction*$_2$ `end`
	|	`for` *var*$_1$ `|->` *var*$_2$ `in` *var*$_3$ `do` *instruction*$_1$ `done`
mval	::=	
	|	`({|`*value*$_1$ `|->` *value*$'_1$`;` .. `;` *value*$_n$ `|->` *value*$'_n$`|}) :` `map` *ty ty*$'$`)`
num_val, *nv*	::=	
	|	*nat_constant*
	|	*int_constant*
	|	*timestamp_constant*
	|	*mutez_constant*
key_constant	::=	
	|	`key` *string_litteral* S
sig_constant	::=	
	|	`sig` *string_litteral* S
tz_constant	::=	
	|	*tz_litteral* S
kt_constant	::=	

| | *kt_litteral* | S

add_val ::=
| | *tz_constant*
| | *kt_constant*

rhs ::= Right-hand side of assignments
| | *arg*
| | *f arg*
| | *var.l*
| | {*var* `with` l_1=var_1 ; ... ; l_n=var_n}
| | x_1 + x_2
| | x_1 - x_2
| | x_1 * x_2
| | x_1 /mod x_2
| | $x_1 << x_2$
| | $x_1 >> x_2$
| | `abs` x_1
| | `is_nat` x_1
| | `int` x_1
| | `match` *var* `with` *branches_rhs* `end`
| | x_1
| | $x_1 \&\& x_2$
| | $x_1 || x_2$
| | $x_1 \uparrow x_2$
| | `concat` x_1 x_2
| | `size` x
| | `slice` x x_1 x_2
| | `pack` x
| | `unpack` *ty* x
| | $var_1 \equiv var_2$
| | $var_1 \neq var_2$
| | $var_1 < var_2$
| | $var_1 <= var_2$
| | $var_1 > var_2$
| | $var_1 >= var_2$

	$[var_1\,;\ \ldots\ ;\ var_n]$		
	var_1 `::` var_2		
	`match` var `with []` `->` rhs_1 `|` var_1 `::` var_2 `->` rhs_2 `end`		
	`map` var `in` var' `do` rhs `done`		
	$\{\|var_1$ `|->` $var'_1\,;\ \ldots\ ;\ var_n$ `|->` $var'_n\|\}$		
	`map` var_1 `|->` var_2 `in` var_3 `do` rhs `done`		
	var_1`[`var_2`]`		map read
	$\{\|var_1$ `with` var_2 `|->` $var_3\|\}$		map update
	`check_signature` var_1 var_2 var_3		
	`amount`		
	`self`		
	`transfer_tokens` var_1 var_2 var_3		
	`set_delegate` var_1		
	`sender`		
	`source`		
	`now`		

$label,\ l,\ var,\ x$	::=		Label / variable
	id	S	
	`function`		
	`argument`		

$ty,\ a,\ b,\ c$	::=		Type
	rty		Record type
	`(`ty`)`	S	Parenthesised type
	$tvar$		Type variable
	`nat`		
	`int`		
	`timestamp`		
	`mutez`		
	`unit`		
	ty `*` ty'		Binary product type
	vty		
	`option` ty		
	a `+` b		
	`bool`		

	`string`
	`bytes`
	`list` a
	`map` a b
	`big_map` a b
	`key`
	`signature`
	`operation`
	`key_hash`
	`address`
	`contract` ty
	ty `->` ty'

f	::=		Function symbol
	|	$fvar$	
	|	`dup`	
	|	(*constructor* `:` *vty*)	
	|	`hash_key`	
	|	$blake2b$	
	|	$sha256$	
	|	$sha512$	
	|	`contract` ty	
	|	`address`	
	|	`implicit_account`	
	|	`exec`	

value, *val*	::=			Value
	|	*rval*		Record
	|	(*value*)	S	
	|	*num_val*		
	|	(*constructor value* `:` *vty*)		
	|	*bool_val*		
	|	*sval*		
	|	*bval*		
	|	*lval*		
	|	*mval*		

| *key_constant*
| *sig_constant*
| *add_val*
| `lambda` *ty* `:` *ty′.instruction* `end`

B Albert type system

$\boxed{\Gamma \vdash lhs \,:\, ty \Rightarrow ty'}$ Left-hand sides typing

$$\frac{}{\Gamma \vdash var \,:\, ty \Rightarrow \{var \,:\, ty\}}\ \textsc{typing\_Tlhs\_var}$$

$$\frac{}{\Gamma \vdash \{l_1{=}x_1\,;\ ..\,;\ l_n{=}x_n\} \,:\, \{l_1 \,:\, ty_1\,;\ ..\,;\ l_n \,:\, ty_n\} \Rightarrow \{x_1 \,:\, ty_1\,;\ ..\,;\ x_n \,:\, ty_n\}}\ \textsc{typing\_Tlhs\_record}$$

$\boxed{\Gamma \vdash_a arg \,:\, ty \Rightarrow ty'}$ Argument typing

$$\frac{}{\Gamma \vdash_a var \,:\, \{var \,:\, ty\} \Rightarrow ty}\ \textsc{typing\_Targ\_var}$$

$$\frac{\Gamma \vdash value \,:\, ty}{\Gamma \vdash_a value \,:\, \{\,\} \Rightarrow ty}\ \textsc{typing\_Targ\_value}$$

$$\frac{}{\Gamma \vdash_a \{l_1{=}x_1\,;\ ..\,;\ l_n{=}x_n\} \,:\, \{x_1 \,:\, ty_1\,;\ ..\,;\ x_n \,:\, ty_n\} \Rightarrow \{l_1 \,:\, ty_1\,;\ ..\,;\ l_n \,:\, ty_n\}}\ \textsc{typing\_Targ\_record}$$

$\boxed{fvar \,:\, ty \Rightarrow ty' \in \Gamma}$ Looking for a function declaration in context

$$\frac{fvar \,:\, ty \Rightarrow ty' \in \Gamma}{fvar \,:\, ty \Rightarrow ty' \in \Gamma, tvar{=}ty''}\ \textsc{typing\_In\_G\_fvardecl}$$

$$\frac{}{fvar \,:\, ty \Rightarrow ty' \in \Gamma, fvar \,:\, ty \Rightarrow ty'}\ \textsc{typing\_In\_G\_fhead}$$

$$\frac{\begin{array}{c}fvar \,:\, ty \Rightarrow ty' \in \Gamma \\ fvar \neq fvar'\end{array}}{fvar \,:\, ty \Rightarrow ty' \in \Gamma, fvar' \,:\, ty'' \Rightarrow ty'''}\ \textsc{typing\_In\_G\_ftail}$$

$\boxed{tvar{=}ty \in \Gamma}$ Looking for a type definition in context

$$\frac{tvar{=}ty \in \Gamma}{tvar{=}ty \in \Gamma, fvar \,:\, ty' \Rightarrow ty''}\ \textsc{typing\_In\_G\_tfundecl}$$

$$\frac{}{tvar{=}ty \in \Gamma, tvar{=}ty}\ \textsc{typing\_In\_G\_thead}$$

$$\frac{\begin{array}{c}tvar{=}ty \in \Gamma \\ tvar \neq tvar'\end{array}}{tvar{=}ty \in \Gamma, tvar'{=}ty'}\ \textsc{typing\_In\_G\_ttail}$$

$\boxed{\Gamma \vdash}$ Context well-fornedness

$$\frac{}{.\vdash}\ \textsc{typing\_Gwf\_empty}$$

$$\frac{\begin{array}{c}tvar\ \texttt{not free in}\ ty \\ \Gamma \vdash ty\end{array}}{\Gamma, tvar{=}ty \vdash}\ \textsc{typing\_Gwf\_typedef}$$

$$\frac{\begin{array}{c}\Gamma \vdash ty \\ \Gamma \vdash ty'\end{array}}{\Gamma, fvar\ :\ ty \Rightarrow ty' \vdash}\ \textsc{typing\_Gwf\_fundecl}$$

$\boxed{\Gamma \vdash ty_1 \equiv ty_2}$ Type equality

$$\frac{}{\Gamma \vdash ty \equiv ty}\ \textsc{typing\_Teq\_refl}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty_2}{\Gamma \vdash ty_2 \equiv ty_1}\ \textsc{typing\_Teq\_sym}$$

$$\frac{\begin{array}{c}\Gamma \vdash ty_1 \equiv ty_2 \\ \Gamma \vdash ty_2 \equiv ty_3\end{array}}{\Gamma \vdash ty_1 \equiv ty_3}\ \textsc{typing\_Teq\_trans}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty'_1 \quad .. \quad \Gamma \vdash ty_n \equiv ty'_n}{\Gamma \vdash \{l_1\ :\ ty_1;\ ..;\ l_n\ :\ ty_n\} \equiv \{l_1\ :\ ty'_1;\ ..;\ l_n\ :\ ty'_n\}}\ \textsc{typing\_Teq\_congr}$$

$$\frac{tvar{=}ty\ \in\ \Gamma}{\Gamma \vdash tvar \equiv ty}\ \textsc{typing\_Teq\_definition}$$

$$\frac{}{\Gamma \vdash \texttt{unit} \equiv \texttt{\{\}}}\ \textsc{typing\_Teq\_unit}$$

$$\frac{}{\Gamma \vdash ty_1\ \texttt{*}\ ty_2 \equiv \texttt{\{car : } ty_1\texttt{; cdr : } ty_2\texttt{\}}}\ \textsc{typing\_Teq\_pair}$$

$$\frac{\Gamma \vdash ty_1 \equiv ty'_1 \quad .. \quad \Gamma \vdash ty_k \equiv ty'_k}{\Gamma \vdash \texttt{[}constructor_1\ \texttt{:}\ ty_1\ \texttt{|}\ ..\ \texttt{|}\ constructor_k\ \texttt{:}\ ty_k\texttt{]} \equiv \texttt{[}constructor_1\ \texttt{:}\ ty'_1\ \texttt{|}\ ..\ \texttt{|}\ constructor_k\ \texttt{:}\ ty'_k\texttt{]}}\ \textsc{typing\_}$$

$$\frac{}{\Gamma \vdash \texttt{option}\ ty \equiv \texttt{[None : unit | Some : } ty\texttt{]}}\ \textsc{typing\_Teq\_option}$$

$$\frac{}{\Gamma \vdash ty\ \texttt{+}\ ty' \equiv \texttt{[Left : } ty\ \texttt{|}\ \texttt{Right : } ty'\texttt{]}}\ \textsc{typing\_Teq\_or}$$

$$\frac{}{\Gamma \vdash \texttt{bool} \equiv \texttt{[False : unit | True : unit]}}\ \textsc{typing\_Teq\_bool}$$

$\boxed{\Gamma \vdash instruction\ :\ ty \Rightarrow ty'}$ Instruction typing

$$\frac{\begin{array}{l}\Gamma \vdash rty_1 \\ \Gamma \vdash rty_2 \\ rty \odot rty''{=}rty_1 \\ rty' \odot rty''{=}rty_2 \\ \Gamma \vdash instruction\ :\ rty \Rightarrow rty'\end{array}}{\Gamma \vdash instruction\ :\ rty_1 \Rightarrow rty_2}\ \textsc{typing\_T\_frame}$$

$$\frac{}{\Gamma \vdash \texttt{noop} : \{\,\} \Rightarrow \{\,\}}\ \textsc{typing\_T\_noop}$$

$$\frac{\Gamma \vdash \mathit{instruction} : \mathit{ty}_1 \Rightarrow \mathit{ty}_2 \quad \Gamma \vdash \mathit{instruction}' : \mathit{ty}_2 \Rightarrow \mathit{ty}_3}{\Gamma \vdash \mathit{instruction}\,;\ \mathit{instruction}' : \mathit{ty}_1 \Rightarrow \mathit{ty}_3}\ \textsc{typing\_T\_seq}$$

$$\frac{\Gamma \vdash \mathit{rhs} : a \Rightarrow b \quad \Gamma \vdash \mathit{lhs} : b \Rightarrow c}{\Gamma \vdash \mathit{lhs}{=}\mathit{rhs} : a \Rightarrow c}\ \textsc{typing\_T\_assign}$$

$$\frac{\mathit{rty} \odot \mathit{rty}'{=}\mathit{rty}_1 \quad \Gamma \vdash_a \mathit{arg} : \mathit{rty} \Rightarrow \mathit{ty}_2}{\Gamma \vdash \texttt{failwith}\,\mathit{arg} : \mathit{rty}_1 \Rightarrow \mathit{rty}_3}\ \textsc{typing\_T\_failwith}$$

$$\frac{}{\Gamma \vdash \texttt{drop}\,\mathit{var} : \{\mathit{var} : \mathit{ty}\} \Rightarrow \texttt{unit}}\ \textsc{typing\_T\_drop}$$

$$\frac{\begin{array}{l}\{\mathit{var} : [\mathit{constructor}_1 : \mathit{ty}_1 \mid \ldots \mid \mathit{constructor}_k : \mathit{ty}_k]\} \odot \mathit{rty}{=}\mathit{rty}' \\ \{\mathit{var}_1 : \mathit{ty}_1\} \odot \mathit{rty}{=}\mathit{rty}_1 \quad \ldots \quad \{\mathit{var}_k : \mathit{ty}_k\} \odot \mathit{rty}{=}\mathit{rty}_k \\ \Gamma \vdash I_1 : \mathit{rty}_1 \Rightarrow \mathit{ty} \quad \ldots \quad \Gamma \vdash I_k : \mathit{rty}_k \Rightarrow \mathit{ty}\end{array}}{\Gamma \vdash \texttt{match}\,\mathit{var}\,\texttt{with}\,\mathit{constructor}_1\,\mathit{var}_1\,\texttt{->}\,I_1 \mid \ldots \mid \mathit{constructor}_k\,\mathit{var}_k\,\texttt{->}\,I_k\,\texttt{end} : \mathit{rty}' \Rightarrow \mathit{ty}}\ \textsc{typing\_T\_matc}$$

$$\frac{\begin{array}{l}\{\mathit{var}' : a + b\} \odot \mathit{rty}{=}\mathit{rty}' \\ \{\mathit{var} : a\} \odot \mathit{rty}{=}\mathit{rty}'' \\ \Gamma \vdash I : \mathit{rty}'' \Rightarrow \mathit{rty}'\end{array}}{\Gamma \vdash \texttt{loop\_left}\,\mathit{var}\,\texttt{in}\,\mathit{var}'\,\texttt{do}\,I\,\texttt{done} : \mathit{rty}' \Rightarrow \mathit{rty}}\ \textsc{typing\_T\_loop\_left}$$

$$\frac{\{\mathit{var} : \texttt{bool}\} \odot \mathit{rty}{=}\mathit{rty}' \quad \Gamma \vdash I : \mathit{rty} \Rightarrow \mathit{rty}'}{\Gamma \vdash \texttt{loop}\,\mathit{var}\,\texttt{do}\,I\,\texttt{done} : \mathit{rty}' \Rightarrow \mathit{rty}}\ \textsc{typing\_T\_loop}$$

$$\frac{\begin{array}{l}\{\mathit{var}' : \texttt{list}\,\mathit{ty}\} \odot \mathit{rty}{=}\mathit{rty}' \\ \{\mathit{var} : \mathit{ty}\} \odot \mathit{rty}{=}\mathit{rty}'' \\ \Gamma \vdash \mathit{instruction}_1 : \mathit{rty}'' \Rightarrow \mathit{rty}\end{array}}{\Gamma \vdash \texttt{for}\,\mathit{var}\,\texttt{in}\,\mathit{var}'\,\texttt{do}\,\mathit{instruction}_1\,\texttt{done} : \mathit{rty}' \Rightarrow \mathit{rty}}\ \textsc{typing\_T\_list\_for}$$

$$\frac{\begin{array}{l}\{\mathit{var} : \texttt{list}\,\mathit{ty}\} \odot \mathit{rty}{=}\mathit{rty}' \\ \{\mathit{var}_1 : \mathit{ty}\,;\ \mathit{var}_2 : \texttt{list}\,\mathit{ty}\} \odot \mathit{rty}{=}\mathit{rty}'' \\ \Gamma \vdash \mathit{instruction}_1 : \mathit{rty} \Rightarrow \mathit{ty}' \\ \Gamma \vdash \mathit{instruction}_2 : \mathit{rty}'' \Rightarrow \mathit{ty}'\end{array}}{\Gamma \vdash \texttt{match}\,\mathit{var}\,\texttt{with}\,\texttt{[]}\,\texttt{->}\,\mathit{instruction}_1 \mid \mathit{var}_1\,\texttt{::}\,\mathit{var}_2\,\texttt{->}\,I_2\,\texttt{end} : \mathit{rty}' \Rightarrow \mathit{ty}'}\ \textsc{typing\_T\_list\_match}$$

$$\frac{\begin{array}{l}\{\mathit{var}_3 : \texttt{map}\,\mathit{ty}_1\,\mathit{ty}_2\} \odot \mathit{rty}{=}\mathit{rty}' \\ \{\mathit{var}_1 : \mathit{ty}_1\,;\ \mathit{var}_2 : \mathit{ty}_2\} \odot \mathit{rty}{=}\mathit{rty}'' \\ \Gamma \vdash \mathit{instruction}_1 : \mathit{rty}'' \Rightarrow \mathit{rty}\end{array}}{\Gamma \vdash \texttt{for}\,\mathit{var}_1\,\texttt{|->}\,\mathit{var}_2\,\texttt{in}\,\mathit{var}_3\,\texttt{do}\,\mathit{instruction}_1\,\texttt{done} : \mathit{rty}' \Rightarrow \mathit{rty}}\ \textsc{typing\_T\_map\_for}$$

$\boxed{\Gamma \vdash rhs \ : \ ty \Rightarrow ty'}$ Right-hand side typing

$$\frac{\Gamma \vdash_a arg \ : \ ty \Rightarrow ty'}{\Gamma \vdash arg \ : \ ty \Rightarrow ty'} \ \textsc{typing\_Trhs\_arg}$$

$$\frac{\Gamma \vdash_a arg \ : \ ty \Rightarrow ty' \quad \Gamma \vdash f \ : \ ty' \Rightarrow ty''}{\Gamma \vdash f \ arg \ : \ ty \Rightarrow ty''} \ \textsc{typing\_Trhs\_f}$$

$$\frac{\{l \ : \ ty\} \odot rty{=}rty' \quad \Gamma \vdash rty'}{\Gamma \vdash var.l \ : \ rty' \Rightarrow ty} \ \textsc{typing\_Trhs\_projection}$$

$$\frac{\Gamma \vdash rty' \quad \{l_1 \ : \ ty_1; \ .. \ ; \ l_n \ : \ ty_n\} \odot rty{=}rty'}{\Gamma \vdash \{var \ \texttt{with} \ l_1{=}var_1; \ .. \ ; \ l_n{=}var_n\} \ : \ \{var \ : \ rty'; \ var_1 \ : \ ty_1; \ .. \ ; \ var_n \ : \ ty_n\} \Rightarrow rty'} \ \textsc{typing\_Trhs\_upd}$$

$$\frac{}{\Gamma \vdash x_1 \ \texttt{+} \ x_2 \ : \ \{x_1 \ : \ \texttt{nat}; \ x_2 \ : \ \texttt{nat}\} \Rightarrow \texttt{nat}} \ \textsc{typing\_Trhs\_add\_nat\_nat}$$

$$\frac{}{\Gamma \vdash x_1 \ \texttt{+} \ x_2 \ : \ \{x_1 \ : \ \texttt{nat}; \ x_2 \ : \ \texttt{int}\} \Rightarrow \texttt{int}} \ \textsc{typing\_Trhs\_add\_nat\_int}$$

$$\frac{}{\Gamma \vdash x_1 \ \texttt{+} \ x_2 \ : \ \{x_1 \ : \ \texttt{int}; \ x_2 \ : \ \texttt{nat}\} \Rightarrow \texttt{int}} \ \textsc{typing\_Trhs\_add\_int\_nat}$$

$$\frac{}{\Gamma \vdash x_1 \ \texttt{+} \ x_2 \ : \ \{x_1 \ : \ \texttt{int}; \ x_2 \ : \ \texttt{int}\} \Rightarrow \texttt{int}} \ \textsc{typing\_Trhs\_add\_int\_int}$$

$$\frac{}{\Gamma \vdash x_1 \ \texttt{+} \ x_2 \ : \ \{x_1 \ : \ \texttt{timestamp}; \ x_2 \ : \ \texttt{int}\} \Rightarrow \texttt{timestamp}} \ \textsc{typing\_Trhs\_add\_time\_int}$$

$$\frac{}{\Gamma \vdash x_1 \ \texttt{+} \ x_2 \ : \ \{x_1 \ : \ \texttt{int}; \ x_2 \ : \ \texttt{timestamp}\} \Rightarrow \texttt{timestamp}} \ \textsc{typing\_Trhs\_add\_int\_time}$$

$$\frac{}{\Gamma \vdash x_1 \ \texttt{+} \ x_2 \ : \ \{x_1 \ : \ \texttt{mutez}; \ x_2 \ : \ \texttt{mutez}\} \Rightarrow \texttt{mutez}} \ \textsc{typing\_Trhs\_add\_mutez\_mutez}$$

$$\frac{}{\Gamma \vdash x_1 \ \texttt{-} \ x_2 \ : \ \{x_1 \ : \ \texttt{nat}; \ x_2 \ : \ \texttt{nat}\} \Rightarrow \texttt{int}} \ \textsc{typing\_Trhs\_sub\_nat\_nat}$$

$$\frac{}{\Gamma \vdash x_1 \ \texttt{-} \ x_2 \ : \ \{x_1 \ : \ \texttt{nat}; \ x_2 \ : \ \texttt{int}\} \Rightarrow \texttt{int}} \ \textsc{typing\_Trhs\_sub\_nat\_int}$$

$$\frac{}{\Gamma \vdash x_1 \ \texttt{-} \ x_2 \ : \ \{x_1 \ : \ \texttt{int}; \ x_2 \ : \ \texttt{nat}\} \Rightarrow \texttt{int}} \ \textsc{typing\_Trhs\_sub\_int\_nat}$$

$$\frac{}{\Gamma \vdash x_1 \ \texttt{-} \ x_2 \ : \ \{x_1 \ : \ \texttt{int}; \ x_2 \ : \ \texttt{int}\} \Rightarrow \texttt{int}} \ \textsc{typing\_Trhs\_sub\_int\_int}$$

$$\frac{}{\Gamma \vdash x_1 \ \texttt{-} \ x_2 \ : \ \{x_1 \ : \ \texttt{timestamp}; \ x_2 \ : \ \texttt{int}\} \Rightarrow \texttt{timestamp}} \ \textsc{typing\_Trhs\_sub\_time\_int}$$

$$\frac{}{\Gamma \vdash x_1 \ \texttt{-} \ x_2 \ : \ \{x_1 \ : \ \texttt{timestamp}; \ x_2 \ : \ \texttt{timestamp}\} \Rightarrow \texttt{int}} \ \textsc{typing\_Trhs\_sub\_time\_time}$$

$$\frac{}{\Gamma \vdash x_1 \texttt{ - } x_2 : \{x_1 : \texttt{mutez};\ x_2 : \texttt{mutez}\} \Rightarrow \texttt{mutez}}\ \textsc{typing\_Trhs\_sub\_mutez\_mutez}$$

$$\frac{}{\Gamma \vdash x_1 \texttt{ * } x_2 : \{x_1 : \texttt{nat};\ x_2 : \texttt{nat}\} \Rightarrow \texttt{nat}}\ \textsc{typing\_Trhs\_mul\_nat\_nat}$$

$$\frac{}{\Gamma \vdash x_1 \texttt{ * } x_2 : \{x_1 : \texttt{nat};\ x_2 : \texttt{int}\} \Rightarrow \texttt{int}}\ \textsc{typing\_Trhs\_mul\_nat\_int}$$

$$\frac{}{\Gamma \vdash x_1 \texttt{ * } x_2 : \{x_1 : \texttt{int};\ x_2 : \texttt{nat}\} \Rightarrow \texttt{int}}\ \textsc{typing\_Trhs\_mul\_int\_nat}$$

$$\frac{}{\Gamma \vdash x_1 \texttt{ * } x_2 : \{x_1 : \texttt{int};\ x_2 : \texttt{int}\} \Rightarrow \texttt{int}}\ \textsc{typing\_Trhs\_mul\_int\_int}$$

$$\frac{}{\Gamma \vdash x_1 \texttt{ * } x_2 : \{x_1 : \texttt{mutez};\ x_2 : \texttt{nat}\} \Rightarrow \texttt{mutez}}\ \textsc{typing\_Trhs\_mul\_mutez\_nat}$$

$$\frac{}{\Gamma \vdash x_1 \texttt{ * } x_2 : \{x_1 : \texttt{nat};\ x_2 : \texttt{mutez}\} \Rightarrow \texttt{mutez}}\ \textsc{typing\_Trhs\_mul\_nat\_mutez}$$

$$\frac{}{\Gamma \vdash x_1 \texttt{ /mod } x_2 : \{x_1 : \texttt{nat};\ x_2 : \texttt{nat}\} \Rightarrow \{\texttt{quotient : nat; remainder : nat}\}}\ \textsc{typing\_Trhs\_ediv\_nat}$$

$$\frac{}{\Gamma \vdash x_1 \texttt{ /mod } x_2 : \{x_1 : \texttt{nat};\ x_2 : \texttt{int}\} \Rightarrow \{\texttt{quotient : int; remainder : nat}\}}\ \textsc{typing\_Trhs\_ediv\_nat}$$

$$\frac{}{\Gamma \vdash x_1 \texttt{ /mod } x_2 : \{x_1 : \texttt{int};\ x_2 : \texttt{nat}\} \Rightarrow \{\texttt{quotient : int; remainder : nat}\}}\ \textsc{typing\_Trhs\_ediv\_int}$$

$$\frac{}{\Gamma \vdash x_1 \texttt{ /mod } x_2 : \{x_1 : \texttt{int};\ x_2 : \texttt{int}\} \Rightarrow \{\texttt{quotient : int; remainder : nat}\}}\ \textsc{typing\_Trhs\_ediv\_int}$$

$$\frac{}{\Gamma \vdash x_1 \texttt{ /mod } x_2 : \{x_1 : \texttt{mutez};\ x_2 : \texttt{nat}\} \Rightarrow \{\texttt{quotient : mutez; remainder : mutez}\}}\ \textsc{typing\_Trhs\_edi}$$

$$\frac{}{\Gamma \vdash x_1 \texttt{ /mod } x_2 : \{x_1 : \texttt{mutez};\ x_2 : \texttt{mutez}\} \Rightarrow \{\texttt{quotient : nat; remainder : mutez}\}}\ \textsc{typing\_Trhs\_edi}$$

$$\frac{}{\Gamma \vdash \texttt{ } x_1 : \{x_1 : \texttt{nat}\} \Rightarrow \texttt{int}}\ \textsc{typing\_Trhs\_not\_nat}$$

$$\frac{}{\Gamma \vdash \texttt{ } x_1 : \{x_1 : \texttt{int}\} \Rightarrow \texttt{int}}\ \textsc{typing\_Trhs\_not\_int}$$

$$\frac{}{\Gamma \vdash x_1 \&\& x_2 : \{x_1 : \texttt{nat};\ x_2 : \texttt{nat}\} \Rightarrow \texttt{nat}}\ \textsc{typing\_Trhs\_and\_nat\_nat}$$

$$\frac{}{\Gamma \vdash x_1 \&\& x_2 : \{x_1 : \texttt{int};\ x_2 : \texttt{nat}\} \Rightarrow \texttt{nat}}\ \textsc{typing\_Trhs\_and\_int\_nat}$$

$$\frac{}{\Gamma \vdash x_1 || x_2 : \{x_1 : \texttt{nat};\ x_2 : \texttt{nat}\} \Rightarrow \texttt{nat}}\ \textsc{typing\_Trhs\_or\_nat\_nat}$$

$$\frac{}{\Gamma \vdash x_1 \uparrow x_2 : \{x_1 : \texttt{nat};\ x_2 : \texttt{nat}\} \Rightarrow \texttt{nat}}\ \textsc{typing\_Trhs\_xor\_nat\_nat}$$

$$\frac{}{\Gamma \vdash x_1 << x_2 : \{x_1 : \texttt{nat};\ x_2 : \texttt{nat}\} \Rightarrow \texttt{nat}}\ \textsc{typing\_Trhs\_lsl}$$

$$\frac{}{\Gamma \vdash x_1 >> x_2 : \{x_1 : \texttt{nat};\ x_2 : \texttt{nat}\} \Rightarrow \texttt{nat}}\ \textsc{typing\_Trhs\_lsr}$$

$$\frac{}{\Gamma \vdash \texttt{abs}\ x_1 \ :\ \{x_1 \ :\ \texttt{int}\} \Rightarrow \texttt{nat}}\ \textsc{typing\_Trhs\_abs}$$

$$\frac{}{\Gamma \vdash \texttt{is\_nat}\ x_1 \ :\ \{x_1 \ :\ \texttt{int}\} \Rightarrow \texttt{option nat}}\ \textsc{typing\_Trhs\_is\_nat}$$

$$\frac{}{\Gamma \vdash \texttt{int}\ x_1 \ :\ \{x_1 \ :\ \texttt{nat}\} \Rightarrow \texttt{int}}\ \textsc{typing\_Trhs\_int}$$

$$\frac{\begin{array}{l}\{var \ :\ [constructor_1 \ :\ ty_1 \ \texttt{|}\ ..\ \texttt{|}\ constructor_k \ :\ ty_k]\} \odot rty{=}rty' \\ \{var_1 \ :\ ty_1\} \odot rty{=}rty_1 \quad .. \quad \{var_k \ :\ ty_k\} \odot rty{=}rty_k \\ \Gamma \vdash rhs_1 \ :\ rty_1 \Rightarrow ty \quad .. \quad \Gamma \vdash rhs_k \ :\ rty_k \Rightarrow ty\end{array}}{\Gamma \vdash \texttt{match}\ var\ \texttt{with}\ constructor_1\ var_1\ \texttt{->}\ rhs_1\ \texttt{|}\ ..\ \texttt{|}\ constructor_k\ var_k\ \texttt{->}\ rhs_k\ \texttt{end}\ :\ rty' \Rightarrow ty}\ \textsc{typing\_Trh}$$

$$\frac{}{\Gamma \vdash \ x_1 \ :\ \{x_1 \ :\ \texttt{bool}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_not}$$

$$\frac{}{\Gamma \vdash x_1 \&\& x_2 \ :\ \{x_1 \ :\ \texttt{bool};\ x_2 \ :\ \texttt{bool}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_and}$$

$$\frac{}{\Gamma \vdash x_1 || x_2 \ :\ \{x_1 \ :\ \texttt{bool};\ x_2 \ :\ \texttt{bool}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_or}$$

$$\frac{}{\Gamma \vdash x_1 \uparrow x_2 \ :\ \{x_1 \ :\ \texttt{bool};\ x_2 \ :\ \texttt{bool}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_xor}$$

$$\frac{}{\Gamma \vdash \texttt{concat}\ x_1\ x_2 \ :\ \{x_1 \ :\ \texttt{string};\ x_2 \ :\ \texttt{string}\} \Rightarrow \texttt{string}}\ \textsc{typing\_Trhs\_string\_concat}$$

$$\frac{}{\Gamma \vdash \texttt{size}\ x \ :\ \{x \ :\ \texttt{string}\} \Rightarrow \texttt{nat}}\ \textsc{typing\_Trhs\_string\_size}$$

$$\frac{}{\Gamma \vdash \texttt{slice}\ x\ x_1\ x_2 \ :\ \{x \ :\ \texttt{string};\ x_1 \ :\ \texttt{nat};\ x_2 \ :\ \texttt{nat}\} \Rightarrow \texttt{option string}}\ \textsc{typing\_Trhs\_string\_slice}$$

$$\frac{}{\Gamma \vdash \texttt{pack}\ x \ :\ \{x \ :\ ty\} \Rightarrow \texttt{bytes}}\ \textsc{typing\_Trhs\_bytes\_pack}$$

$$\frac{}{\Gamma \vdash \texttt{unpack}\ ty\ x \ :\ \{x \ :\ \texttt{bytes}\} \Rightarrow \texttt{option}\ ty}\ \textsc{typing\_Trhs\_bytes\_unpack}$$

$$\frac{}{\Gamma \vdash \texttt{concat}\ x_1\ x_2 \ :\ \{x_1 \ :\ \texttt{bytes};\ x_2 \ :\ \texttt{bytes}\} \Rightarrow \texttt{bytes}}\ \textsc{typing\_Trhs\_bytes\_concat}$$

$$\frac{}{\Gamma \vdash \texttt{size}\ x \ :\ \{x \ :\ \texttt{bytes}\} \Rightarrow \texttt{nat}}\ \textsc{typing\_Trhs\_bytes\_size}$$

$$\frac{}{\Gamma \vdash \texttt{slice}\ x\ x_1\ x_2 \ :\ \{x \ :\ \texttt{bytes};\ x_1 \ :\ \texttt{nat};\ x_2 \ :\ \texttt{nat}\} \Rightarrow \texttt{option bytes}}\ \textsc{typing\_Trhs\_bytes\_slice}$$

$$\frac{}{\Gamma \vdash var_1 \equiv var_2 \ :\ \{var_1 \ :\ \texttt{nat};\ var_2 \ :\ \texttt{nat}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_eq\_nat\_nat}$$

$$\frac{}{\Gamma \vdash var_1 \equiv var_2 \ :\ \{var_1 \ :\ \texttt{nat};\ var_2 \ :\ \texttt{int}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_eq\_nat\_int}$$

$$\frac{}{\Gamma \vdash var_1 \equiv var_2 \ :\ \{var_1 \ :\ \texttt{int};\ var_2 \ :\ \texttt{nat}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_eq\_int\_nat}$$

$$\frac{}{\Gamma \vdash var_1 \equiv var_2 : \{var_1 : \texttt{int};\ var_2 : \texttt{int}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_eq\_int\_int}$$

$$\frac{}{\Gamma \vdash var_1 \equiv var_2 : \{var_1 : \texttt{timestamp};\ var_2 : \texttt{timestamp}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_eq\_time\_time}$$

$$\frac{}{\Gamma \vdash var_1 \equiv var_2 : \{var_1 : \texttt{mutez};\ var_2 : \texttt{mutez}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_eq\_mutez\_mutez}$$

$$\frac{}{\Gamma \vdash var_1 \equiv var_2 : \{var_1 : \texttt{string};\ var_2 : \texttt{string}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_eq\_string\_string}$$

$$\frac{}{\Gamma \vdash var_1 \equiv var_2 : \{var_1 : \texttt{bytes};\ var_2 : \texttt{bytes}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_eq\_bytes\_bytes}$$

$$\frac{}{\Gamma \vdash var_1 \equiv var_2 : \{var_1 : \texttt{address};\ var_2 : \texttt{address}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_eq\_address\_address}$$

$$\frac{}{\Gamma \vdash var_1 \neq var_2 : \{var_1 : \texttt{nat};\ var_2 : \texttt{nat}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_neq\_nat\_nat}$$

$$\frac{}{\Gamma \vdash var_1 \neq var_2 : \{var_1 : \texttt{nat};\ var_2 : \texttt{int}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_neq\_nat\_int}$$

$$\frac{}{\Gamma \vdash var_1 \neq var_2 : \{var_1 : \texttt{int};\ var_2 : \texttt{nat}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_neq\_int\_nat}$$

$$\frac{}{\Gamma \vdash var_1 \neq var_2 : \{var_1 : \texttt{int};\ var_2 : \texttt{int}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_neq\_int\_int}$$

$$\frac{}{\Gamma \vdash var_1 \neq var_2 : \{var_1 : \texttt{timestamp};\ var_2 : \texttt{timestamp}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_neq\_time\_time}$$

$$\frac{}{\Gamma \vdash var_1 \neq var_2 : \{var_1 : \texttt{mutez};\ var_2 : \texttt{mutez}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_neq\_mutez\_mutez}$$

$$\frac{}{\Gamma \vdash var_1 \neq var_2 : \{var_1 : \texttt{string};\ var_2 : \texttt{string}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_neq\_string\_string}$$

$$\frac{}{\Gamma \vdash var_1 \neq var_2 : \{var_1 : \texttt{bytes};\ var_2 : \texttt{bytes}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_neq\_bytes\_bytes}$$

$$\frac{}{\Gamma \vdash var_1 \neq var_2 : \{var_1 : \texttt{address};\ var_2 : \texttt{address}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_neq\_address\_address}$$

$$\frac{}{\Gamma \vdash var_1 < var_2 : \{var_1 : \texttt{nat};\ var_2 : \texttt{nat}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_inf\_nat\_nat}$$

$$\frac{}{\Gamma \vdash var_1 < var_2 : \{var_1 : \texttt{nat};\ var_2 : \texttt{int}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_inf\_nat\_int}$$

$$\frac{}{\Gamma \vdash var_1 < var_2 : \{var_1 : \texttt{int};\ var_2 : \texttt{nat}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_inf\_int\_nat}$$

$$\frac{}{\Gamma \vdash var_1 < var_2 : \{var_1 : \texttt{int};\ var_2 : \texttt{int}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_inf\_int\_int}$$

$$\frac{}{\Gamma \vdash var_1 < var_2 : \{var_1 : \texttt{timestamp};\ var_2 : \texttt{timestamp}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_inf\_time\_time}$$

$$\frac{}{\Gamma \vdash var_1 < var_2 : \{var_1 : \texttt{mutez};\ var_2 : \texttt{mutez}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_inf\_mutez\_mutez}$$

$$\frac{}{\Gamma \vdash var_1 < var_2 : \{var_1 : \texttt{string};\ var_2 : \texttt{string}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_inf\_string\_string}$$

$$\frac{}{\Gamma \vdash var_1 < var_2 : \{var_1 : \texttt{bytes};\ var_2 : \texttt{bytes}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_inf\_bytes\_bytes}$$

$$\frac{}{\Gamma \vdash var_1 <= var_2 : \{var_1 : \texttt{nat};\ var_2 : \texttt{nat}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_infeq\_nat\_nat}$$

$$\frac{}{\Gamma \vdash var_1 <= var_2 : \{var_1 : \texttt{nat};\ var_2 : \texttt{int}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_infeq\_nat\_int}$$

$$\frac{}{\Gamma \vdash var_1 <= var_2 : \{var_1 : \texttt{int};\ var_2 : \texttt{nat}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_infeq\_int\_nat}$$

$$\frac{}{\Gamma \vdash var_1 <= var_2 : \{var_1 : \texttt{int};\ var_2 : \texttt{int}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_infeq\_int\_int}$$

$$\frac{}{\Gamma \vdash var_1 <= var_2 : \{var_1 : \texttt{timestamp};\ var_2 : \texttt{timestamp}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_infeq\_time\_time}$$

$$\frac{}{\Gamma \vdash var_1 <= var_2 : \{var_1 : \texttt{mutez};\ var_2 : \texttt{mutez}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_infeq\_mutez\_mutez}$$

$$\frac{}{\Gamma \vdash var_1 <= var_2 : \{var_1 : \texttt{string};\ var_2 : \texttt{string}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_infeq\_string\_string}$$

$$\frac{}{\Gamma \vdash var_1 <= var_2 : \{var_1 : \texttt{bytes};\ var_2 : \texttt{bytes}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_infeq\_bytes\_bytes}$$

$$\frac{}{\Gamma \vdash var_1 > var_2 : \{var_1 : \texttt{nat};\ var_2 : \texttt{nat}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_sup\_nat\_nat}$$

$$\frac{}{\Gamma \vdash var_1 > var_2 : \{var_1 : \texttt{nat};\ var_2 : \texttt{int}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_sup\_nat\_int}$$

$$\frac{}{\Gamma \vdash var_1 > var_2 : \{var_1 : \texttt{int};\ var_2 : \texttt{nat}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_sup\_int\_nat}$$

$$\frac{}{\Gamma \vdash var_1 > var_2 : \{var_1 : \texttt{int};\ var_2 : \texttt{int}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_sup\_int\_int}$$

$$\frac{}{\Gamma \vdash var_1 > var_2 : \{var_1 : \texttt{timestamp};\ var_2 : \texttt{timestamp}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_sup\_time\_time}$$

$$\frac{}{\Gamma \vdash var_1 > var_2 : \{var_1 : \texttt{mutez};\ var_2 : \texttt{mutez}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_sup\_mutez\_mutez}$$

$$\frac{}{\Gamma \vdash var_1 > var_2 : \{var_1 : \texttt{string};\ var_2 : \texttt{string}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_sup\_string\_string}$$

$$\frac{}{\Gamma \vdash var_1 > var_2 : \{var_1 : \texttt{bytes};\ var_2 : \texttt{bytes}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_sup\_bytes\_bytes}$$

$$\frac{}{\Gamma \vdash var_1 >= var_2 : \{var_1 : \texttt{nat};\ var_2 : \texttt{nat}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_supeq\_nat\_nat}$$

$$\frac{}{\Gamma \vdash var_1 >= var_2 : \{var_1 : \texttt{nat};\ var_2 : \texttt{int}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_supeq\_nat\_int}$$

$$\frac{}{\Gamma \vdash var_1 >= var_2 : \{var_1 : \texttt{int};\ var_2 : \texttt{nat}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_supeq\_int\_nat}$$

$$\frac{}{\Gamma \vdash var_1 >= var_2 : \{var_1 : \texttt{int};\ var_2 : \texttt{int}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_supeq\_int\_int}$$

$$\frac{}{\Gamma \vdash var_1 >= var_2 : \{var_1 : \texttt{timestamp};\ var_2 : \texttt{timestamp}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_supeq\_time\_time}$$

$$\frac{}{\Gamma \vdash var_1 >= var_2 : \{var_1 : \texttt{mutez};\ var_2 : \texttt{mutez}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_supeq\_mutez\_mutez}$$

$$\frac{}{\Gamma \vdash var_1 >= var_2 : \{var_1 : \texttt{string};\ var_2 : \texttt{string}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_supeq\_string\_string}$$

$$\frac{}{\Gamma \vdash var_1 >= var_2 : \{var_1 : \texttt{bytes};\ var_2 : \texttt{bytes}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Trhs\_supeq\_bytes\_bytes}$$

$$\frac{}{\Gamma \vdash [var_1;\ \ldots;\ var_n] : \{var_1 : ty;\ \ldots;\ var_n : ty\} \Rightarrow \texttt{list}\ ty}\ \textsc{typing\_Trhs\_list}$$

$$\frac{}{\Gamma \vdash x_1 :: x_2 : \{x_1 : ty;\ x_2 : \texttt{list}\ ty\} \Rightarrow \texttt{list}\ ty}\ \textsc{typing\_Trhs\_list\_cons}$$

$$\frac{}{\Gamma \vdash \texttt{size}\ x : \{x : \texttt{list}\ ty\} \Rightarrow \texttt{nat}}\ \textsc{typing\_Trhs\_list\_size}$$

$$\frac{\begin{array}{l}\{var : \texttt{list}\ ty\} \odot rty{=}rty' \\ \{var_1 : ty;\ var_2 : \texttt{list}\ ty\} \odot rty{=}rty'' \\ \Gamma \vdash rhs_1 : rty \Rightarrow ty' \\ \Gamma \vdash rhs_2 : rty'' \Rightarrow ty'\end{array}}{\Gamma \vdash \texttt{match}\ var\ \texttt{with}\ \texttt{[]}\ \texttt{->}\ rhs_1\ \texttt{|}\ var_1 :: var_2\ \texttt{->}\ rhs_2\ \texttt{end} : rty' \Rightarrow ty'}\ \textsc{typing\_Trhs\_list\_match}$$

$$\frac{\Gamma \vdash rhs : \{var : ty\} \Rightarrow ty'}{\Gamma \vdash \texttt{map}\ var\ \texttt{in}\ var'\ \texttt{do}\ rhs\ \texttt{done} : \{var' : \texttt{list}\ ty\} \Rightarrow \texttt{list}\ ty'}\ \textsc{typing\_Trhs\_list\_map}$$

$$\frac{}{\Gamma \vdash \{|var_1\ \texttt{|->}\ var_1';\ ..;\ var_n\ \texttt{|->}\ var_n'|\} : \{var_1 : ty;\ ..;\ var_n : ty;\ var_1' : ty';\ ..;\ var_n' : ty'\} \Rightarrow \texttt{map}\ ty\ ty'}$$

$$\frac{\Gamma \vdash rhs : \{var_1 : ty;\ var_2 : ty'\} \Rightarrow ty''}{\Gamma \vdash \texttt{map}\ var_1\ \texttt{|->}\ var_2\ \texttt{in}\ var_3\ \texttt{do}\ rhs\ \texttt{done} : \{var_3 : \texttt{map}\ ty\ ty'\} \Rightarrow \texttt{map}\ ty\ ty''}\ \textsc{typing\_Trhs\_map\_map}$$

$$\frac{}{\Gamma \vdash var_1[var_2] : \{var_1 : \texttt{map}\ ty\ ty';\ var_2 : ty\} \Rightarrow \texttt{option}\ ty'}\ \textsc{typing\_Trhs\_map\_get}$$

$$\frac{}{\Gamma \vdash \{|var_1\ \texttt{with}\ var_2\ \texttt{|->}\ var_3|\} : \{var_1 : \texttt{map}\ ty\ ty';\ var_2 : ty;\ var_3 : \texttt{option}\ ty'\} \Rightarrow \texttt{map}\ ty\ ty'}\ \textsc{typing\_T}$$

$$\frac{}{\Gamma \vdash \texttt{size}\ x : \{x : \texttt{map}\ ty\ ty'\} \Rightarrow \texttt{nat}}\ \textsc{typing\_Trhs\_map\_size}$$

$$\frac{}{\Gamma \vdash \texttt{check\_signature}\ var_1\ var_2\ var_3 : \{var_1 : \texttt{key};\ var_2 : \texttt{signature};\ var_3 : \texttt{bytes}\} \Rightarrow \texttt{bool}}\ \textsc{typing\_Tr}$$

$$\frac{}{\Gamma \vdash \texttt{amount} : \{\} \Rightarrow \texttt{mutez}}\ \textsc{typing\_Trhs\_amount}$$

$$\frac{}{\Gamma \vdash \texttt{self} : \{\} \Rightarrow \texttt{contract}\ ty}\ \textsc{typing\_Trhs\_self}$$

$$\frac{}{\Gamma \vdash \texttt{transfer\_tokens}\ var_1\ var_2\ var_3 : \{var_1 : ty;\ var_2 : \texttt{mutez};\ var_3 : \texttt{contract}\ ty\} \Rightarrow \texttt{operation}}\ \textsc{typing}$$

$$\frac{}{\Gamma \vdash \texttt{set\_delegate}\ var_1 : \{var_1 : \texttt{option key\_hash}\} \Rightarrow \texttt{operation}}\ \textsc{typing\_Trhs\_set\_delegate}$$

$$\frac{}{\Gamma \vdash \texttt{sender} : \{\} \Rightarrow \texttt{address}}\ \textsc{typing\_Trhs\_sender}$$

$$\frac{}{\Gamma \vdash \texttt{source} : \{\} \Rightarrow \texttt{address}}\ \textsc{typing\_Trhs\_source}$$

$$\frac{}{\Gamma \vdash \texttt{now} : \{\} \Rightarrow \texttt{timestamp}}\ \textsc{typing\_Trhs\_now}$$

$\boxed{\Gamma \vdash ty}$ Type well-formedness

$$\frac{\begin{array}{l}\texttt{sorted}\{l_1;\ ..;\ l_n\} \\ \Gamma \vdash ty_1 \quad .. \quad \Gamma \vdash ty_n \\ \Gamma \vdash\end{array}}{\Gamma \vdash \{l_1 : ty_1;\ ..;\ l_n : ty_n\}}\ \textsc{typing\_twf\_record}$$

$$\frac{\begin{array}{l}\Gamma \vdash \\ tvar{=}ty \in \Gamma\end{array}}{\Gamma \vdash tvar}\ \textsc{typing\_twf\_var}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \texttt{nat}}\ \textsc{typing\_twf\_nat}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \texttt{int}}\ \textsc{typing\_twf\_int}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \texttt{timestamp}}\ \textsc{typing\_twf\_timestamp}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \texttt{mutez}}\ \textsc{typing\_twf\_mutez}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \texttt{unit}}\ \textsc{typing\_twf\_unit}$$

$$\frac{\begin{array}{l}\Gamma \vdash ty \\ \Gamma \vdash ty'\end{array}}{\Gamma \vdash ty\ \texttt{*}\ ty'}\ \textsc{typing\_twf\_pair}$$

$$\frac{\begin{array}{l}\texttt{sorted}\{constructor_1;\ ..;\ constructor_k\} \\ \Gamma \vdash ty_1 \quad .. \quad \Gamma \vdash ty_k \\ \Gamma \vdash\end{array}}{\Gamma \vdash [constructor_1 : ty_1 \mid .. \mid constructor_k : ty_k]}\ \textsc{typing\_twf\_variant}$$

$$\frac{\Gamma \vdash ty}{\Gamma \vdash \texttt{option}\ ty}\ \text{TYPING\_TWF\_OPTION}$$

$$\frac{\Gamma \vdash ty \quad \Gamma \vdash ty'}{\Gamma \vdash ty\ \texttt{+}\ ty'}\ \text{TYPING\_TWF\_OR}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \texttt{bool}}\ \text{TYPING\_TWF\_BOOL}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \texttt{string}}\ \text{TYPING\_TWF\_STRING}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \texttt{bytes}}\ \text{TYPING\_TWF\_BYTES}$$

$$\frac{\Gamma \vdash ty}{\Gamma \vdash \texttt{list}\ ty}\ \text{TYPING\_TWF\_LIST}$$

$$\frac{\Gamma \vdash ty \quad \Gamma \vdash ty'}{\Gamma \vdash \texttt{map}\ ty\ ty'}\ \text{TYPING\_TWF\_MAP}$$

$$\frac{\Gamma \vdash ty \quad \Gamma \vdash ty'}{\Gamma \vdash \texttt{big\_map}\ ty\ ty'}\ \text{TYPING\_TWF\_BIG\_MAP}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \texttt{key}}\ \text{TYPING\_TWF\_KEY}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \texttt{signature}}\ \text{TYPING\_TWF\_SIGNATURE}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \texttt{operation}}\ \text{TYPING\_TWF\_OPERATION}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \texttt{key\_hash}}\ \text{TYPING\_TWF\_KEY\_HASH}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash \texttt{address}}\ \text{TYPING\_TWF\_ADDRESS}$$

$$\frac{\Gamma \vdash ty}{\Gamma \vdash \texttt{contract}\ ty}\ \text{TYPING\_TWF\_CONTRACT}$$

$$\frac{\Gamma \vdash ty \quad \Gamma \vdash ty'}{\Gamma \vdash ty\ \texttt{->}\ ty'}\ \text{TYPING\_TWF\_LAMBDA}$$

$\boxed{\Gamma \vdash f \ : \ ty \Rightarrow ty'}$ Function symbol typing

$$\frac{fvar \ : \ ty \Rightarrow ty' \ \in \ \Gamma}{\Gamma \vdash fvar \ : \ ty \Rightarrow ty'} \ \text{TYPING\_TF\_CONTEXT}$$

$$\frac{}{\Gamma \vdash \texttt{dup} \ : \ ty \Rightarrow ty \ \texttt{*} \ ty} \ \text{TYPING\_TF\_DUP}$$

$$\frac{\texttt{[}constructor \ : \ ty'\texttt{]} \odot vty'\texttt{=}vty}{\Gamma \vdash \texttt{(}constructor \ : \ vty\texttt{)} \ : \ ty' \Rightarrow vty} \ \text{TYPING\_TF\_CONSTRUCTOR}$$

$$\frac{}{\Gamma \vdash \texttt{hash\_key} \ : \ \texttt{key} \Rightarrow \texttt{key\_hash}} \ \text{TYPING\_TF\_HASH\_KEY}$$

$$\frac{}{\Gamma \vdash blake2b \ : \ \texttt{bytes} \Rightarrow \texttt{bytes}} \ \text{TYPING\_TF\_BLAKE2B}$$

$$\frac{}{\Gamma \vdash sha256 \ : \ \texttt{bytes} \Rightarrow \texttt{bytes}} \ \text{TYPING\_TF\_SHA256}$$

$$\frac{}{\Gamma \vdash sha512 \ : \ \texttt{bytes} \Rightarrow \texttt{bytes}} \ \text{TYPING\_TF\_SHA512}$$

$$\frac{}{\Gamma \vdash \texttt{contract}\, ty \ : \ \texttt{address} \Rightarrow \texttt{option (contract}\, ty\texttt{)}} \ \text{TYPING\_TF\_CONTRACT}$$

$$\frac{}{\Gamma \vdash \texttt{address} \ : \ \texttt{contract}\, ty \Rightarrow \texttt{address}} \ \text{TYPING\_TF\_ADDRESS}$$

$$\frac{}{\Gamma \vdash \texttt{implicit\_account} \ : \ \texttt{key\_hash} \Rightarrow \texttt{contract unit}} \ \text{TYPING\_TF\_IMPLICIT\_ACCOUNT}$$

$$\frac{}{\Gamma \vdash \texttt{exec} \ : \ \texttt{\{argument} \ : \ ty\texttt{;} \ \texttt{function} \ : \ ty \ \texttt{->} \ ty'\texttt{\}} \Rightarrow ty'} \ \text{TYPING\_TF\_EXEC}$$

$\boxed{\Gamma \vdash value \ : \ ty}$ Value typing

$$\frac{\Gamma \vdash val_1 \ : \ ty_1 \quad .. \quad \Gamma \vdash val_n \ : \ ty_n}{\Gamma \vdash \texttt{\{}l_1\texttt{=}val_1\texttt{;} \ .. \ \texttt{;} \ l_n\texttt{=}val_n\texttt{\}} \ : \ \texttt{\{}l_1 \ : \ ty_1\texttt{;} \ .. \ \texttt{;} \ l_n \ : \ ty_n\texttt{\}}} \ \text{TYPING\_TVAL\_RECORD}$$

$$\frac{}{\Gamma \vdash nat\_constant \ : \ \texttt{nat}} \ \text{TYPING\_TVAL\_NUM\_NAT}$$

$$\frac{}{\Gamma \vdash int\_constant \ : \ \texttt{int}} \ \text{TYPING\_TVAL\_NUM\_INT}$$

$$\frac{}{\Gamma \vdash mutez\_constant \ : \ \texttt{mutez}} \ \text{TYPING\_TVAL\_NUM\_MUTEZ}$$

$$\frac{}{\Gamma \vdash timestamp\_constant \ : \ \texttt{timestamp}} \ \text{TYPING\_TVAL\_NUM\_TIMESTAMP}$$

$$\frac{\begin{array}{c}\Gamma \vdash value \ : \ ty' \\ \texttt{[}constructor \ : \ ty'\texttt{]} \odot vty'\texttt{=}vty\end{array}}{\Gamma \vdash \texttt{(}constructor\, value \ : \ vty\texttt{)} \ : \ vty} \ \text{TYPING\_TVAL\_CONSTRUCTOR}$$

$$\frac{}{\Gamma \vdash \mathit{bool\_val} \ : \ \texttt{bool}} \quad \textsc{typing\_Tval\_bool}$$

$$\frac{}{\Gamma \vdash \mathit{sval} \ : \ \texttt{string}} \quad \textsc{typing\_Tval\_string}$$

$$\frac{}{\Gamma \vdash \mathit{bval} \ : \ \texttt{bytes}} \quad \textsc{typing\_Tval\_bytes}$$

$$\frac{\Gamma \vdash \mathit{val}_1 \ : \ \mathit{ty} \quad .. \quad \Gamma \vdash \mathit{val}_k \ : \ \mathit{ty}}{\Gamma \vdash (\texttt{[}\mathit{val}_1\texttt{;} \ ..\texttt{;} \ \mathit{val}_k\texttt{]} \ : \ \texttt{list}\,\mathit{ty}) \ : \ \texttt{list}\,\mathit{ty}} \quad \textsc{typing\_Tval\_list}$$

$$\frac{\begin{array}{c}\Gamma \vdash \mathit{value}_1 \ : \ \mathit{ty} \quad .. \quad \Gamma \vdash \mathit{value}_n \ : \ \mathit{ty} \\ \Gamma \vdash \mathit{value}'_1 \ : \ \mathit{ty} \quad .. \quad \Gamma \vdash \mathit{value}'_n \ : \ \mathit{ty}\end{array}}{\Gamma \vdash (\{|\mathit{val}_1 \ \texttt{|->} \ \mathit{val}'_1\texttt{;} \ ..\texttt{;} \ \mathit{val}_n \ \texttt{|->} \ \mathit{val}'_n|\} \ : \ \texttt{map}\,\mathit{ty}\,\mathit{ty}') \ : \ \texttt{map}\,\mathit{ty}\,\mathit{ty}'} \quad \textsc{typing\_Tval\_map}$$

$$\frac{}{\Gamma \vdash \texttt{key}\,\mathit{string\_litteral} \ : \ \texttt{key}} \quad \textsc{typing\_Tval\_key}$$

$$\frac{}{\Gamma \vdash \texttt{sig}\,\mathit{string\_litteral} \ : \ \texttt{signature}} \quad \textsc{typing\_Tval\_signature}$$

$$\frac{}{\Gamma \vdash \mathit{tz\_constant} \ : \ \texttt{key\_hash}} \quad \textsc{typing\_Tval\_tzlitteral}$$

$$\frac{}{\Gamma \vdash \mathit{kt\_constant} \ : \ \texttt{address}} \quad \textsc{typing\_Tval\_ktlitteral}$$

$$\frac{\Gamma \vdash \mathit{val} \ : \ \texttt{contract}\,\mathit{ty}}{\Gamma \vdash \mathit{val} \ : \ \texttt{address}} \quad \textsc{typing\_Tval\_contract\_to\_address}$$

$$\frac{\Gamma \vdash \mathit{val} \ : \ \texttt{key\_hash}}{\Gamma \vdash \mathit{val} \ : \ \texttt{contract unit}} \quad \textsc{typing\_Tval\_key\_hash\_to\_contract}$$

$$\frac{\Gamma \vdash \mathit{instruction} \ : \ \mathit{ty} \Rightarrow \mathit{ty}'}{\Gamma \vdash \texttt{lambda}\,\mathit{ty} \ : \ \mathit{ty}'.\mathit{instruction}\,\texttt{end} \ : \ \mathit{ty} \ \texttt{->} \ \mathit{ty}'} \quad \textsc{typing\_Tval\_lambda}$$

$\boxed{\mathit{tvar}\ \texttt{not free in}\ \mathit{ty}}$ Type variable freshness

$$\frac{\mathit{tvar}\ \texttt{not free in}\ \mathit{ty}_1 \quad .. \quad \mathit{tvar}\ \texttt{not free in}\ \mathit{ty}_n}{\mathit{tvar}\ \texttt{not free in}\ \{l_1 \ : \ \mathit{ty}_1\texttt{;} \ ..\texttt{;} \ l_n \ : \ \mathit{ty}_n\texttt{}\}} \quad \textsc{typing\_Tfresh\_tvarrecord}$$

$$\frac{}{\mathit{tvar}\ \texttt{not free in nat}} \quad \textsc{typing\_Tfresh\_tvarnat}$$

$$\frac{}{\mathit{tvar}\ \texttt{not free in int}} \quad \textsc{typing\_Tfresh\_tvarint}$$

$$\frac{}{\mathit{tvar}\ \texttt{not free in timestamp}} \quad \textsc{typing\_Tfresh\_tvartimestamp}$$

$$\frac{}{\mathit{tvar}\ \texttt{not free in mutez}} \quad \textsc{typing\_Tfresh\_tvarmutez}$$

$$\frac{}{\mathit{tvar}\ \texttt{not free in unit}} \quad \textsc{typing\_Tfresh\_tvarunit}$$

$$\frac{tvar\ \texttt{not free in}\ ty \qquad tvar\ \texttt{not free in}\ ty'}{tvar\ \texttt{not free in}\ ty\ \texttt{*}\ ty'}\ \textsc{typing\_Tfresh\_tvarpair}$$

$$\frac{tvar\ \texttt{not free in}\ ty_1 \quad .. \quad tvar\ \texttt{not free in}\ ty_k}{tvar\ \texttt{not free in [}constructor_1\ \texttt{:}\ ty_1\ \texttt{|}\ ..\ \texttt{|}\ constructor_k\ \texttt{:}\ ty_k\texttt{]}}\ \textsc{typing\_Tfresh\_tvarvariant}$$

$$\frac{tvar\ \texttt{not free in}\ ty}{tvar\ \texttt{not free in option}\ ty}\ \textsc{typing\_Tfresh\_tvaroption}$$

$$\frac{tvar\ \texttt{not free in}\ ty \qquad tvar\ \texttt{not free in}\ ty'}{tvar\ \texttt{not free in}\ ty\ \texttt{+}\ ty'}\ \textsc{typing\_Tfresh\_tvaror}$$

$$\frac{}{tvar\ \texttt{not free in bool}}\ \textsc{typing\_Tfresh\_tvarbool}$$

$$\frac{}{tvar\ \texttt{not free in string}}\ \textsc{typing\_Tfresh\_tvarstring}$$

$$\frac{}{tvar\ \texttt{not free in bytes}}\ \textsc{typing\_Tfresh\_tvarbytes}$$

$$\frac{tvar\ \texttt{not free in}\ ty}{tvar\ \texttt{not free in list}\ ty}\ \textsc{typing\_Tfresh\_tvarlist}$$

$$\frac{tvar\ \texttt{not free in}\ ty \qquad tvar\ \texttt{not free in}\ ty'}{tvar\ \texttt{not free in map}\ ty\ ty'}\ \textsc{typing\_Tfresh\_tvarmap}$$

$$\frac{tvar\ \texttt{not free in}\ ty \qquad tvar\ \texttt{not free in}\ ty'}{tvar\ \texttt{not free in big\_map}\ ty\ ty'}\ \textsc{typing\_Tfresh\_tvarbig\_map}$$

$$\frac{}{tvar\ \texttt{not free in key}}\ \textsc{typing\_Tfresh\_tvarkey}$$

$$\frac{}{tvar\ \texttt{not free in signature}}\ \textsc{typing\_Tfresh\_tvarsignature}$$

$$\frac{}{tvar\ \texttt{not free in operation}}\ \textsc{typing\_Tfresh\_tvaroperation}$$

$$\frac{}{tvar\ \texttt{not free in key\_hash}}\ \textsc{typing\_Tfresh\_tvarkey\_hash}$$

$$\frac{}{tvar\ \texttt{not free in address}}\ \textsc{typing\_Tfresh\_tvaraddress}$$

$$\frac{tvar\ \texttt{not free in}\ ty}{tvar\ \texttt{not free in contract}\ ty}\ \textsc{typing\_Tfresh\_tvarcontract}$$

$$\frac{tvar\ \texttt{not free in}\ ty \qquad tvar\ \texttt{not free in}\ ty'}{tvar\ \texttt{not free in}\ ty\ \texttt{->}\ ty'}\ \textsc{typing\_Tfresh\_tvarlambda}$$

C Albert semantics

$\boxed{E \vdash lhs/val \texttt{ -> } val'}$ Left-hand side evaluation

$$\frac{}{E \vdash var/val \texttt{ -> \{}var\texttt{=}val\texttt{\}}} \text{ EVAL\_LHS\_VAR}$$

$$\frac{}{E \vdash \texttt{\{}l_1\texttt{=}x_1\texttt{;}\ ..\ \texttt{;}\ l_n\texttt{=}x_n\texttt{\}/\{}l_1\texttt{=}val_1\texttt{;}\ ..\ \texttt{;}\ l_n\texttt{=}val_n\texttt{\} -> \{}x_1\texttt{=}val_1\texttt{;}\ ..\ \texttt{;}\ x_n\texttt{=}val_n\texttt{\}}} \text{ EVAL\_LHS\_RECORD}$$

$\boxed{E \vdash arg/aval \texttt{ -> } val'}$ Argument evaluation

$$\frac{}{E \vdash var/a\texttt{\{}var\texttt{=}val\texttt{\} -> }val} \text{ EVAL\_ARG\_VAR}$$

$$\frac{}{E \vdash val/a\texttt{\{\} -> }val} \text{ EVAL\_ARG\_VAL}$$

$$\frac{}{E \vdash \texttt{\{}l_1\texttt{=}x_1\texttt{;}\ ..\ \texttt{;}\ l_n\texttt{=}x_n\texttt{\}/}a\texttt{\{}x_1\texttt{=}val_1\texttt{;}\ ..\ \texttt{;}\ x_n\texttt{=}val_n\texttt{\} -> \{}l_1\texttt{=}val_1\texttt{;}\ ..\ \texttt{;}\ l_n\texttt{=}val_n\texttt{\}}} \text{ EVAL\_ARG\_RECORD}$$

$\boxed{fvar\texttt{=}instruction \in E}$ Looking for a function declaration in context

$$\frac{}{fvar\texttt{=}instruction \in E, fvar\texttt{=}instruction} \text{ EVAL\_IN\_E\_FHEAD}$$

$$\frac{\begin{array}{l} fvar\texttt{=}instruction \in E \\ fvar \neq fvar' \end{array}}{fvar\texttt{=}instruction \in E, fvar'\texttt{=}instruction'} \text{ EVAL\_IN\_E\_FTAIL}$$

$\boxed{E \vdash rhs/val \texttt{ -> } val'}$ Right-hand side evaluation

$$\frac{E \vdash arg/aval \texttt{ -> } val'}{E \vdash arg/val \texttt{ -> } val'} \text{ EVAL\_RHS\_ARG}$$

$$\frac{\begin{array}{l} E \vdash arg/aval \texttt{ -> } val' \\ E \vdash f/val' \texttt{ -> } val'' \end{array}}{E \vdash f\ arg/val \texttt{ -> } val''} \text{ EVAL\_RHS\_F}$$

$$\frac{\texttt{\{}l\texttt{=}val\texttt{\}} \odot rval\texttt{=}rval'}{E \vdash var.l/rval' \texttt{ -> } val} \text{ EVAL\_RHS\_PROJECTION}$$

$$\frac{\begin{array}{l} \texttt{\{}l_1\texttt{=}val_1'\texttt{;}\ ..\ \texttt{;}\ l_n\texttt{=}val_n'\texttt{\}} \odot rval\texttt{=}rval' \\ \texttt{\{}l_1\texttt{=}val_1\texttt{;}\ ..\ \texttt{;}\ l_n\texttt{=}val_n\texttt{\}} \odot rval\texttt{=}rval'' \end{array}}{E \vdash \texttt{\{}var\ \texttt{with}\ l_1\texttt{=}var_1\texttt{;}\ ..\ \texttt{;}\ l_n\texttt{=}var_n\texttt{\}/\{}var\texttt{=}rval'\texttt{;}\ var_1\texttt{=}val_1\texttt{;}\ ..\ \texttt{;}\ var_n\texttt{=}val_n\texttt{\} -> }rval''} \text{ EVAL\_RHS\_UPDATE}$$

$$\frac{}{E \vdash x_1\ \texttt{+}\ x_2\texttt{/\{}x_1\texttt{=}nv_1\texttt{;}\ x_2\texttt{=}nv_2\texttt{\} -> }nv_1\ \texttt{+}\ nv_2} \text{ EVAL\_RHS\_ADD}$$

$$\frac{}{E \vdash x_1\ \texttt{-}\ x_2\texttt{/\{}x_1\texttt{=}nv_1\texttt{;}\ x_2\texttt{=}nv_2\texttt{\} -> }nv_1\ \texttt{-}\ nv_2} \text{ EVAL\_RHS\_SUB}$$

$$\frac{}{E \vdash x_1 \texttt{ * } x_2/\{x_1\texttt{=}nv_1\texttt{ ; } x_2\texttt{=}nv_2\texttt{\}} \texttt{ -> } nv_1 \texttt{ * } nv_2}\ \text{EVAL\_RHS\_MUL}$$

$$\frac{}{E \vdash x_1 \texttt{ /mod } x_2/\{x_1\texttt{=}nv_1\texttt{ ; } x_2\texttt{=}nv_2\texttt{\}} \texttt{ -> \{quotient=}nv_1/nv_2\texttt{; remainder=}nv_1\%nv_2\texttt{\}}}\ \text{EVAL\_RHS\_EDIV}$$

$$\frac{}{E \vdash \texttt{ } x_1/\{x_1\texttt{=}nv_1\texttt{\}} \texttt{ -> } \texttt{ } nv_1}\ \text{EVAL\_RHS\_NOT\_NAT}$$

$$\frac{}{E \vdash x_1\&\&x_2/\{x_1\texttt{=}nv_1\texttt{ ; } x_2\texttt{=}nv_2\texttt{\}} \texttt{ -> } nv_1\&\&nv_2}\ \text{EVAL\_RHS\_AND\_INT\_NAT}$$

$$\frac{}{E \vdash x_1||x_2/\{x_1\texttt{=}nv_1\texttt{ ; } x_2\texttt{=}nv_2\texttt{\}} \texttt{ -> } nv_1||nv_2}\ \text{EVAL\_RHS\_OR\_NAT\_NAT}$$

$$\frac{}{E \vdash x_1\uparrow x_2/\{x_1\texttt{=}nv_1\texttt{ ; } x_2\texttt{=}nv_2\texttt{\}} \texttt{ -> } nv_1\uparrow nv_2}\ \text{EVAL\_RHS\_XOR\_NAT\_NAT}$$

$$\frac{nv_2 <= 256}{E \vdash x_1 << x_2/\{x_1\texttt{=}nv_1\texttt{ ; } x_2\texttt{=}nv_2\texttt{\}} \texttt{ -> } nv_1 << nv_2}\ \text{EVAL\_RHS\_LSL\_OK}$$

$$\frac{nv_2 <= 256}{E \vdash x_1 >> x_2/\{x_1\texttt{=}nv_1\texttt{ ; } x_2\texttt{=}nv_2\texttt{\}} \texttt{ -> } nv_1 >> nv_2}\ \text{EVAL\_RHS\_LSR\_OK}$$

$$\frac{}{E \vdash \texttt{abs } x_1/\{x_1\texttt{=}nv_1\texttt{\}} \texttt{ -> abs } nv_1}\ \text{EVAL\_RHS\_ABS}$$

$$\frac{nv_1 >= 0}{E \vdash \texttt{is\_nat } x_1/\{x_1\texttt{=}nv_1\texttt{\}} \texttt{ -> (Some } nv_1 \texttt{ : [None : unit | Some : nat])}}\ \text{EVAL\_RHS\_IS\_NAT\_SOME}$$

$$\frac{nv_1 < 0}{E \vdash \texttt{is\_nat } x_1/\{x_1\texttt{=}nv_1\texttt{\}} \texttt{ -> (None\{\} : [None : unit | Some : nat])}}\ \text{EVAL\_RHS\_IS\_NAT\_NONE}$$

$$\frac{}{E \vdash \texttt{int } x_1/\{x_1\texttt{=}nv_1\texttt{\}} \texttt{ -> } nv_1}\ \text{EVAL\_RHS\_INT}$$

$$\frac{E \vdash rhs/\{var'\texttt{=}val'\texttt{\}} \texttt{ -> } val}{E \vdash \texttt{match } var \texttt{ with } constructor\ var' \texttt{ -> } rhs \texttt{ | } < branches\_rhs > \texttt{ end/\{}var\texttt{=(}constructor\ val' \texttt{ : } vty\texttt{)\}} \texttt{ -> } val}$$

$$\frac{constructor <> constructor' \quad E \vdash \texttt{match } var \texttt{ with } branches\_rhs \texttt{ end/\{}var\texttt{=(}constructor\ val' \texttt{ : } vty\texttt{)\}} \texttt{ -> } val'}{E \vdash \texttt{match } var \texttt{ with } constructor'\ var' \texttt{ -> } rhs \texttt{ | } < branches\_rhs > \texttt{ end/\{}var\texttt{=(}constructor\ val' \texttt{ : } vty\texttt{)\}} \texttt{ -> } val}$$

$$\frac{}{E \vdash \texttt{ } x_1/\{x_1\texttt{=}bool\_val_1\texttt{\}} \texttt{ -> } \texttt{ } bool\_val_1}\ \text{EVAL\_RHS\_NOT}$$

$$\frac{}{E \vdash x_1\&\&x_2/\{x_1\texttt{=}bool\_val_1\texttt{ ; } x_2\texttt{=}bool\_val_2\texttt{\}} \texttt{ -> } bool\_val_1\&\&bool\_val_2}\ \text{EVAL\_RHS\_AND}$$

$$\frac{}{E \vdash x_1||x_2/\{x_1\texttt{=}bool\_val_1\texttt{ ; } x_2\texttt{=}bool\_val_2\texttt{\}} \texttt{ -> } bool\_val_1||bool\_val_2}\ \text{EVAL\_RHS\_OR}$$

$$\frac{}{E \vdash x_1\uparrow x_2/\{x_1\texttt{=}bool\_val_1\texttt{ ; } x_2\texttt{=}bool\_val_2\texttt{\}} \texttt{ -> } bool\_val_1\uparrow bool\_val_2}\ \text{EVAL\_RHS\_XOR}$$

$$\frac{}{E \vdash \texttt{concat}\ x_1\ x_2/\{x_1{=}sval_1\texttt{;}\ x_2{=}sval_2\}\ \texttt{->}\ sval_1 + sval_2}\ \textsc{eval\_rhs\_string\_concat}$$

$$\frac{}{E \vdash \texttt{size}\ x/\{x{=}sval\}\ \texttt{->}\ |sval|}\ \textsc{eval\_rhs\_string\_size}$$

$$\frac{\begin{array}{c} nv_1 <= nv_2 \\ nv_2 <= |sval| \end{array}}{E \vdash \texttt{slice}\ x\ x_1\ x_2/\{x{=}sval\texttt{;}\ x_1{=}nv_1\texttt{;}\ x_2{=}nv_2\}\ \texttt{->}\ (\texttt{Some}\ (\texttt{slice}\ sval\ nv_1\ nv_2)\ \texttt{:}\ [\texttt{None}\ \texttt{:}\ \texttt{unit}\ \texttt{|}\ \texttt{Some}\ \texttt{:}\ \texttt{string}]}$$

$$\frac{nv_1 > nv_2}{E \vdash \texttt{slice}\ x\ x_1\ x_2/\{x{=}sval\texttt{;}\ x_1{=}nv_1\texttt{;}\ x_2{=}nv_2\}\ \texttt{->}\ (\texttt{None}\ \{\}\ \texttt{:}\ [\texttt{None}\ \texttt{:}\ \texttt{unit}\ \texttt{|}\ \texttt{Some}\ \texttt{:}\ \texttt{string}])}\ \textsc{eval\_rhs\_s}$$

$$\frac{nv_2\ \texttt{+}\ nv_2 > |sval|}{E \vdash \texttt{slice}\ x\ x_1\ x_2/\{x{=}sval\texttt{;}\ x_1{=}nv_1\texttt{;}\ x_2{=}nv_2\}\ \texttt{->}\ (\texttt{None}\ \{\}\ \texttt{:}\ [\texttt{None}\ \texttt{:}\ \texttt{unit}\ \texttt{|}\ \texttt{Some}\ \texttt{:}\ \texttt{string}])}\ \textsc{eval\_rhs\_s}$$

$$\frac{}{E \vdash \texttt{pack}\ x/\{x{=}val\}\ \texttt{->}\ \texttt{pack}\ val}\ \textsc{eval\_rhs\_bytes\_pack}$$

$$\frac{\Gamma \vdash \texttt{unpack}\ bval\ \texttt{:}\ ty}{E \vdash \texttt{unpack}\ ty\ x/\{x{=}bval\}\ \texttt{->}\ (\texttt{Some}\ (\texttt{unpack}\ bval)\ \texttt{:}\ [\texttt{None}\ \texttt{:}\ \texttt{unit}\ \texttt{|}\ \texttt{Some}\ \texttt{:}\ ty])}\ \textsc{eval\_rhs\_bytes\_unp}$$

$$\frac{(\Gamma \vdash \texttt{unpack}\ bval\ \texttt{:}\ ty)}{E \vdash \texttt{unpack}\ ty\ x/\{x{=}bval\}\ \texttt{->}\ (\texttt{None}\ \{\}\ \texttt{:}\ [\texttt{None}\ \texttt{:}\ \texttt{unit}\ \texttt{|}\ \texttt{Some}\ \texttt{:}\ ty])}\ \textsc{eval\_rhs\_bytes\_unpack\_fail}$$

$$\frac{}{E \vdash \texttt{concat}\ x_1\ x_2/\{x_1{=}bval_1\texttt{;}\ x_2{=}bval_2\}\ \texttt{->}\ bval_1 + bval_2}\ \textsc{eval\_rhs\_bytes\_concat}$$

$$\frac{}{E \vdash \texttt{size}\ x/\{x{=}bval\}\ \texttt{->}\ |bval|}\ \textsc{eval\_rhs\_bytes\_size}$$

$$\frac{\begin{array}{c} nv_1 <= nv_2 \\ nv_2 <= |bval| \end{array}}{E \vdash \texttt{slice}\ x\ x_1\ x_2/\{x{=}bval\texttt{;}\ x_1{=}nv_1\texttt{;}\ x_2{=}nv_2\}\ \texttt{->}\ (\texttt{Some}\ (\texttt{slice}\ bval\ nv_1\ nv_2)\ \texttt{:}\ [\texttt{None}\ \texttt{:}\ \texttt{unit}\ \texttt{|}\ \texttt{Some}\ \texttt{:}\ \texttt{bytes}])}$$

$$\frac{nv_1 > nv_2}{E \vdash \texttt{slice}\ x\ x_1\ x_2/\{x{=}bval\texttt{;}\ x_1{=}nv_1\texttt{;}\ x_2{=}nv_2\}\ \texttt{->}\ (\texttt{None}\ \{\}\ \texttt{:}\ [\texttt{None}\ \texttt{:}\ \texttt{unit}\ \texttt{|}\ \texttt{Some}\ \texttt{:}\ \texttt{bytes}])}\ \textsc{eval\_rhs\_by}$$

$$\frac{nv_2\ \texttt{+}\ nv_2 > |bval|}{E \vdash \texttt{slice}\ x\ x_1\ x_2/\{x{=}bval\texttt{;}\ x_1{=}nv_1\texttt{;}\ x_2{=}nv_2\}\ \texttt{->}\ (\texttt{None}\ \{\}\ \texttt{:}\ [\texttt{None}\ \texttt{:}\ \texttt{unit}\ \texttt{|}\ \texttt{Some}\ \texttt{:}\ \texttt{bytes}])}\ \textsc{eval\_rhs\_by}$$

$$\frac{nv_1 \equiv nv_2}{E \vdash x_1 \equiv x_2/\{x_1{=}nv_1\texttt{;}\ x_2{=}nv_2\}\ \texttt{->}\ \texttt{true}}\ \textsc{eval\_rhs\_eq\_num}$$

$$\frac{nv_1 \neq nv_2}{E \vdash x_1 \equiv x_2/\{x_1{=}nv_1\texttt{;}\ x_2{=}nv_2\}\ \texttt{->}\ \texttt{false}}\ \textsc{eval\_rhs\_eq\_num\_not}$$

$$\frac{sval_1 \equiv sval_2}{E \vdash x_1 \equiv x_2/\{x_1{=}sval_1\texttt{;}\ x_2{=}sval_2\}\ \texttt{->}\ \texttt{true}}\ \textsc{eval\_rhs\_eq\_str}$$

$$\frac{sval_1 \neq sval_2}{E \vdash x_1 \equiv x_2/\{x_1\texttt{=}sval_1\texttt{;}\ x_2\texttt{=}sval_2\texttt{\}}\ \texttt{->}\ \texttt{false}}\ \text{EVAL\_RHS\_EQ\_STR\_NOT}$$

$$\frac{bval_1 \equiv bval_2}{E \vdash x_1 \equiv x_2/\{x_1\texttt{=}bval_1\texttt{;}\ x_2\texttt{=}bval_2\texttt{\}}\ \texttt{->}\ \texttt{true}}\ \text{EVAL\_RHS\_EQ\_BYTES}$$

$$\frac{bval_1 \neq bval_2}{E \vdash x_1 \equiv x_2/\{x_1\texttt{=}bval_1\texttt{;}\ x_2\texttt{=}bval_2\texttt{\}}\ \texttt{->}\ \texttt{false}}\ \text{EVAL\_RHS\_EQ\_BYTES\_NOT}$$

$$\frac{nv_1 \neq nv_2}{E \vdash x_1 \neq x_2/\{x_1\texttt{=}nv_1\texttt{;}\ x_2\texttt{=}nv_2\texttt{\}}\ \texttt{->}\ \texttt{true}}\ \text{EVAL\_RHS\_NEQ\_NUM}$$

$$\frac{nv_1 \equiv nv_2}{E \vdash x_1 \neq x_2/\{x_1\texttt{=}nv_1\texttt{;}\ x_2\texttt{=}nv_2\texttt{\}}\ \texttt{->}\ \texttt{false}}\ \text{EVAL\_RHS\_NEQ\_NUM\_NOT}$$

$$\frac{nv_1 < nv_2}{E \vdash x_1 < x_2/\{x_1\texttt{=}nv_1\texttt{;}\ x_2\texttt{=}nv_2\texttt{\}}\ \texttt{->}\ \texttt{true}}\ \text{EVAL\_RHS\_INF\_NUM}$$

$$\frac{nv_1 >= nv_2}{E \vdash x_1 < x_2/\{x_1\texttt{=}nv_1\texttt{;}\ x_2\texttt{=}nv_2\texttt{\}}\ \texttt{->}\ \texttt{false}}\ \text{EVAL\_RHS\_INF\_NUM\_NOT}$$

$$\frac{sval_1 < sval_2}{E \vdash x_1 < x_2/\{x_1\texttt{=}sval_1\texttt{;}\ x_2\texttt{=}sval_2\texttt{\}}\ \texttt{->}\ \texttt{true}}\ \text{EVAL\_RHS\_INF\_STR}$$

$$\frac{sval_1 >= sval_2}{E \vdash x_1 < x_2/\{x_1\texttt{=}sval_1\texttt{;}\ x_2\texttt{=}sval_2\texttt{\}}\ \texttt{->}\ \texttt{false}}\ \text{EVAL\_RHS\_INF\_STR\_NOT}$$

$$\frac{bval_1 < bval_2}{E \vdash x_1 < x_2/\{x_1\texttt{=}bval_1\texttt{;}\ x_2\texttt{=}bval_2\texttt{\}}\ \texttt{->}\ \texttt{true}}\ \text{EVAL\_RHS\_INF\_BYTES}$$

$$\frac{bval_1 >= bval_2}{E \vdash x_1 < x_2/\{x_1\texttt{=}bval_1\texttt{;}\ x_2\texttt{=}bval_2\texttt{\}}\ \texttt{->}\ \texttt{false}}\ \text{EVAL\_RHS\_INF\_BYTES\_NOT}$$

$$\frac{nv_1 <= nv_2}{E \vdash x_1 <= x_2/\{x_1\texttt{=}nv_1\texttt{;}\ x_2\texttt{=}nv_2\texttt{\}}\ \texttt{->}\ \texttt{true}}\ \text{EVAL\_RHS\_INFEQ\_NUM}$$

$$\frac{nv_1 > nv_2}{E \vdash x_1 <= x_2/\{x_1\texttt{=}nv_1\texttt{;}\ x_2\texttt{=}nv_2\texttt{\}}\ \texttt{->}\ \texttt{false}}\ \text{EVAL\_RHS\_INFEQ\_NUM\_NOT}$$

$$\frac{sval_1 <= sval_2}{E \vdash x_1 <= x_2/\{x_1\texttt{=}sval_1\texttt{;}\ x_2\texttt{=}sval_2\texttt{\}}\ \texttt{->}\ \texttt{true}}\ \text{EVAL\_RHS\_INFEQ\_STR}$$

$$\frac{sval_1 > sval_2}{E \vdash x_1 <= x_2/\{x_1\texttt{=}sval_1\texttt{;}\ x_2\texttt{=}sval_2\texttt{\}}\ \texttt{->}\ \texttt{false}}\ \text{EVAL\_RHS\_INFEQ\_STR\_NOT}$$

$$\frac{bval_1 <= bval_2}{E \vdash x_1 <= x_2/\{x_1\texttt{=}bval_1\texttt{;}\ x_2\texttt{=}bval_2\texttt{\}}\ \texttt{->}\ \texttt{true}}\ \text{EVAL\_RHS\_INFEQ\_BYTES}$$

$$\frac{bval_1 > bval_2}{E \vdash x_1 <= x_2/\{x_1{=}bval_1\,;\ x_2{=}bval_2\}\ \texttt{->}\ \texttt{false}}\ \textsc{Eval\_Rhs\_Infeq\_Bytes\_Not}$$

$$\frac{nv_1 > nv_2}{E \vdash x_1 > x_2/\{x_1{=}nv_1\,;\ x_2{=}nv_2\}\ \texttt{->}\ \texttt{true}}\ \textsc{Eval\_Rhs\_Sup\_Num}$$

$$\frac{nv_1 <= nv_2}{E \vdash x_1 > x_2/\{x_1{=}nv_1\,;\ x_2{=}nv_2\}\ \texttt{->}\ \texttt{false}}\ \textsc{Eval\_Rhs\_Sup\_Num\_Not}$$

$$\frac{sval_1 > sval_2}{E \vdash x_1 > x_2/\{x_1{=}sval_1\,;\ x_2{=}sval_2\}\ \texttt{->}\ \texttt{true}}\ \textsc{Eval\_Rhs\_Sup\_Str}$$

$$\frac{sval_1 <= sval_2}{E \vdash x_1 > x_2/\{x_1{=}sval_1\,;\ x_2{=}sval_2\}\ \texttt{->}\ \texttt{false}}\ \textsc{Eval\_Rhs\_Sup\_Str\_Not}$$

$$\frac{bval_1 > bval_2}{E \vdash x_1 > x_2/\{x_1{=}bval_1\,;\ x_2{=}bval_2\}\ \texttt{->}\ \texttt{true}}\ \textsc{Eval\_Rhs\_Sup\_Bytes}$$

$$\frac{bval_1 <= bval_2}{E \vdash x_1 > x_2/\{x_1{=}bval_1\,;\ x_2{=}bval_2\}\ \texttt{->}\ \texttt{false}}\ \textsc{Eval\_Rhs\_Sup\_Bytes\_Not}$$

$$\frac{nv_1 >= nv_2}{E \vdash x_1 >= x_2/\{x_1{=}nv_1\,;\ x_2{=}nv_2\}\ \texttt{->}\ \texttt{true}}\ \textsc{Eval\_Rhs\_Supeq\_Num}$$

$$\frac{nv_1 < nv_2}{E \vdash x_1 >= x_2/\{x_1{=}nv_1\,;\ x_2{=}nv_2\}\ \texttt{->}\ \texttt{false}}\ \textsc{Eval\_Rhs\_Supeq\_Num\_Not}$$

$$\frac{sval_1 >= sval_2}{E \vdash x_1 >= x_2/\{x_1{=}sval_1\,;\ x_2{=}sval_2\}\ \texttt{->}\ \texttt{true}}\ \textsc{Eval\_Rhs\_Supeq\_Str}$$

$$\frac{sval_1 < sval_2}{E \vdash x_1 >= x_2/\{x_1{=}sval_1\,;\ x_2{=}sval_2\}\ \texttt{->}\ \texttt{false}}\ \textsc{Eval\_Rhs\_Supeq\_Str\_Not}$$

$$\frac{bval_1 >= bval_2}{E \vdash x_1 >= x_2/\{x_1{=}bval_1\,;\ x_2{=}bval_2\}\ \texttt{->}\ \texttt{true}}\ \textsc{Eval\_Rhs\_Supeq\_Bytes}$$

$$\frac{bval_1 < bval_2}{E \vdash x_1 >= x_2/\{x_1{=}bval_1\,;\ x_2{=}bval_2\}\ \texttt{->}\ \texttt{false}}\ \textsc{Eval\_Rhs\_Supeq\_Bytes\_Not}$$

$$\frac{}{E \vdash [var_1\,;\ ..\,;\ var_n]/\{var_1{=}val_1\,;\ ..\,;\ var_n{=}val_n\}\ \texttt{->}\ ([val_1\,;\ ..\,;\ val_n]\ \texttt{:}\ \texttt{list}\ ty)}\ \textsc{Eval\_Rhs\_List}$$

$$\frac{}{E \vdash x_1\ \texttt{::}\ x_2/\{x_1{=}val_1\,;\ x_2{=}lval_2\}\ \texttt{->}\ val_1\ \texttt{::}\ lval_2}\ \textsc{Eval\_Rhs\_List\_Cons}$$

$$\frac{}{E \vdash \texttt{size}\,x/\{x{=}lval\}\ \texttt{->}\ |lval|}\ \textsc{Eval\_Rhs\_List\_Size}$$

$$\frac{\{var\texttt{=}(\texttt{[]}\ \texttt{:}\ \texttt{list}\ ty)\} \odot rval\texttt{=}rval' \qquad E \vdash rhs_1/rval\ \texttt{->}\ val}{E \vdash \texttt{match}\ var\ \texttt{with}\ \texttt{[]}\ \texttt{->}\ rhs_1\ \texttt{|}\ var_1\ \texttt{::}\ var_2\ \texttt{->}\ rhs_2\ \texttt{end}/rval'\ \texttt{->}\ val}\ \text{EVAL\_RHS\_LIST\_MATCH\_NIL}$$

$$\frac{\{var\texttt{=}val_1\ \texttt{::}\ tl\} \odot rval\texttt{=}rval' \qquad \{var_1\texttt{=}val_1\texttt{;}\ var_2\texttt{=}tl\} \odot rval\texttt{=}rval'' \qquad E \vdash rhs_2/rval''\ \texttt{->}\ val}{E \vdash \texttt{match}\ var\ \texttt{with}\ \texttt{[]}\ \texttt{->}\ rhs_1\ \texttt{|}\ var_1\ \texttt{::}\ var_2\ \texttt{->}\ rhs_2\ \texttt{end}/rval'\ \texttt{->}\ val}\ \text{EVAL\_RHS\_LIST\_MATCH\_CONS}$$

$$\frac{}{E \vdash \texttt{map}\ var\ \texttt{in}\ var'\ \texttt{do}\ rhs\ \texttt{done}/\{var'\texttt{=}(\texttt{[]}\ \texttt{:}\ \texttt{list}\ ty)\}\ \texttt{->}\ (\texttt{[]}\ \texttt{:}\ \texttt{list}\ ty)}\ \text{EVAL\_RHS\_LIST\_MAP\_NIL}$$

$$\frac{E \vdash rhs/\{var\texttt{=}val\}\ \texttt{->}\ val' \qquad E \vdash \texttt{map}\ var\ \texttt{in}\ var'\ \texttt{do}\ rhs\ \texttt{done}/\{var'\texttt{=}tl\}\ \texttt{->}\ tl'}{E \vdash \texttt{map}\ var\ \texttt{in}\ var'\ \texttt{do}\ rhs\ \texttt{done}/\{var'\texttt{=}val\ \texttt{::}\ tl\}\ \texttt{->}\ val'\ \texttt{::}\ tl'}\ \text{EVAL\_RHS\_LIST\_MAP\_CONS}$$

$$\frac{}{E \vdash \{|var_1\ \texttt{|->}\ var_1'\texttt{;}\ ..\texttt{;}\ var_n\ \texttt{|->}\ var_n'|\}/\{var_1\texttt{=}val_1\texttt{;}\ ..\texttt{;}\ var_n\texttt{=}val_n\texttt{;}\ var_1'\texttt{=}val_1'\texttt{;}\ ..\texttt{;}\ var_n'\texttt{=}val_n'\}\ \texttt{->}\ (\{|val_1\ |}$$

$$\frac{}{E \vdash \texttt{map}\ var_1\ \texttt{|->}\ var_2\ \texttt{in}\ var_3\ \texttt{do}\ rhs\ \texttt{done}/\{var_3\texttt{=}(\{||\}\ \texttt{:}\ \texttt{map}\ ty\ ty')\}\ \texttt{->}\ (\{||\}\ \texttt{:}\ \texttt{map}\ ty\ ty')}\ \text{EVAL\_RHS\_MA}$$

$$\frac{E \vdash rhs/\{var_1\texttt{=}val_1\texttt{;}\ var_2\texttt{=}val_2\}\ \texttt{->}\ val_2' \qquad E \vdash \texttt{map}\ var_1\ \texttt{|->}\ var_2\ \texttt{in}\ var_3\ \texttt{do}\ rhs\ \texttt{done}/\{var_3\texttt{=}mval\}\ \texttt{->}\ mval'}{E \vdash \texttt{map}\ var_1\ \texttt{|->}\ var_2\ \texttt{in}\ var_3\ \texttt{do}\ rhs\ \texttt{done}/\{var_3\texttt{=}\{|val_1\ \texttt{|->}\ val_2\texttt{;}\ <mval>|\}\}\ \texttt{->}\ \{|val_1\ \texttt{|->}\ val_2'\texttt{;}\ <mval'>}$$

$$\frac{}{E \vdash var_1\texttt{[}var_2\texttt{]}/\{var_1\texttt{=}(\{||\}\ \texttt{:}\ \texttt{map}\ ty\ ty')\texttt{;}\ var_2\texttt{=}val_2\}\ \texttt{->}\ (\texttt{None}\,\texttt{\{\}}\ \texttt{:}\ \texttt{[None : unit | Some :}\ ty\texttt{]})}\ \text{EVAL\_RH}$$

$$\frac{}{E \vdash var_1\texttt{[}var_2\texttt{]}/\{var_1\texttt{=}\{|val_2\ \texttt{|->}\ val\texttt{;}\ <mval>|\}\texttt{;}\ var_2\texttt{=}val_2\}\ \texttt{->}\ (\texttt{Some}\ val\ \texttt{:}\ \texttt{[None : unit | Some :}\ ty\texttt{])}$$

$$\frac{val_2 <> val_2' \qquad E \vdash var_1\texttt{[}var_2\texttt{]}/\{var_1\texttt{=}mval\texttt{;}\ var_2\texttt{=}val_2\}\ \texttt{->}\ val'}{E \vdash var_1\texttt{[}var_2\texttt{]}/\{var_1\texttt{=}\{|val_2\ \texttt{|->}\ val\texttt{;}\ <mval>|\}\texttt{;}\ var_2\texttt{=}val_2\}\ \texttt{->}\ val'}\ \text{EVAL\_RHS\_MAP\_GET\_NEXT}$$

$$\frac{}{E \vdash \{|var_1\ \texttt{with}\ var_2\ \texttt{|->}\ var_3|\}/\{var_1\texttt{=}(\{||\}\ \texttt{:}\ \texttt{map}\ ty\ ty')\texttt{;}\ var_2\texttt{=}val_2\texttt{;}\ var_3\texttt{=}(\texttt{None}\,\texttt{\{\}}\ \texttt{:}\ \texttt{[None : unit | Some}$$

$$\frac{}{E \vdash \{|var_1\ \texttt{with}\ var_2\ \texttt{|->}\ var_3|\}/\{var_1\texttt{=}(\{||\}\ \texttt{:}\ \texttt{map}\ ty\ ty')\texttt{;}\ var_2\texttt{=}val_2\texttt{;}\ var_3\texttt{=}(\texttt{Some}\ val_3\ \texttt{:}\ \texttt{[None : unit | Some}$$

$$\frac{}{E \vdash \{|var_1\ \texttt{with}\ var_2\ \texttt{|->}\ var_3|\}/\{var_1\texttt{=}\{|val_2\ \texttt{|->}\ val_3'\texttt{;}\ <mval>|\}\texttt{;}\ var_2\texttt{=}val_2\texttt{;}\ var_3\texttt{=}(\texttt{None}\,\texttt{\{\}}\ \texttt{:}\ \texttt{[None : un}$$

$$\frac{}{E \vdash \{|var_1\ \texttt{with}\ var_2\ \texttt{|->}\ var_3|\}/\{var_1\texttt{=}\{|val_2\ \texttt{|->}\ val_3'\texttt{;}\ <mval>|\}\texttt{;}\ var_2\texttt{=}val_2\texttt{;}\ var_3\texttt{=}(\texttt{Some}\ val_3\ \texttt{:}\ \texttt{[None : u}$$

$$\frac{value_2 <> value_2' \qquad E \vdash \{|var_1\ \texttt{with}\ var_2\ \texttt{|->}\ var_3|\}/\{var_1\texttt{=}mval\texttt{;}\ var_2\texttt{=}value_2\texttt{;}\ var_3\texttt{=}(\texttt{Some}\ val_3\ \texttt{:}\ \texttt{[None}}{E \vdash \{|var_1\ \texttt{with}\ var_2\ \texttt{|->}\ var_3|\}/\{var_1\texttt{=}\{|value_2'\ \texttt{|->}\ value_3'\texttt{;}\ <mval>|\}\texttt{;}\ var_2\texttt{=}val_2\texttt{;}\ var_3\texttt{=}(\texttt{Some}\ val_3\ \texttt{:}\ \texttt{[Non}}$$

$$\frac{}{E \vdash \texttt{size}\ x/\{x{=}mval\}\ \texttt{->}\ |mval|}\ \textsc{eval\_rhs\_map\_size}$$

$\boxed{E \vdash instruction/val\ \texttt{->}\ val'}$ Instruction evaluation

$$\frac{\begin{array}{l}E \vdash instruction/rval\ \texttt{->}\ rval' \\ rval \odot rval''{=}rval_1 \\ rval' \odot rval''{=}rval_2\end{array}}{E \vdash instruction/rval_1\ \texttt{->}\ rval_2}\ \textsc{eval\_instr\_frame}$$

$$\frac{}{E \vdash \texttt{noop}/\{\}\ \texttt{->}\ \{\}}\ \textsc{eval\_instr\_noop}$$

$$\frac{\begin{array}{l}E \vdash I_1/val\ \texttt{->}\ val' \\ E \vdash I_2/val'\ \texttt{->}\ val''\end{array}}{E \vdash I_1\texttt{;}\ I_2/val\ \texttt{->}\ val''}\ \textsc{eval\_instr\_seq}$$

$$\frac{\begin{array}{l}E \vdash rhs/val\ \texttt{->}\ val' \\ E \vdash lhs/val'\ \texttt{->}\ val''\end{array}}{E \vdash lhs{=}rhs/val\ \texttt{->}\ val''}\ \textsc{eval\_instr\_assign}$$

$$\frac{}{E \vdash \texttt{drop}\ var/\{var{=}val\}\ \texttt{->}\ \{\}}\ \textsc{eval\_instr\_drop}$$

$$\frac{\begin{array}{l}\{var{=}(constructor\ val'\ :\ vty)\} \odot rval{=}rval' \\ \{var'{=}val'\} \odot rval{=}rval' \\ E \vdash I/rval'\ \texttt{->}\ rval_1\end{array}}{E \vdash \texttt{match}\ var\ \texttt{with}\ constructor\ var'\ \texttt{->}\ I\ \texttt{|}\ <branches\_ins>\ \texttt{end}/rval'\ \texttt{->}\ rval_1}\ \textsc{eval\_instr\_match\_f}$$

$$\frac{\begin{array}{l}\{var{=}(constructor\ val'\ :\ vty)\} \odot rval{=}rval' \\ constructor <> constructor' \\ E \vdash \texttt{match}\ var\ \texttt{with}\ branches\_ins\ \texttt{end}/rval'\ \texttt{->}\ rval_1\end{array}}{E \vdash \texttt{match}\ var\ \texttt{with}\ constructor'\ var'\ \texttt{->}\ I\ \texttt{|}\ <branches\_ins>\ \texttt{end}/rval'\ \texttt{->}\ rval_1}\ \textsc{eval\_instr\_match\_}$$

$$\frac{\{var'{=}(\texttt{Right}\ val\ :\ vty)\} \odot rval{=}rval'}{E \vdash \texttt{loop\_left}\ var\ \texttt{in}\ var'\ \texttt{do}\ I\ \texttt{done}/rval'\ \texttt{->}\ rval}\ \textsc{eval\_instr\_loop\_left\_end}$$

$$\frac{\begin{array}{l}\{var'{=}(\texttt{Left}\ val\ :\ vty)\} \odot rval{=}rval' \\ \{var{=}val\} \odot rval{=}rval'' \\ E \vdash I/rval''\ \texttt{->}\ rval_1 \\ E \vdash \texttt{loop\_left}\ var\ \texttt{in}\ var'\ \texttt{do}\ I\ \texttt{done}/rval_1\ \texttt{->}\ rval_2\end{array}}{E \vdash \texttt{loop\_left}\ var\ \texttt{in}\ var'\ \texttt{do}\ I\ \texttt{done}/rval'\ \texttt{->}\ rval_2}\ \textsc{eval\_instr\_loop\_left\_continue}$$

$$\frac{\{var'{=}(\texttt{False}\,\{\}\ :\ vty)\} \odot rval{=}rval'}{E \vdash \texttt{loop}\ var\ \texttt{do}\ I\ \texttt{done}/rval'\ \texttt{->}\ rval}\ \textsc{eval\_instr\_loop\_end}$$

$$\frac{\{var'\texttt{=(True \{ \} : } vty\texttt{)}\} \odot rval\texttt{=}rval' \quad E \vdash I/rval \texttt{ -> } rval_1 \quad E \vdash \texttt{loop } var \texttt{ do } I \texttt{ done}/rval_1 \texttt{ -> } rval_2}{E \vdash \texttt{loop } var \texttt{ do } I \texttt{ done}/rval' \texttt{ -> } rval_2} \text{ EVAL\_INSTR\_LOOP\_CONTINUE}$$

$$\frac{\{var\texttt{=([] : list } ty\texttt{)}\} \odot rval\texttt{=}rval' \quad E \vdash instruction_1/rval \texttt{ -> } val}{E \vdash \texttt{match } var \texttt{ with [] -> } instruction_1 \texttt{ | } var_1 \texttt{ :: } var_2 \texttt{ -> } I_2 \texttt{ end}/rval' \texttt{ -> } val} \text{ EVAL\_INSTR\_LIST\_MATCH\_N}$$

$$\frac{\{var_1\texttt{=}val_1 \texttt{ :: } tl\texttt{}\} \odot rval\texttt{=}rval' \quad \{var_1\texttt{=}val_1\texttt{; } var_2\texttt{=}tl\texttt{}\} \odot rval\texttt{=}rval'' \quad E \vdash instruction_2/rval'' \texttt{ -> } val}{E \vdash \texttt{match } var \texttt{ with [] -> } instruction_1 \texttt{ | } var_1 \texttt{ :: } var_2 \texttt{ -> } instruction_2 \texttt{ end}/rval' \texttt{ -> } val} \text{ EVAL\_INSTR\_LIST\_}$$

$$\frac{\{var'\texttt{=([] : list } ty\texttt{)}\} \odot rval\texttt{=}rval'}{E \vdash \texttt{for } var \texttt{ in } var' \texttt{ do } instruction_1 \texttt{ done}/rval' \texttt{ -> } rval} \text{ EVAL\_INSTR\_LIST\_FOR\_NIL}$$

$$\frac{\{var\texttt{=}val\texttt{}\} \odot rval\texttt{=}rval' \quad E \vdash instruction_1/rval' \texttt{ -> } rval_1 \quad \{var'\texttt{=}tl\texttt{}\} \odot rval_1\texttt{=}rval'' \quad E \vdash \texttt{for } var \texttt{ in } var' \texttt{ do } instruction_1 \texttt{ done}/rval'' \texttt{ -> } rval_2}{E \vdash \texttt{for } var \texttt{ in } var' \texttt{ do } instruction_1 \texttt{ done}/\{var'\texttt{=}val \texttt{ :: } tl\texttt{}\} \texttt{ -> } rval_2} \text{ EVAL\_INSTR\_LIST\_FOR\_CONS}$$

$$\frac{\{var''\texttt{=}(\{|\,|\}\texttt{ : map } ty\ ty'\texttt{)}\} \odot rval\texttt{=}rval'}{E \vdash \texttt{for } var \texttt{ |-> } var' \texttt{ in } var'' \texttt{ do } instruction_1 \texttt{ done}/rval' \texttt{ -> } rval} \text{ EVAL\_INSTR\_MAP\_FOR\_NIL}$$

$$\frac{\{var\texttt{=}val\texttt{; } var'\texttt{=}val'\texttt{}\} \odot rval\texttt{=}rval' \quad E \vdash instruction_1/rval' \texttt{ -> } rval_1 \quad \{var''\texttt{=}mval\texttt{}\} \odot rval_1\texttt{=}rval'' \quad E \vdash \texttt{for } var \texttt{ |-> } var' \texttt{ in } var'' \texttt{ do } instruction_1 \texttt{ done}/rval'' \texttt{ -> } rval_2}{E \vdash \texttt{for } var \texttt{ |-> } var' \texttt{ in } var'' \texttt{ do } instruction_1 \texttt{ done}/\{var''\texttt{=}\{|val \texttt{ |-> } val'\texttt{; } < mval > |\}\texttt{}\} \texttt{ -> } rval_2} \text{ EVAL\_INST}$$

$\boxed{E \vdash f/val \texttt{ -> } val'}$ Function symbol evaluation

$$\frac{fvar\texttt{=}instruction \in E \quad E \vdash instruction/val \texttt{ -> } val'}{E \vdash fvar/val \texttt{ -> } val'} \text{ EVAL\_F\_FVAR}$$

$$\frac{}{E \vdash \texttt{dup}/val \texttt{ -> \{car=}val\texttt{; cdr=}val\texttt{\}}} \text{ EVAL\_F\_DUP}$$

$$\frac{}{E \vdash \texttt{(}constructor \texttt{ : } vty\texttt{)}/val \texttt{ -> (}constructor\ val \texttt{ : } vty\texttt{)}} \text{ EVAL\_F\_CONSTRUCTOR}$$

$$\frac{}{E \vdash \texttt{contract } ty/add\_val \texttt{ -> get\_contract } add\_val} \text{ EVAL\_F\_CONTRACT}$$

$$\frac{}{E \vdash \texttt{address}/add\_val \texttt{ -> get\_address } add\_val} \text{ EVAL\_F\_ADDRESS}$$

$$\frac{}{E \vdash \texttt{implicit\_account}/add\_val \texttt{ -> get\_implicit\_account } add\_val} \text{ EVAL\_F\_IMPLICIT\_ACCOUNT}$$

$$\frac{E \vdash instruction/val_1 \texttt{ -> } val_2}{E \vdash \texttt{exec}/\{\texttt{argument=}val_2\texttt{; function=lambda } ty \texttt{ : } ty'.instruction \texttt{ end}\} \texttt{ -> } val_2} \text{ EVAL\_F\_EXEC}$$