

Quantities
oooooooo

Syntax
ooooooo

Contexts
oooooooo

Programming
oooooooooooo

Logic
ooooooo

Totality
oo

T-Axi (declarative)

Wojciech Kołowski
Mateusz Pyzik

TODO

- Kinding.
- Polymorphism.
- Higher-order quantification.
- Totality checker.
- Computation and conversion checking.
- Inductives and records.
- Algorithmic system.
- Distinguish noncomputable terms.
- Distinguish partial terms.

Quantities

Quantities:

$r ::= 0 \mid 1 \mid ? \mid + \mid *$

- 0 means a resource has been used up and is no longer available.
- 1 means a resource must be used exactly once.
- ? (pronounced “few”) means a resource must be used at most once.
- + (pronounced “many”) means a resource must be used at least once.
- * (pronounced “any”) means no restrictions on usage.

* is the default quantity, so when there's nothing to indicate quantity, it means it's *.

Subusage ordering

$r_1 \leq r_2$ means that a resource with quantity r_1 may be used when quantity r_2 is expected. This ordering is called sub-usaging. The definition below is just the skeleton, the full ordering is the reflexive-transitive closure of it.

$$* \leq ?$$

$$* \leq +$$

$$+ \leq 1$$

$$? \leq 1$$

$$? \leq 0$$

We will denote the greatest lower bound in this order with $r_1 \sqcap r_2$ and the least upper bound (if it exists) with $r_1 \sqcup r_2$.

Addition of quantities

When we have two quantities of the same resource, we can sum the quantities.

$$0 + r = r$$

$$r + 0 = r$$

$$? + ? = *$$

$$? + * = *$$

$$* + ? = *$$

$$* + * = *$$

$$_ + _ = +$$

Multiplication of quantities

When we have a quantity r_1 of resource A that contains quantity r_2 of resource B , then we in fact have quantity $r_1 \cdot r_2$ of resource B .

$$0 \cdot r = 0$$

$$r \cdot 0 = 0$$

$$1 \cdot r = r$$

$$r \cdot 1 = r$$

$$?\cdot?=?$$

$$+\cdot+=+$$

$$-\cdot-=*$$

The algebra of quantities

Quantities \mathcal{Q} form a positive ordered commutative semiring with no zero divisors, i.e.:

- $(\mathcal{Q}, +, 0)$ is a commutative monoid.
- $(\mathcal{Q}, \cdot, 1)$ is a commutative monoid.
- 0 annihilates multiplication.
- Multiplication distributes over addition.
- Addition and multiplication preserve the subusage ordering in both arguments.
- If $r_1 + r_2 = 0$, then $r_1 = 0$ and $r_2 = 0$.
- If $r_1 \cdot r_2 = 0$, then $r_1 = 0$ or $r_2 = 0$.

Subtraction of quantities

$r_1 - r_2$ is the least r' such that $r_1 \leq r' + r_2$.

$r_1 - r_2$	0	1	?	+	*
0	0				
1	1	0			
?	?	0	0		
+	+	*	+	*	+
*	*	*	*	*	*

Division with remainder

$a/b = (q, r)$ when $a = b \cdot q + r$, with q as large as possible and r being as small as possible according to the order
 $0 \leq 1 \leq ? \leq + \leq *$. Note that $a/b = q$ means that $r = 0$.

r_1/r_2	0	1	?	+	*
0	*	0	0	0	0
1	(*, 1)	1	(0, 1)	(0, 1)	(0, 1)
?	(*, ?)	?	?	(0, ?)	(0, ?)
+	(*, +)	+	(*, 1)	+	(*, 1)
*	(*, *)	*	*	*	*

Quantities
oooooooo

Syntax
●oooooooo

Contexts
oooooooooooo

Programming
ooooooooooooo

Logic
oooooooo

Totality
oo

Types

Types:

$A, B ::= r A \rightarrow B \mid !_r A \mid A \otimes B \mid A \oplus B \mid \text{Unit} \mid \text{Empty}$

Terms

Terms:

$e ::=$

$$\begin{aligned} & x \mid \lambda_r x : A. e \mid e_1 \ e_2 \mid \\ & \text{box}_r \ e \mid \text{letbox } x : A = e_1 \ \text{in } e_2 \\ & (e_1, e_2) \mid \text{letpair } (x, y) = e_1 \ \text{in } e_2 \mid \\ & \text{inl}_A \ e \mid \text{inr}_A \ e \mid \text{case } e \text{ of } (e_1, e_2) \mid \\ & \text{unit} \mid \text{letunit}_A \ \text{unit} = e_1 \ \text{in } e_2 \mid \\ & \text{Empty-elim}_A \ e \mid \\ & \text{let}_r \ x : A = e_1 \ \text{in } e_2 \mid \\ & \text{let noncomputable } x : A = e_1 \ \text{in } e_2 \mid \\ & \text{choose}_A \ p \mid \text{choose-witness } x \ h \text{ for } p \text{ in } e \end{aligned}$$

$\text{choose}_A \ p$ and $\text{choose-witness } x \ h \text{ for } p \text{ in } e$ are noncomputable terms, whereas all others are computable.

Propositions

Propositions:

$P, Q ::=$

$\top \mid \perp \mid P \Rightarrow Q \mid P \wedge Q \mid P \vee Q \mid$

$\forall x : A. P \mid \exists x : A. P \mid$

$e_1 =_A e_2$

Notations:

$\neg P$ stands for $P \Rightarrow \perp$

$P \Leftrightarrow Q$ stands for $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$

Proofterms

Proofterms (P, Q are propositions, e are terms, h are variables):

$p, q ::=$

$h \mid \text{assumption} \mid \text{trivial} \mid \text{absurd } p$

$\text{assume } h : P \text{ in } q \mid \text{apply } p_1 \ p_2 \mid$

$\text{both } p_1 \ p_2 \mid \text{and-left } p \mid \text{and-right } p \mid$

$\text{or-left } p \mid \text{or-right } p \mid \text{cases } p_1 \ p_2 \ p_3 \mid$

$\text{lemma } h : P \text{ by } p \text{ in } q \mid \text{proving } P \text{ by } p \mid$

$\text{suffices } P \text{ by } q \text{ in } p \mid$

$\text{pick-any } x : A \text{ in } e \mid \text{instantiate } p \text{ with } e \mid$

$\text{witness } e \text{ such that } p \mid \text{pick-witness } x \ h \text{ for } p_1 \text{ in } p_2 \mid$

$\text{refl } e \mid \text{rewrite } p_1 \text{ in } p_2 \mid \text{funext } x : A \text{ in } p$

$\text{by-contradiction } h : \neg P \text{ in } q \mid$

$\text{choose-spec } p \mid \text{choose-witness } x \ h \text{ for } p \text{ in } q$

Quantities
○○○○○○○

Syntax
○○○●○○

Contexts
○○○○○○○○

Programming
○○○○○○○○○○

Logic
○○○○○○○

Totality
○○

Contexts

Contexts:

$$\Gamma ::= \cdot \mid \Gamma, rx : A \mid \Gamma, rx : A := e \mid \Gamma, h : P$$

Judgements

Well-formed context judgement: $\Gamma \text{ ctx}$

Cartesian context judgement: $\Gamma \text{ cartesian}$

Well-formed type judgement: $\Gamma \vdash A \text{ type}$

Typing judgement: $\Gamma \vdash; e : A$

Well-formed proposition judgement: $\Gamma \vdash P \text{ prop}$

Proof judgement: $\Gamma \vdash p : P$

Sanity checks

We'll set up the system so that:

- If $\Gamma \vdash_i e : A$, then $|\Gamma| \vdash A$ type.
- If $\Gamma \vdash p : P$, then $\Gamma \vdash P$ prop.
- If $\Gamma \vdash P$ prop, then Γ cartesian.
- If $\Gamma \vdash A$ type, then Γ cartesian.
- If Γ cartesian, then Γ ctx.

Operations on contexts

Operations on contexts:

$\Gamma_1 \leq \Gamma_2$ – context subusaging

$\Gamma_1 + \Gamma_2$ – context addition

$r\Gamma$ – context scaling

Γ/r – context division (with remainder)

$|\Gamma|$ – cartesianization

$|\Gamma|_x$ – spotlight x

Context subusaging

$$\overline{\cdot \leq \cdot}$$

$$\frac{\Gamma_1 \leq \Gamma_2 \quad r_1 \leq r_2}{\Gamma_1, r_1 x : A \leq \Gamma_2, r_2 x : A}$$

$$\frac{\Gamma_1 \leq \Gamma_2 \quad r_1 \leq r_2}{\Gamma_1, r_1 x : A := e \leq \Gamma_2, r_2 x : A := e}$$

$$\frac{\Gamma_1 \leq \Gamma_2}{\Gamma_1, h : P \leq \Gamma_2, h : P}$$

Context addition

$$\cdot + \cdot = \cdot$$

$$(\Gamma_1, r_1 x : A) + (\Gamma_2, r_2 x : A) = (\Gamma_1 + \Gamma_2), (r_1 + r_2) x : A$$

$$(\Gamma_1, r_1 x : A := e) + (\Gamma_2, r_2 x : A := e) = (\Gamma_1 + \Gamma_2), (r_1 + r_2) x : A := e$$

$$(\Gamma_1, p : P) + (\Gamma_2, h : P) = (\Gamma_1 + \Gamma_2), h : P$$

Context scaling

$$\begin{aligned}s \cdot &= \cdot \\ s(\Gamma, rx : A) &= s\Gamma, (s \cdot r)x : A \\ s(\Gamma, rx : A := e) &= s\Gamma, (s \cdot r)x : A := e \\ s(\Gamma, h : P) &= s\Gamma, h : P\end{aligned}$$

Spotlight

$$|\cdot|_x = \cdot$$

$$|\Gamma, rx : A|_x = 0 \Gamma, 1x : A$$

$$|\Gamma, ry : A|_x = |\Gamma|_x, 0y : A$$

$$|\Gamma, rx : A := e|_x = 0 \Gamma, 1x : A := e$$

$$|\Gamma, ry : A := e|_x = |\Gamma|_x, 0y : A := e$$

$$|\Gamma, h : P|_x = |\Gamma|_x, h : P$$

Context division with remainder

$$\overline{\cdot / r} = \cdot$$

$$\frac{\Gamma/q = (\Gamma_1, \Gamma_2) \quad r/q = (r_1, r_2)}{(\Gamma, r x : A)/q = ((\Gamma_1, r_1 x : A), (\Gamma_2, r_2 x : A))}$$

$$\frac{\Gamma/q = (\Gamma_1, \Gamma_2) \quad r/q = (r_1, r_2)}{(\Gamma, r x : A := e)/q = ((\Gamma_1, r_1 x : A := e), (\Gamma_2, r_2 x : A := e))}$$

$$\frac{\Gamma/q = \Gamma'}{(\Gamma, h : P)/q = ((\Gamma', h : P), (\Gamma', h : P))}$$

Cartesianization

Cartesianization turns a context into a traditional context that doesn't mention any quantities.

$$\begin{aligned} |\cdot| &= \cdot \\ |\Gamma, r x : A| &= |\Gamma|, x : A \\ |\Gamma, r x : A := e| &= |\Gamma|, x : A := e \\ |\Gamma, h : P| &= |\Gamma|, h : P \end{aligned}$$

Well-formed contexts

 $\frac{}{\cdot \text{ ctx}}$

$$\frac{\Gamma \text{ ctx} \quad |\Gamma| \vdash A \text{ type} \quad x \notin \Gamma}{\Gamma, r x : A \text{ ctx}}$$

$$\frac{\Gamma_1 \text{ ctx} \quad \Gamma_1 \leq \Gamma + r \Gamma_2 \quad \Gamma_2 \vdash_i e : A \quad x \notin \Gamma \quad i \text{ inferred from } e}{\Gamma, r x : A := e \text{ ctx}}$$

$$\frac{\Gamma \text{ ctx} \quad |\Gamma| \vdash P \text{ prop} \quad h \notin \Gamma}{\Gamma, h : P \text{ ctx}}$$

Q: Do we really need to check type well-formedness in a cartesianized context?

Cartesian contexts

$$\frac{\Gamma \text{ ctx} \quad \Gamma = |\Gamma|}{\Gamma \text{ cartesian}}$$

Well-formed types

$$\frac{\Gamma \text{ cartesian}}{\Gamma \vdash \text{Unit type}}$$

$$\frac{\Gamma \text{ cartesian}}{\Gamma \vdash \text{Empty type}}$$

$$\frac{\Gamma \vdash A \text{ type} \quad \Gamma \vdash B \text{ type}}{\Gamma \vdash rA \rightarrow B \text{ type}}$$

$$\frac{\Gamma \vdash A \text{ type}}{\Gamma \vdash !_r A \text{ type}}$$

$$\frac{\Gamma \vdash A \text{ type} \quad \Gamma \vdash B \text{ type}}{\Gamma \vdash A \otimes B \text{ type}}$$

$$\frac{\Gamma \vdash A \text{ type} \quad \Gamma \vdash B \text{ type}}{\Gamma \vdash A \oplus B \text{ type}}$$

The inherent quantity of a type

$$\begin{aligned} \text{qty}(\text{Unit}) &= * \\ \text{qty}(\text{Empty}) &= * \\ \text{qty}(!_0 A) &= * \\ \text{qty}(!_r A) &= r \cdot \text{qty}(A) \\ \text{qty}(A \otimes B) &= \text{qty}(A) \sqcup \text{qty}(B) \\ \text{qty}(A \oplus B) &= \text{qty}(A) \sqcup \text{qty}(B) \\ \text{qty}(r A \rightarrow B) &= 1 \end{aligned}$$

Quantities
oooooooo

Syntax
ooooooo

Contexts
oooooooo

Programming
oo●oooooooo

Logic
ooooooo

Totality
oo

Total quantity added to context

$$\begin{aligned}r_A^{\text{nc}} &= * \\r_A^{\text{c}} &= r \cdot \text{qty}(A)\end{aligned}$$

Using variables

$$\frac{\Gamma \text{ ctx } (x : A) \in \Gamma \quad \Gamma \leq |\Gamma|_x}{\Gamma \vdash_i x : A}$$

Functions

$$\frac{\Gamma, r_A^i x : A \vdash_i e : B}{\Gamma \vdash_i \lambda_r x : A. e : r A \rightarrow B}$$

$$\frac{\Gamma \leq \Gamma_1 + r \Gamma_2 \quad \Gamma_1 \vdash_i e_1 : r A \rightarrow B \quad \Gamma_2 \vdash_i e_2 : A}{\Gamma \vdash_i e_1 \ e_2 : B}$$

Box

$$\frac{\Gamma \leq r \Gamma' \quad \Gamma' \vdash_i e : A}{\Gamma \vdash; \text{box}_r e : !_r A}$$

$$\frac{\Gamma \leq \Gamma_1 + \Gamma_2 \quad \Gamma_1 \vdash_i e_1 : !_r A \quad \Gamma_2, r_A^i x : A \vdash_i e_2 : B}{\Gamma \vdash; \text{letbox } x : A = e_1 \text{ in } e_2 : B}$$

Q: Notation for box intro and elim?

Empty

$$\frac{|\Gamma| \vdash A \text{ type} \quad \Gamma \vdash; e : \text{Empty}}{\Gamma \vdash; \text{Empty-elim}_A e : A}$$

Unit

$$\frac{\Gamma \text{ ctx} \quad \Gamma \leq 0 \Gamma}{\Gamma \vdash; \text{unit} : \text{Unit}}$$

$$\frac{\Gamma \leq \Gamma_1 + \Gamma_2 \quad \Gamma_1 \vdash; e_1 : \text{Unit} \quad \Gamma_2 \vdash; e_2 : A}{\Gamma \vdash; \text{letunit}_A \text{ unit} = e_1 \text{ in } e_2 : A}$$

Products

$$\frac{\Gamma \leq \Gamma_1 + \Gamma_2 \quad \Gamma_1 \vdash_i a : A \quad \Gamma_2 \vdash_i b : B}{\Gamma \vdash_i (a, b) : A \otimes B}$$

$$\frac{\Gamma \leq \Gamma_1 + \Gamma_2 \quad \Gamma_1 \vdash_i e_1 : A \otimes B \quad \Gamma_2, 1_A^i x : A, 1_B^i y : B \vdash_i e_2 : C}{\Gamma \vdash_i \text{letpair } (x, y) = e_1 \text{ in } e_2 : C}$$

Sums

$$\frac{\Gamma \vdash_i e : A \quad |\Gamma| \vdash B \text{ type}}{\Gamma \vdash_i \text{inl}_B e : A \oplus B} \quad \frac{|\Gamma| \vdash A \text{ type} \quad \Gamma \vdash_i e : B}{\Gamma \vdash_i \text{inr}_A e : A \oplus B}$$

$$\frac{\Gamma \leq \Gamma_1 + \Gamma_2 \quad \Gamma_1 \vdash_i e : A \oplus B \quad \Gamma_2 \vdash_i f : 1A \rightarrow C \quad \Gamma_2 \vdash_i g : 1B \rightarrow C}{\Gamma \vdash_i \text{case } e \text{ of } (f, g) : C}$$

Q: Do we want first-order representation of the branches?
Probably yes.

Let

$$\frac{\Gamma \leq r \Gamma_1 + \Gamma_2 \quad \Gamma_1 \vdash_i e_1 : A \quad \Gamma_2, r_A^i x : A := e_1 \vdash_i e_2 : B}{\Gamma \vdash_i \text{let}_r x : A = e_1 \text{ in } e_2 : B}$$

Well-formed propositions

$$\frac{}{\Gamma \vdash \top \text{ prop}}$$

$$\frac{}{\Gamma \vdash \perp \text{ prop}}$$

$$\frac{\Gamma \vdash P \text{ prop} \quad \Gamma \vdash Q \text{ prop}}{\Gamma \vdash P \Rightarrow Q \text{ prop}}$$

$$\frac{\Gamma \vdash P \text{ prop} \quad \Gamma \vdash Q \text{ prop}}{\Gamma \vdash P \wedge Q \text{ prop}}$$

$$\frac{\Gamma \vdash P \text{ prop} \quad \Gamma \vdash Q \text{ prop}}{\Gamma \vdash P \vee Q \text{ prop}}$$

$$\frac{\Gamma \vdash A \text{ type} \quad \Gamma, x : A \vdash P \text{ prop}}{\Gamma \vdash \forall x : A. P \text{ prop}}$$

$$\frac{\Gamma \vdash A \text{ type} \quad \Gamma, x : A \vdash P \text{ prop}}{\Gamma \vdash \exists x : A. P \text{ prop}}$$

$$\frac{\Gamma \vdash A \text{ type} \quad \Gamma \vdash_{\text{nc}} e_1 : A \quad \Gamma \vdash_{\text{nc}} e_2 : A}{\Gamma \vdash e_1 =_A e_2 \text{ prop}}$$

Substitution

The notation is $P[x := e]$ for substitution in propositions.

Q: Do we need to check type well-formedness when checking proposition well-formedness? It should follow from the other sanity checks.

Type well-formedness checks are redundant.

Assumptions and implication

$$\frac{\Gamma \text{ cartesian } (h : P) \in \Gamma}{\Gamma \vdash h : P} \quad \frac{\Gamma \text{ cartesian } (h : P) \in \Gamma}{\Gamma \vdash \mathbf{assumption} : P}$$

$$\frac{\Gamma, h : P \vdash q : Q}{\Gamma \vdash \mathbf{assume} \ h : P \ \mathbf{in} \ q : P \Rightarrow Q}$$

$$\frac{\Gamma \vdash q : P \Rightarrow Q \quad \Gamma \vdash p : P}{\Gamma \vdash \mathbf{apply} \ q \ p : Q}$$

Propositional logic

$$\frac{\Gamma \text{ cartesian}}{\Gamma \vdash \mathbf{trivial} : \top}$$

$$\frac{\Gamma \vdash Q \text{ prop} \quad \Gamma \vdash p : \perp}{\Gamma \vdash \mathbf{absurd} \ p : Q}$$

$$\frac{\Gamma \vdash p : P \quad \Gamma \vdash q : Q}{\Gamma \vdash \mathbf{both} \ p \ q : P \wedge Q}$$

$$\frac{\Gamma \vdash pq : P \wedge Q}{\Gamma \vdash \mathbf{and-left} \ pq : P}$$

$$\frac{\Gamma \vdash pq : P \wedge Q}{\Gamma \vdash \mathbf{and-right} \ pq : Q}$$

$$\frac{\Gamma \vdash p : P \quad \Gamma \vdash Q \text{ prop}}{\Gamma \vdash \mathbf{or-left} \ p : P \vee Q}$$

$$\frac{\Gamma \vdash P \text{ prop} \quad \Gamma \vdash q : Q}{\Gamma \vdash \mathbf{or-right} \ q : P \vee Q}$$

$$\frac{\Gamma \vdash pq : P \vee Q \quad \Gamma \vdash r_1 : P \Rightarrow R \quad \Gamma \vdash r_2 : Q \Rightarrow R}{\Gamma \vdash \mathbf{cases} \ pq \ r_1 \ r_2 : R}$$

Utilities

$$\frac{\Gamma \vdash p : P \quad \Gamma, h : P \vdash q : Q}{\Gamma \vdash \mathbf{lemma} \ h : P \ \mathbf{by} \ p \ \mathbf{in} \ q : Q}$$

$$\frac{\Gamma \vdash p : P}{\Gamma \vdash \mathbf{proving} \ P \ \mathbf{by} \ p : P}$$

$$\frac{\Gamma \vdash pq : P \Rightarrow Q \quad \Gamma \vdash p : P}{\Gamma \vdash \mathbf{suffices} \ P \ \mathbf{by} \ pq \ \mathbf{in} \ p : Q}$$

Quantifiers

$$\frac{\Gamma, x : A \vdash p : P}{\Gamma \vdash \mathbf{pick-any} \ x : A \ \mathbf{in} \ p : \forall x : A. P}$$

$$\frac{\Gamma \vdash p : \forall x : A. P \quad \Gamma \vdash_{\text{nc}} e : A}{\Gamma \vdash \mathbf{stantiate} \ p \ \mathbf{with} \ e : P[x := e]}$$

$$\frac{\Gamma \vdash_{\text{nc}} e : A \quad \Gamma, x : A \vdash P \text{ prop} \quad \Gamma \vdash p : P[x := e]}{\Gamma \vdash \mathbf{witness} \ e \ \mathbf{such \ that} \ p : \exists x : A. P}$$

$$\frac{\Gamma \vdash p : \exists x : A. P \quad \Gamma \vdash Q \text{ prop} \quad \Gamma, x : A, h : P \vdash q : Q}{\Gamma \vdash \mathbf{pick-witness} \ x \ h \ \mathbf{for} \ p \ \mathbf{in} \ q : Q}$$

Equality

$$\frac{\Gamma \vdash_{\text{nc}} e : A}{\Gamma \vdash \mathbf{refl} e : e =_A e}$$

$$\frac{\Gamma \vdash q : e_1 =_A e_2 \quad \Gamma, x : A \vdash P \text{ prop} \quad \Gamma \vdash p : P[x := e_2]}{\Gamma \vdash \mathbf{rewrite} \ q \ \mathbf{in} \ p : P[x := e_1]}$$

$$\frac{\Gamma, x : A \vdash p : f \ x =_B g \ x}{\Gamma \vdash \mathbf{funext} \ x : A \ \mathbf{in} \ p : f =_{rA \rightarrow B} g}$$

Classical Logic

$$\frac{\Gamma, h : \neg P \vdash q : \perp}{\Gamma \vdash \text{by-contradiction } h : \neg P \text{ in } q : P}$$

$$\frac{|\Gamma| \vdash p : \exists x : A. P}{\Gamma \vdash_{\text{nc}} \text{choose } p : A}$$

$$\frac{\Gamma \vdash p : \exists x : A. P}{\Gamma \vdash \text{choose-spec } p : P[x := \text{choose } p]}$$

$$\frac{\Gamma \vdash p : \exists x : A. P \quad \Gamma \vdash Q \text{ prop} \quad \Gamma, x : A := \text{choose } p, h : P \vdash q : Q}{\Gamma \vdash \text{choose-witness } x \text{ for } p \text{ in } q : Q}$$

$$\frac{|\Gamma| \vdash p : \exists x : A. P \quad |\Gamma| \vdash B \text{ type} \quad \Gamma, x : A := \text{choose } p, h : P \vdash_{\text{nc}} e : B}{\Gamma \vdash_{\text{nc}} \text{choose-witness } x \text{ for } p \text{ in } e : B}$$

Environments

Global environments:

$\Sigma ::=$

$\emptyset \mid \Sigma, h : P := p \mid \Sigma, x : A := e \mid$

$\Sigma, \text{partial } x : A := e \mid \Sigma, \text{totality } x p$

Well-formed environments

 $\overline{\emptyset \text{ env}}$

$$\frac{\Sigma \text{ env} \quad h \notin \Sigma \quad \Sigma | \cdot \vdash p : P}{\Sigma, h : P := p \text{ env}}$$

$$\frac{\Sigma \text{ env} \quad x \notin \Sigma \quad \Sigma | \cdot \vdash_c e : A}{\Sigma, x : A := e \text{ env}}$$

$$\frac{\Sigma \text{ env} \quad x \notin \Sigma \quad \Sigma | \cdot \vdash_c e : A \quad e \text{ fails syntactic check}}{\Sigma, \text{partial } x : A := e \text{ env}}$$

$$\frac{\Sigma \text{ env} \quad \Sigma = \Sigma_1, \text{partial } x : A := e, \Sigma_2 \quad \Sigma | \cdot \vdash p : \exists r : A. e = ? = r}{\Sigma, \text{totality } x : p \text{ env}}$$