Modeling Reachability Types with Logical Relations: Semantic Type Soundness, Termination, Effect Safety, and Equational Theory (Artifact Document)

YUYAN BAO, Augusta University, USA
SONGLIN JIA, Purdue University, USA
GUANNAN WEI*, Tufts University, USA
OLIVER BRAČEVAC†, EPFL, Switzerland
TIARK ROMPF, Purdue University, USA

Contents

*Work completed while at Purdue University.
†Work completed while at Purdue University.

This is the artifact document for the paper Bao et al. [2025a] and its extended version [Bao et al. 2025b]. The Rocq developments are available online at https://github.com/tiarkrompf/reachability.

Authors' addresses: Yuyan Bao, Augusta University, USA, yubao@augusta.edu; Songlin Jia, Purdue University, USA, jia137@purdue.edu; Guannan Wei, Tufts University, USA, guannan.wei@tufts.edu; Oliver Bračevac, EPFL, Switzerland, oliver.bracevac@epfl.ch; Tiark Rompf, Purdue University, USA, tiark@purdue.edu.

A ROCQ MECHANIZATION FOR SECTION 2 IN PAPER (WITHOUT HIGHLIGHTED)

We outline the correspondence between the formalism in Section 2 (without highlighted) of the paper and its implementation in Rocq.

A.1 Model (sec2_stlc.v)

`ty`	$S, T, U, V :=$	`Type`
`TBool`	$Bool$	`Boolean Type`
`TRef T`	Ref T	`Reference Type`
`TFun T U`	$T \to U$	`Function Type`
`tm`	$t :=$	`Term`
`ttrue`	true	`Term true`
`tfalse`	false	`Term false`
`tvar x`	x	`Variable`
`tref t`	ref t	`Store Allocation`
`tget t`	! t	`Store Read`
`tput t₁ t₂` → `tput` t_1 t_2	$t_1 := t_2$	`Store Write`
`tapp` t_1 t_2	$t_1\ t_2$	`Functaion Application`
`tabs t`	$\lambda x.\ t$	`Abstraction`
`tseq` t_1 t_2	$t_1; t_2$	`Sequence Term`

A.2 Big-Step Operational Semantics (Extended Version of the Paper (Figure 1) [Bao et al. 2025b])

`venv`	H	`Value environment`
`vl`	$v :=$	`Value`
`vbool c`	c	`Boolean constant value`
`vref` ℓ	ℓ	`Store location`
`vabs H t`	$\langle H,\ \lambda\ x\ t\rangle$	`Closure record`
`teval(n : nat)(M : stor)(env : venv)(t : tm) : nat * stor * option(option vl)`	$t,\ H,\ \sigma \Downarrow v,\ \sigma'$	`Reduction`

The semantics in Rocq, i.e., `teval`, extends the big-step semantics $\Downarrow$ to a total evaluation function, making a distinction between timeout(`None`), errors (`Some None`), and normal values (`Some(Somev)`).

A.3 Type System (Paper (Figure 1))

`G`	Γ	`Typing Environment`
`has_type G t T`	$\Gamma \vdash t : T$	`Typing`

A.4 Logical Relations (Paper (Figure 2 Without Highlighted))

`M`	Σ	`Store Typing`
`S`	σ	`Store`
`store_type S M`	$\sigma : \Sigma$	`Well-Defined Store` *`w.r.t.`* `Store Typing`
`st_chain M M1`	$\Sigma \sqsubseteq \Sigma_1$	`Store Typing Monotonicity`
`val_type M H v T`	$V[[\mathrm{T}]]$	`Value Interpretation of Types`
`exp_type S M H t T`	$E[[T]]$	`Term Interpretation of Types`
`env_type M H G`	$G[[\Gamma]]$	`Semantic Typing Context Interpretation`
`sem_type G t T`	$\Gamma \models t : T$	`Semantic Typing Judgment`

A.5 Soundness Proofs

1. Compatibility Lemmas: Lemmas `sem_true`, `sem_false`, `sem_var`, `sem_ref`, `sem_get`, `sem_put`, `sem_abs`, `sem_app` and `sem_seq`.
2. Fundamental Theorem (Theorem 2.1 in paper): Theorem `fundamental_property`.
3. Adequacy of Unary Logical Relations (Theorem 2.2 in paper): Corollary `safety`.

B ROCQ MECHANIZATION FOR SECTION 2 IN PAPER (WITH HIGHLIGHTED)

We outline the correspondence between the formalism in Section 2 (with highlighted) of the paper and its implementation in Rocq.

B.1 Model (sec2_stlc_highlighted.v)

The definitions of types and terms, values, semantics and type systems are the same as what are defined in Appendix A, thus are omitted.

B.2 Logical Relations(Paper (Figure 2 with Highlighted))

`fv(m : nat)(t : tm) : ql`	$\mathrm{fv}(t)$	`Free variables from a given term`
`val_locs(v : vl)`	$L(v)$	`Reachable locations from a value`
`vars_locs (E : venv)` FV`(t)` ℓ	$\ell \in [[FV(t)]]_H$	`Reachable locations from a term`
`store_type S M`	$\sigma : \Sigma$	`Well-Defined Store` *`w.r.t.`* `Store Typing`
`st_chain M M1 q`	$\Sigma \equiv_q \Sigma_1$	`Relational Store Typing`
`store_effect S S1 p`	$\sigma \longrightarrow_p \sigma_1$	`Value Preservation`
`val_qual M M1 H v q`	$L(v) \subseteq [[q]]_H \cup \overline{\mathrm{dom}(\Sigma)}$	`Value Reachability`
`val_type M H v T`	$V[[\mathrm{T}]]$	`Value Interpretation of Types`
`exp_type S M H t T`	$E[[T]]$	`Term Interpretation of Types`
`env_type M H G p`	$G[[\Gamma^p]]$	`Semantic Typing Context Interpretation`
`sem_type G t T`	$\Gamma \models t : T$	`Semantic Typing Judgment`

In the Rocq implementation, the definition of `vars_locs` has parameter `q`, which is instantiated with the free variables of a given term in the proof context. To avoid confusion, we write $FV(\mathtt{t})$ in the above. The paper formalization for reference type interpretation is simplified due to the limited of space.

B.3 Soundness Proofs

1. The Time Travelling Property (Lemma 2.3 in the paper): Lemma `val_store_change`.
2. Compatibility Lemmas: Lemmas `sem_true`, `sem_false`, `sem_var`, `sem_ref`, `sem_get`, `sem_put`, `sem_abs`, `sem_app` and `sem_seq`.
3. Fundamental Theorem: Theorem `fundamental_property`.
4. Adequacy of Unary Logical Relations: Corollary `safety`.

C ROCQ MECHANIZATION FOR SECTION 3.1-3.4 IN PAPER

We outline the correspondence between the formalism in Section 3.1-3.4 of the paper and its implementation in Rocq.

C.1 Model (sec3_reach.v)

The definitions of terms, values, and semantics are the same as what are defined in Appendix A, thus are omitted.

`p, q`	$\varphi, q \in \mathcal{P}_{\mathsf{fin}}(\mathsf{Var})$	`Observations`
`fr1, fr2`	$\blacklozenge$	`Freshness Marker`
`fn1, fn2`	$\spadesuit$	`Self-Reference Marker`
`fr q`	$p \in \mathcal{P}_{\mathsf{fin}}(\mathsf{Var} \uplus \{\blacklozenge\})$	`Reachability Qualifiers`
`fn fr s`	$s \in \mathcal{P}_{\mathsf{fin}}(\mathsf{Var} \uplus \{\blacklozenge\} \uplus \{\spadesuit\})$	`Function Domain/Codomain qualifier`

In the Rocq formalization, we represent variable and qualifier sets using the type `pl`, e.g., `p : pl` means that `p` denote a set of variables. We use boolean values to denote the freshness marker, self-reference marker and parameters appeared in qualifiers. For example, `fr p` correspondences to qualifiers that may include the freshness marker $\blacklozenge$ in paper formalization, where the symbol p is often used.

`ty`	$S, T, U, V :=$	`Type`
`TBool`	Bool	`Boolean Type`
`TRef fr p T`	$\mathsf{Ref}\ T^{p}$	`Reference Type`
`TFun T fn1 fr1 s U fn2 ar2 fr2 r`	$(x : T^{s}) \to U^{r}$	`Function Type`

The following lists examples of reference types encoded in Rocq and expressed in paper.

`TRef false q T`	$\mathsf{Ref}\ T^{q}$	`Both q and` q `are variables.`
`TRef true p T`	$\mathsf{Ref}\ T^{p}$, `where` $\blacklozenge \in p$	
`TRef fr p T`	$\mathsf{Ref}\ T^{p}$	

Function types are similar. For example, the type `TFun T false true s U false true false r` in Rocq correspondences to $(x : T^{s}) \to U^{r}$, where $\spadesuit \notin s$, $\blacklozenge \in s$, $\spadesuit \notin r$, $x \in r$ and $\blacklozenge \notin r$. It means that the argument is fresh, or does not include the self-reference, and the return value's qualifier includes the bound variable (i.e., it may alias with the argument), but is not fresh or aliased with the function.

C.2 Type System

`G`	Γ	`Typing Environment`
`has_type G t T p fr q`	$\Gamma^{\varphi} \vdash t : T^{p}$	`Typing`

Here, `p` correspondences to φ, and `fr q` correspondences to p. In the later section, we often write φ instead of `p` for convenience.

C.3 Reachability

val_locs(v : vl)	$L(v)$	`Reachable locations from a given value`
vars_locs (E : venv) q ℓ	$\ell \in [[q]]_H$	`Reachable locations from reachability qualifier`
vars_trans' G q	$\Gamma \vdash q*$	`qualifier Saturation`

C.4 Logical Relations

store_type S M	$\sigma : \Sigma$	`Well-Defined Store`
st_chain M M1 q	$\Sigma \equiv_q \Sigma_1$	`Relational Store Typing`
store_effect S S1 p	$\sigma \longrightarrow_p \sigma_1$	`Value Preservation`
val_qual M M1 H v φ fr q	$L(v) \subseteq [[\varphi \cap q]]_H \cup_{\texttt{fr}=true} \overline{\mathrm{dom}(\Sigma)}$	`Value Reachability`
val_type M H v T	$V[[\mathrm{T}]]$	`Value Interpretation of Types`
exp_type S M H t Tφ fr q	$E[[T^p]]_\varphi$	`Term Interpretation of Types`
env_type M H G p	$G[[\Gamma^p]]$	`Typing Context Interpretation`
sem_type G t T φ fr p	$\Gamma^\varphi \models t : T^p$	`Semantic Typing Judgment`

C.5 Soundness Proofs

1. The Time Travelling Property: Lemma `val_store_change`.
2. Encapsulated Computations (Lemma 3.1 in paper): Lemma `encapsulation`.
3. Compatibility Lemmas (Semantic Typing Rules (Figure 6 in Paper)): Lemmas `sem_true`, `sem_false`, `sem_var`, `sem_ref`, `sem_get`, `sem_put`, `sem_abs`, `sem_app` and `sem_seq`.
4. Fundamental Theorem: Theorem `fundamental_property`.
5. Adequacy of Unary Logical Relations: Corollary `safety`.

D ROCQ MECHANIZATION FOR SECTION 3.5 IN EXTENDED VERSION OF THE PAPER [Bao et al. 2025b]

We outline the correspondence between the formalism in Section 3.5 of the extended version of the paper [Bao et al. 2025b] and its implementation in Rocq.

D.1 Model (sec3_reach_sub.v)

The definitions of terms, values, semantics, and types are the same as what are defined in Appendix A, thus are omitted.

D.2 Logical Relations

sem_qtp G q1 q2	$\Gamma \models q_1 <: q_2$	`Interpretation of Subqualifiers`
sem_stp G T1 T2	$\Gamma \models T_1 <: T_2$	`Interpretation of Subpretypes`
sem_sqtp G φ T1 fr1 p1 T2 fr2 p2	$\Gamma^\varphi \models T_1^{p_1} <: T_2^{p_2}$	`Interpretation of Subtypes`

D.3 Soundness Proofs

1. Semantic Subqualifiers Lemmas (Figure 7 in the extended version of the paper [Bao et al. 2025b]): Lemmas `sem_qtp_sub`, `sem_qtp_var` and `sem_qtp_cong`.
2. Semantic Subpretypes Lemmas (Figure 7 in the extended version of the paper [Bao et al. 2025b]): Lemmas `sem_stp_bool`, `sem_stp_ref` and `sem_stp_fun`.
3. Fundamental Theorem for Subqualifiers (Lemma 3.2 in the extended version of the paper [Bao et al. 2025b]): Theorems `qtp_fundamental` and `qtp_fr_fundamental`.
4. Fundamental Theorem for Subpretypes (Lemma 3.3 in the extended version of the paper [Bao et al. 2025b]): Theorem `stp_fundamental`.

5. Fundamental Theorem for Subtypes (Lemma 3.4 in the extended version of the paper [Bao et al. 2025b]) : Theorem `sqtp_fundamental`.
6. Fundamental Theorem: Theorem `fundamental_property`.
7. Adequacy of Unary Logical Relations: Corollary `safety`.
8. Encapsulated Computations: Lemma `encapsulation`.

E ROCQ MECHANIZATION FOR SECTION 3.5 IN PAPER

We outline the correspondence between the formalism in Section 3.5 of the paper and its implementation in Rocq.

E.1 Model (sec4_reach_nested.v)

The definitions of terms, values, and semantics are the same as what are defined in Appendix A, thus are omitted. The definitions of types are the same as what are defined in Appendix C, thus are omitted.

E.2 Reachability

`locs_locs_stty` M $\mathbb{L}$	$\mathbb{L}^*_\Sigma$	`Location Saturation`

E.3 Logical Relations

The definition of logical relations extends from Appendix C, i.e., they use the same signatures.

E.4 Soundness Proofs

1. The Time Travelling Property (Lemma 3.2 in paper): Lemma `val_store_change`.
2. Encapsulated Computations: Lemma `encapsulation`.
3. Compatibility Lemmas (Semantic Typing Rules (Figure 7 in Paper)): Lemmas `sem_true`, `sem_false`, `sem_var`, `sem_ref`, `sem_get`, `sem_put`, `sem_abs`, `sem_app` and `sem_seq`.
4. Fundamental Theorem: Theorem `fundamental_property`.
5. Adequacy of Unary Logical Relations: Corollary `safety`.

F ROCQ MECHANIZATION FOR SECTION 3.6 IN PAPER

We outline the correspondence between the formalism in Section 3.6 of the paper and its implementation in Rocq.

F.1 Model (sec5_reach_nested_effs.v)

The definitions of terms, values, and semantics are the same as what are defined in Appendix A, thus are omitted.

`e`	$\varepsilon \in \mathcal{P}_{\text{fin}}(\text{Var})$	`Effects`
`ty`	$S, T, U, V :=$	`Type`
`TBool`	Bool	`Boolean Type`
`TRef fr q T`	Ref T^q	`Reference Type`
`TFun T fn1 fr1 s U fn2 ar2 fr2 r e2f e2x e2`	$(x : T^s) \to^{\varepsilon} U^r$	`Function Type`

F.2 Type System

`G`	Γ	`Typing Environment`
`has_type G t T` φ `fr p e`	$\Gamma^{\varphi} \vdash t : T^{p}\ \varepsilon$	`Typing`

F.3 Logical Relations

The definitions of logical relations follow those in Appendix E. The following lists the differences:

`exp_type S M H t T`φ `fr q e`	$E[[T^p\ \varepsilon]]_\varphi$	`Term Interpretation of Types`
`sem_type G t T` φ `fr p e`	$\Gamma^\varphi \models t : T^p\ \varepsilon$	`Semantic Typing Judgment`

F.4 Soundness Proofs

1. The Time Travelling Property: Lemma `val_store_change`.
2. Encapsulated Computations (Lemma 3.6 in the extended version of the paper [Bao et al. 2025b]): Lemma `encapsulation`.
3. Compatibility Lemmas (Semantic Typing Rules (Figure 8 in Paper, and Figure 10 in the extended version of the paper [Bao et al. 2025b])): Lemmas `sem_true`, `sem_false`, `sem_var`, `sem_ref`, `sem_get`, `sem_put`, `sem_abs`, `sem_app` and `sem_seq`.
4. Fundamental Theorem: Theorem `fundamental_property`.
5. Adequacy of Unary Logical Relations: Corollary `safety`.

G ROCQ MECHANIZATION FOR SECTION 4 IN PAPER

We outline the correspondence between the formalism in Section 4 of the paper and its implementation in Rocq.

G.1 Model (sec6_reach_binary.v and sec6_reach_binary_effs.v)

The definitions of terms, values, and semantics are the same as what are defined in Appendix A, thus are omitted. The models sec6_reach_binary.v and sec6_reach_binary_effs.v correspondence formalism in teal and in pink in Figure 9 of the paper, respectively.

`ctx_type`	$C : (\Gamma^\varphi; T^p\ \varepsilon) \Rrightarrow (\Gamma'^{\varphi'}; T'^{p'}\ \varepsilon')$	`Context Typing`
`context_equiv`	`Boolean Context Refinement in Section 6 of Supplement [Bao et al. 2025b]`	`Definition of Contextual Equivalence`

In the definition of context typing (`ctx_type`), we define one-step context rules, and rule `cx_trans` is used for composing those context typing rules.

G.2 The World Model

`M : stty`	W	`Wolrd`
`S : stor`	σ	`Store`
`st_filter M L1 L2`	$\mathrm{WR}(\mathrm{W}, L_1, L_2)$	`Well-Formed Relations`
`st_chain_partialM M1 L1 L2`	$\mathrm{W} \equiv_{(L_1, L_2)} \mathrm{W}_1$	`Relational Worlds`
`store_type S1 S2 M`	$(\sigma_1, \sigma_2) : \mathrm{W}$	`Well-Defined Stores`

G.3 Logical Relations

`val_type M H1 H2 v1 v2 T1T2`	$V[[\mathrm{T}_1, \mathrm{T}_2]]$	`Value Interpretation of Types`
`exp_type S1 S2 M H1 H2 t1 t2 T` φ `fr q` e	$E[[T^p\ \varepsilon]]_\varphi$	`Term Interpretation of Types`
`env_type M H1 H2 G p`	$G[[\Gamma^p]]$	`Typing Context Interpretation`
`sem_type G` t_1 t_2 `T` φ `fr p` e	$\Gamma^\varphi \models t_1 \approx_{\log} t_2 : T^p\ \varepsilon$	`Semantic Typing Judgment`

G.4 Soundness Proofs

1. Semantic Typing Context Tightening (Lemma 4.3 in paper): Lemma `envt_tighten`.
2. Relation Tightening (Lemma 4.4 in paper): Lemma `envt_filter_deep`.

3. Relational Worlds Tightening (Lemma 4.5 in paper): Lemma `stchain_tighten`.
4. The Time Travelling for binaries (Lemma 4.6 in Paper): Lemma `val_store_change`.
5. Encapsulated Computations: Lemma `encapsulation`.
6. Compatibility Lemmas: Lemmas `sem_true`, `sem_false`, `sem_var`, `sem_ref`, `sem_get`, `sem_put`, `sem_abs`, `sem_app` and `sem_seq`.
7. Fundamental Theorem (Theorem 4.7 in paper): Theorem `fundamental_property`.
8. Congruence of binary Logical Relations (Lemma 4.8 in paper): Theorem `congr`.
9. Adequacy of binary Logical Relations (Lemma 4.9 in paper): Corollary `safety`.
10. Soundness of binary Logical Relations (Lemma 4.10 in paper): Theorem `soundness`.

H ROCQ MECHANIZATION FOR SECTION 5 IN PAPER

We outline the correspondence between the formalism in Section 5 of the paper and its implementation in Rocq.

H.1 Model (sec7_beta.v, sec7_store_invariants.v, sec7_reorder.v, sec7_reorder_effs.v and sec7_store_invariants_effs.v)

The files sec7_beta.v and sec7_reorder.v are proofs of rules β-EQUIV and RE-ORDER-$\lambda^{\blacklozenge}$ in Figure 10 of paper, respectively, where the file sec7_store_invariants.v are store invariants used in those proofs.

The file sec7_reorder_effs.v is the proof of rule RE-ORDER-$\lambda^{\blacklozenge}_{\varepsilon}$ in Figure 10 of the paper, where the file sec7_store_invariants_effs.v are store invariants used in the proof.

H.2 Soundness Proofs

1. Term Equivalence Preservation (Lemma 5.1 in paper): Lemma `store_invariance2` in file sec7_store_invariants_effs.v.
2. Term Equivalence Preservation & Reachability (Lemma 5.2 in paper): Lemma `store_invariance2'` in file sec7_store_invariants_effs.v.
3. Soundness of rule RE-ORDERING (Lemma 5.3 in paper): Theorem `reorder_seq` in file sec7_reorder.v.
4. Soundness of rule RE-ORDERING-$\lambda^{\blacklozenge}_{\varepsilon}$ (Lemma 5.3 in paper): Theorem `reorder_seq` in file sec7_reorder_effs.v.
5. Lemma 5.4 in paper: Theorem `store_invariance` in file sec7_store_invariants.v.
6. Semantic Weakening (Lemma 5.5 in ppaer): Lemma `st_weaken1` in file sec7_beta.v.
7. Semantic Substitution (Lemma 5.6 in paper): Lemma `st_subst1` in file sec7_beta.v.
8. Soundness of rule β-EQUIV: Theorem `beta_equivalence` in file sec7_beta.v.

REFERENCES

Yuyan Bao, Songlin Jia, Guannan Wei, Oliver Bračevac, and Tiark Rompf. 2025a. Modeling Reachability Types with Logical Relations. *Proc. ACM Program. Lang.* 9, OOPSLA (2025).

Yuyan Bao, Songlin Jia, Guannan Wei, Oliver Bračevac, and Tiark Rompf. 2025b. Modeling Reachability Types with Logical Relations: Semantic Type Soundness, Termination, and Equational Theory (Extended Version). arXiv:2309.05885 [cs.PL]