

Authority Engineering RC-1 Machine-Readable Formal Theory

Verbatim SMT-LIB 2 source edition

```
; L-AE Formal Theory Presentation 1.0
; Canonical formal sentence presentation compiled from the existing RC-1 constraints.
; This file adds no AE primitive, law, or empirical claim.
(set-logic ALL)
(declare-sort Agent 0)
(declare-sort Analysis 0)
(declare-sort Assessment 0)
(declare-sort Attack 0)
(declare-sort Decision 0)
(declare-sort Degree 0)
(declare-sort ExecutionCondition 0)
(declare-sort Fabric 0)
(declare-sort Flow 0)
(declare-sort Future 0)
(declare-sort Guard 0)
(declare-sort Intention 0)
(declare-sort Organization 0)
(declare-sort Outcome 0)
(declare-sort Path 0)
(declare-sort Role 0)
(declare-sort Stack 0)
(declare-sort State 0)
(declare-sort Surface 0)
(declare-sort System 0)
(declare-sort TFH 0)
(declare-sort TFV 0)
(declare-sort Time 0)
(declare-sort Topology 0)
(declare-sort Transition 0)
(declare-fun src (Transition) State)
(declare-fun tgt (Transition) State)
(declare-fun A (State Transition) Bool)
(declare-fun Accumulated (Transition Time) Bool)
(declare-fun Assesses (Assessment Outcome State Transition ExecutionCondition) Bool)
(declare-fun AuthorityAttack (Attack) Bool)
(declare-fun AuthorizedToInitiate (Role Decision) Bool)
(declare-fun AuthorizedToInterrupt (Role Decision) Bool)
(declare-fun AuthorizedToOwnConsequence (Role Decision) Bool)
(declare-fun BAR (State Transition) Bool)
(declare-fun BeforeH (TFH Guard Guard) Bool)
(declare-fun BoundedOutcome (Outcome State Transition ExecutionCondition) Bool)
(declare-fun BreaksSystem (Attack) Bool)
(declare-fun C (Transition) Bool)
(declare-fun CCR (Transition) Bool)
(declare-fun CarriedBy (Fabric Path State Transition System) Bool)
(declare-fun Carries (Path State Transition) Bool)
(declare-fun Contends (Agent Agent Intention Intention Future Future) Bool)
(declare-fun Continuous (Path) Bool)
(declare-fun DecisionRelation (Decision State Transition) Bool)
(declare-fun DegreeOf (Assessment Degree) Bool)
(declare-fun Distributed (System) Bool)
(declare-fun DistributedRealization (State Transition System) Bool)
```

Publication Record

Authority Engineering RC-1 Machine-Readable Formal Theory: Verbatim SMT-LIB 2 Source Edition

Author: Rob Caswell

Publisher: Cyber Terrain LLC

Published: 2026-01-01

Zenodo version: v1.0

DOI: <https://doi.org/10.5281/zenodo.21742384>

Website: <https://cyberterrain.ai/papers/rc1-formal-theory-presentation.pdf>

Corpus Context

Corpus: Authority Engineering

Document ID: AE-RC1-FORMAL-0002

Object family: AE-RC1-FORMAL

Publisher-imprint edition prepared for versioned republication.

The prior Zenodo record remains the citable record until its new version is published.

```

(declare-fun Dynamic (ExecutionCondition) Bool)
(declare-fun Earlier (Time Time) Bool)
(declare-fun FabricOf (Fabric System) Bool)
(declare-fun FidelityAssessment (Assessment) Bool)
(declare-fun FlowInTopology (Flow Topology State Transition) Bool)
(declare-fun FlowRelation (Flow State Transition) Bool)
(declare-fun FutureCondition (State Transition Time) Bool)
(declare-fun HAt (TFH State Transition) Bool)
(declare-fun HMember (TFH Guard) Bool)
(declare-fun HOwner (TFH Guard) Bool)
(declare-fun Holds (Agent Intention) Bool)
(declare-fun Identifiable (Role) Bool)
(declare-fun Incompatible (Intention Intention) Bool)
(declare-fun IntendedAuthority (State Transition) Bool)
(declare-fun IntendedRelation (Intention State Transition) Bool)
(declare-fun Legitimate (Flow) Bool)
(declare-fun Local (State Transition) Bool)
(declare-fun LocallyValid (Agent Intention) Bool)
(declare-fun Manipulates (Attack Flow) Bool)
(declare-fun Material (Decision) Bool)
(declare-fun Meaningful (State Transition) Bool)
(declare-fun Member (Role Organization) Bool)
(declare-fun Modifies (Transition State Transition Time Time) Bool)
(declare-fun Observable (State Transition) Bool)
(declare-fun OnSurface (Surface State Transition) Bool)
(declare-fun Over (Intention Future) Bool)
(declare-fun OwnershipTest (Decision Organization) Bool)
(declare-fun Owns (Guard State Transition) Bool)
(declare-fun Participates (Guard State Transition) Bool)
(declare-fun Preq (State Transition) Bool)
(declare-fun Present (Guard State) Bool)
(declare-fun Preserved (Path State Transition) Bool)
(declare-fun Proceed (State Transition) Bool)
(declare-fun Reachable (State Transition) Bool)
(declare-fun Realized (Outcome) Bool)
(declare-fun SMember (Stack Guard) Bool)
(declare-fun STOICAnalysis (Analysis) Bool)
(declare-fun SharedOrIntersecting (Future Future) Bool)
(declare-fun Sigma (State Transition) Bool)
(declare-fun StackAt (Stack State) Bool)
(declare-fun Stop (State Transition) Bool)
(declare-fun Studies (Analysis Transition State Transition Surface Topology Time Time) Bool)
(declare-fun Supplies (Guard State Transition) Bool)
(declare-fun SurfaceOf (Surface System) Bool)
(declare-fun TopoMember (Topology State Transition) Bool)
(declare-fun TopologyOf (Topology System) Bool)
(declare-fun TraversalOrMutation (Path) Bool)
(declare-fun VAT (TFV State Transition) Bool)
(declare-fun VMember (TFV Guard) Bool)
(declare-fun Within (State System) Bool)
; RC-1
(assert (forall ((s State) (t Transition)) (=> (Sigma s t) (and (A s t) (C t) (= (src t) s)))))
; BAR-definition
(assert (forall ((s State) (t Transition)) (= (BAR s t) (and (Sigma s t) (A s t) (C t) (Local s t) (not (CCR t))))))
; GC-1
(assert (forall ((g Guard) (s State) (t Transition)) (=> (Supplies g s t) (and (Present g s) (Sigma s t) (BAR s t)))))
; GC-2
(assert (forall ((g Guard) (s State) (t Transition)) (=> (Owns g s t) (and (Present g s) (Participates g s t) (Sigma s t) (BAR s t)))))
; HC-1

```

```

(assert (forall ((h TFH) (s State) (t Transition)) (=> (HAt h s t) (and (Sigma s t) (BAR s t)))))
; HC-2
(assert (forall ((h TFH) (g Guard) (s State) (t Transition)) (=> (and (HMember h g) (HAt h s t))
(Participates g s t)))))
; HC-3
(assert (forall ((h TFH) (g Guard) (s State) (t Transition)) (=> (and (HOwner h g) (HAt h s t)) (and
(HMember h g) (Owns g s t)))))
; HC-4
(assert (forall ((h TFH) (g Guard) (gp Guard)) (=> (BeforeH h g gp) (and (HMember h g) (HMember h gp)
(distinct g gp)))))
; VC-1
(assert (forall ((v TFV) (s State) (t Transition)) (=> (VAt v s t) (Sigma s t)))))
; VC-2
(assert (forall ((v TFV) (g Guard) (s State) (t Transition)) (=> (and (VMember v g) (VAt v s t)) (and
(Present g s) (not (Participates g s t)))))
; SC-1
(assert (forall ((k Stack) (g Guard) (s State)) (=> (and (SMember k g) (StackAt k s)) (Present g s)))))
; SS-1
(assert (forall ((s State) (t Transition)) (=> (Stop s t) (and (Sigma s t) (BAR s t)))))
; SS-2
(assert (forall ((s State) (t Transition)) (=> (Stop s t) (= (Proceed s t) (Preq s t)))))
; CT-1
(assert (forall ((p Path)) (=> (Continuous p) (TraversalOrMutation p)))))
; CT-2
(assert (forall ((p Path) (s State) (t Transition)) (=> (and (Continuous p) (Carries p s t)) (and (A s t)
(C t) (BAR s t) (Preserved p s t) (Meaningful s t)))))
; AS-1
(assert (forall ((u Surface) (s State) (t Transition)) (=> (OnSurface u s t) (and (A s t) (C t)
(Reachable s t) (Observable s t)))))
; AS-2
(assert (forall ((u Surface) (x System) (s State) (t Transition)) (=> (and (SurfaceOf u x) (OnSurface u
s t)) (Within s x)))))
; AT-1
(assert (forall ((q Topology) (s State) (t Transition)) (=> (TopoMember q s t) (and (A s t) (C t)
(Reachable s t)))))
; AT-2
(assert (forall ((q Topology) (x System) (s State) (t Transition)) (=> (and (TopologyOf q x)
(TopoMember q s t)) (Within s x)))))
; FB-1
(assert (forall ((f Fabric) (x System)) (=> (FabricOf f x) (Distributed x)))))
; FB-2
(assert (forall ((f Fabric) (p Path) (s State) (t Transition) (x System)) (=> (CarriedBy f p s t x)
(and (FabricOf f x) (Carries p s t) (Continuous p) (A s t) (C t) (DistributedRealization s t x)))))
; FB-3
(assert (forall ((s State) (t Transition) (x System)) (=> (DistributedRealization s t x) (exists ((f
Fabric) (p Path)) (CarriedBy f p s t x)))))
; ST-1
(assert (forall ((a Analysis) (cc Transition) (s State) (t Transition) (u Surface) (q Topology) (t0
Time) (t1 Time)) (=> (and (STOICAnalysis a) (Studies a cc s t u q t0 t1)) (and (C cc) (OnSurface u s t)
(TopoMember q s t) (BAR s t)))))
; ST-2
(assert (forall ((a Analysis) (cc Transition) (s State) (t Transition) (u Surface) (q Topology) (t0
Time) (t1 Time)) (=> (and (STOICAnalysis a) (Studies a cc s t u q t0 t1)) (and (Accumulated cc t0)
(Earlier t0 t1) (FutureCondition s t t1) (Modifies cc s t t0 t1)))))
; CC-1
(assert (forall ((a Agent) (b Agent) (i Intention) (j Intention) (f Future) (g Future)) (=> (Contains a
b i j f g) (and (distinct a b) (distinct i j) (Holds a i) (Holds b j) (LocallyValid a i) (LocallyValid
b j) (Incompatible i j) (Over i f) (Over j g) (SharedOrIntersecting f g)))))
; CC-2
(assert (forall ((a Agent) (b Agent) (i Intention) (j Intention) (f Future) (g Future)) (=> (Contains a
b i j f g) (exists ((s State) (t Transition) (u State) (v Transition) (q Topology) (p Path) (r Path))

```

```

(and (IntendedRelation i s t) (IntendedRelation j u v) (A s t) (C t) (A u v) (C v) (TopoMember q s t)
(TopoMember q u v) (Carries p s t) (Continuous p) (Carries r u v) (Continuous r))))))
; OT-1
(assert (forall ((d Decision) (o Organization)) (=> (OwnershipTest d o) (Material d))))
; OT-2
(assert (forall ((d Decision) (o Organization)) (=> (OwnershipTest d o) (exists ((ri Role) (rx Role)
(ro Role)) (and (Identifiable ri) (Identifiable rx) (Identifiable ro) (Member ri o) (Member rx o)
(Member ro o) (AuthorizedToInitiate ri d) (AuthorizedToInterrupt rx d) (AuthorizedToOwnConsequence ro
d))))))
; OT-3
(assert (forall ((d Decision) (o Organization)) (=> (OwnershipTest d o) (exists ((s State) (t
Transition)) (and (DecisionRelation d s t) (A s t) (C t))))))
; AA-1
(assert (forall ((k Attack)) (=> (AuthorityAttack k) (exists ((f Flow) (q Topology) (s State) (t
Transition)) (and (Manipulates k f) (Legitimate f) (FlowRelation f s t) (A s t) (FlowInTopology f q s
t) (TopoMember q s t))))))
; AA-2
(assert (forall ((k Attack)) (=> (AuthorityAttack k) (not (BreaksSystem k))))))
; CF-1
(assert (forall ((a Assessment) (o Outcome) (s State) (t Transition) (e ExecutionCondition)) (=> (and
(FidelityAssessment a) (Assesses a o s t e)) (and (Realized o) (IntendedAuthority s t) (A s t) (C t)
(Dynamic e) (BoundedOutcome o s t e))))
; CF-2
(assert (forall ((a Assessment) (o Outcome) (s State) (t Transition) (e ExecutionCondition)) (=> (and
(FidelityAssessment a) (Assesses a o s t e)) (exists ((d Degree)) (DegreeOf a d))))
(check-sat)

```