

A6 Composition Transition Predicate Addendum v0.1.4

EXIT / SPLIT / HOLD / DISSOLVE classification for A6 composition under governed binding

Kotov Ivan

2026-07-06

Abstract

This third-pass convergent review-integrated bounded addendum clarifies the A6 transition predicate for EXIT, SPLIT, HOLD/FREEZE, DISSOLVE, and CONTINUE. It preserves the unique-lawful-successor threshold while closing the final leaf-level predicate-completeness issue: actor_or_role_departed is now explicitly classed and guarded.

Contents

Document Metadata	3
0. Executive definition	5
0.1 Review-integration delta from v0.1.1, v0.1.2, and v0.1.3	5
0.2 Predicate-review convergence note	6
1. Precedence and non-replacement	6
1.1 Parent artifact remains fixed	6
1.2 Clarification role	6
1.3 Future consolidation	6
1.4 Claim-strength map	7
1.5 Conflict handling	7
2. Purpose	7
3. Scope and non-goals	8
3.1 In scope	8
3.2 Out of scope	8
3.3 Non-claim	9
4. Corpus bridge set	9
4.1 Explicit bridge: $c = a + b$ and governed binding	9
4.2 Hidden bridge I: Ashby and requisite variety	9
4.3 Hidden bridge II: information theory and unique successor state	10
4.4 Hidden bridge III: engineering inspection and load paths	10
4.5 Earth paragraph	10
5. Definitions	10
5.1 Composition state K	10

5.2 Event e	11
5.3 Lawful successor composition	11
5.4 Predicate class map	11
5.5 Bounded no-claim status	13
5.6 Lawful successor candidate	13
5.7 Unique lawful successor	13
5.8 Incompatible successor claims	14
5.9 Successor-space exhaustion record	14
5.10 Temporal window state	15
6. Transition operator	15
7. Core classification rule	16
7.1 Decision order	16
7.2 Classification table	17
8. EXIT predicate	17
8.1 Definition	17
8.2 Required checks	18
8.3 Invalid EXIT reductions	18
8.4 EXIT examples	18
9. SPLIT predicate	19
9.1 Definition	19
9.2 Required checks	19
9.3 Invalid SPLIT reductions	19
9.4 SPLIT examples	20
10. HOLD / FREEZE predicate	20
10.1 Definition	20
10.2 HOLD is not continuation	21
10.3 FREEZE escalation	21
10.4 HOLD examples	21
11. DISSOLVE predicate	21
11.1 Definition	21
11.2 DISSOLVE default rule	22
11.3 Required checks	22
11.4 DISSOLVE is not SPLIT	22
11.5 DISSOLVE examples	23
12. CONTINUE predicate	23
12.1 Definition	23
12.2 CONTINUE must not hide EXIT	23
13. Critical invariants	24
14. Witness binding	24
14.1 Required witness events	24
14.2 Minimum event fields	25
14.3 Witness anti-laundering rule	26
15. Semantic checker rule names	26

15.1 Predicate status vocabulary	27
15.2 Checker result vocabulary	27
15.3 Checker non-goals	28
16. Fixture index seed	28
17. Implementation mapping note	29
17.1 Machine/human boundary	30
17.2 WINDOWED determinism rule	30
17.3 Witness namespace implementation rule	31
18. Relationship to ARL	31
19. Relationship to $c = a + b$	31
20. Public claim discipline	32
21. Red-line failures	32
22. Compact operational rule	33
22.1 Non-machine-decidability guard	34
22.2 DISSOLVE guard	35
22.3 SPLIT incompatibility guard	35
22.4 WINDOWED time guard	35
22.5 Actor / role departure guard	35
23. Changelog	35
v0.1.4 — 2026-07-06	35
v0.1.3 — 2026-07-06	36
v0.1.2 — 2026-07-06	37
v0.1.1 — 2026-07-06	37
24. Closing invariant	38

Document Metadata

Status: Draft bounded addendum v0.1.4; convergent b-layer review integrated

Date: 2026-07-06

Document ID: A6_Composition_Transition_Predicate_Addendum_v0_1_4

Short name: A6-CTP v0.1.4

Layer: A6 / $c = a + b$ / governed binding / ARL / standing / witness / re-entry / composition continuity

Document class: bounded addendum / transition-predicate clarification / control-layer companion

Assertion class: section-mapped; normative clarification (C-A4), witness/hash claims (C-A7), control-layer and checker guidance (C-A10)

Parent artifact: A6 v0.1 DOI-published document

Parent DOI: TBD: insert parent A6 DOI before publication

Replacement status: non-replacement; the parent DOI artifact remains unchanged

Deployment status: not deployment-ready

Conformance status: no conformance claim

Author: Kotov Ivan

Location: Bruxelles, Belgium

Year: 2026

Review integration: integrates b-layer review records A6CTP-REV-20260706-b, A6CTP-REV-20260706-b2, and A6CTP-REV-20260706-b3

Prior target hashes: v0.1.1 SHA be1a6a750b770cef7d00e149ffec5db8c8a2d070dce8c49a8f57a3d7
v0.1.2 SHA e5aa43d591cdfd7d24b58ac40fe9a6198f1eb13b36c9bd0cb2215b75ffd73d2e;
v0.1.3 SHA 3d62a70a1770a57712c28d9091197795e60c3a8a838870ea9bd18121880e55d6

0. Executive definition

This addendum makes explicit a transition predicate that is already implied by A6 and ARL discipline:

EXIT = removal with preserved unique lawful successor.

SPLIT = removal with lost unique lawful successor and remaining
 ↪ incompatible successor claims.

HOLD = evidence or standing insufficient to classify the transition.

DISSOLVE = no admissible successor composition remains, after witnessed
 ↪ successor-space exhaustion.

A6 composition is not a participant list.

A6 is a governed binding of A0–A5 under standing, role, obligation, quorum, witness, memory precedence, authority boundary, and exit law.

An actor leaving a composition does not automatically mean that the composition has split. It also does not automatically mean that the departure is a clean exit. The transition must be classified by whether the departure preserves or destroys a unique lawful successor composition.

Compact formula:

The threshold is not headcount.

The threshold is unique lawful continuation.

0.1 Review-integration delta from v0.1.1, v0.1.2, and v0.1.3

This revision preserves the v0.1.2 integration of b-layer review findings F1-F3, preserves the v0.1.3 integration of second-pass findings N1-N3, and accepts the third-pass finding T1 as bounded leaf-completeness input:

F1 accepted: `lawful_successors` is split into DECIDABLE, WINDOWED, and
 ↪ ESCALATE_ONLY subpredicates.

F2 accepted: DISSOLVE now requires a witnessed
 ↪ `successor_space_exhaustion_record`; otherwise HOLD.

F3 accepted: `claim_strength` is mapped by section instead of declared
 ↪ only at document level.

N1 accepted: `a6.*` is the canonical witness namespace; `arl.*` is a
 ↪ generated ARL-facing projection, not an alias fork.

N2 accepted: `successor_claims_incompatible` is classified as a predicate
 ↪ with DECIDABLE / WINDOWED / ESCALATE_ONLY handling.

N3 accepted: WINDOWED predicates require explicit temporal-window
 ↪ state; a checker must not use unstated wall-clock inference.

T1 accepted: `actor_or_role_departed` is now an explicitly classified
 ↪ predicate rather than an implicit leaf discriminator.

The core threshold remains unchanged:

EXIT = exactly one lawful successor.

SPLIT = multiple incompatible successor claims, after incompatibility
 ↪ status is resolved.

HOLD = unresolved evidence, standing, authority, window, escalation,
 ↪ incompatibility, or contested-departure state.

DISSOLVE = no admissible successor after witnessed exhaustion, not mere
↔ absence in the visible record.

0.2 Predicate-review convergence note

The third b-layer review record (A6CTP-REV-20260706-b3) characterizes the review sequence as convergent rather than open-ended. This addendum does not treat that reviewer assessment as proof. It records only the bounded process conclusion relevant to this draft:

F1 closed: `valid_successor` predicate classification.

N2 closed: `successor_claims_incompatible` predicate classification.

T1 closed: `actor_or_role_departed` predicate classification.

After T1, the successor-adjacent predicates used by the compact operational rule are explicitly named and classed. Remaining future work is expected to concern schemas, fixtures, implementation, review records, and conformance evidence rather than further predicate-completeness repair.

This is a draft-level review-cycle closure note. It is not a conformance claim.

1. Precedence and non-replacement

1.1 Parent artifact remains fixed

The DOI-published A6 v0.1 artifact remains the fixed parent reference.

This addendum does not alter, replace, silently amend, rewrite, or supersede the parent DOI artifact.

1.2 Clarification role

Where A6 v0.1 describes composition, standing, exit, split, review, witness, or re-entry behavior without an explicit transition predicate, this addendum supplies the explicit classification rule.

This addendum externalizes an implicit operator rule. It does not create a new root ontology.

1.3 Future consolidation

A future consolidated A6 v0.2 may incorporate this addendum directly.

Until then:

A6 v0.1 DOI artifact = frozen parent witness.

A6-CTP v0.1.4 = bounded transition-predicate companion.

1.4 Claim-strength map

This addendum uses section-level assertion classes. The document-level header is not a blanket upgrade.

Section range	Claim-strength class	Meaning
0-13, 18-21, 24	C - A4	draft normative clarification / operator vocabulary
14	C - A7 where hashes, witness events, signatures, or provenance claims are used; otherwise C - A10	witness-binding guidance and event vocabulary
15-17, 22	C - A10	checker / fixture / implementation-mapping guidance; no conformance claim
23	C - A7 only for stated hashes or review-record linkage; otherwise administrative changelog	version and review-integration record

No section of this addendum proves legal status, personhood, final safety, product certification, or deployment readiness.

Compact rule:

Claim class binds to the section and evidence class.
It does not bind to the document title alone.

1.5 Conflict handling

If this addendum conflicts with a stricter parent rule, the stricter fail-closed interpretation prevails.

If the parent A6 document and this addendum differ only by explicitness, this addendum controls the local classification vocabulary for:

CONTINUE / EXIT / SPLIT / HOLD_OR_FREEZE / DISSOLVE

2. Purpose

This addendum answers one operational question:

When one actor, role, anchor, claimant, or participant leaves an A6 composition, how do we determine whether the transition is EXIT, SPLIT, HOLD, or DISSOLVE?

The document prevents four common failures:

1. treating any departure as a clean EXIT;

2. treating any emotionally significant departure as SPLIT;
3. allowing a composition to continue silently when standing or authority became ambiguous;
4. using witness or review language after the fact to launder an unclassified transition.

The core requirement is simple:

No transition label without successor analysis.

No successor analysis without standing, obligation, quorum, authority,
 ↪ and witness checks.

3. Scope and non-goals

3.1 In scope

This addendum applies to A6 compositions where A0–A5 elements are bound into a governed composition and where a transition may affect:

- standing;
- role structure;
- obligation / claim / liability continuity;
- quorum or authority rules;
- witness lineage;
- memory precedence;
- re-entry legality;
- successor composition claims;
- composition identity or lawful continuation.

The addendum is relevant to:

actor exit;
 anchor departure;
 role loss;
 authority transfer;
 quorum fracture;
 project split;
 federation split;
 post-anchor review;
 lineage dispute;
 continuity-bearing composition change.

3.2 Out of scope

This addendum does not define:

- the full A0–A5 taxonomy;
- legal corporate dissolution in any jurisdiction;
- estate law;
- divorce or family law;
- general contract law;
- model training or parameter sharing;
- consciousness, personhood, or metaphysical identity;
- a complete ARL runtime implementation;

- a UI flow;
- a replacement for Beacon, ARL, AGL, L4 Witness, Continuity Bundle, or PAMDC.

Where actual law, medical authority, regulated professional standing, or institutional governance applies, this addendum routes to competent review rather than pretending to decide.

3.3 Non-claim

This addendum does not claim that every A6 composition can always be preserved.

It claims only this:

A6 transitions must not be labelled EXIT, SPLIT, HOLD, or DISSOLVE without a bounded classification predicate.

4. Corpus bridge set

4.1 Explicit bridge: $c = a + b$ and governed binding

In $c = a + b$, the sign $+$ is not ordinary addition.

It is a governed binding operator:

$c = \text{Bind}(a, b \mid \text{memory, witness, authority, L4, permission, continuity})$

The same rule applies to A6 composition.

A6 is not:

$A6 = A0 + A1 + A2 + A3 + A4 + A5$

as a flat aggregation.

A6 is:

$A6 = \text{Bind}(A0..A5 \mid \text{standing, role, obligation, quorum, witness, authority, exit-law})$

Therefore, EXIT and SPLIT are not informal descriptions. They are outcomes of a governed transition operator.

4.2 Hidden bridge I: Ashby and requisite variety

A composition survives environmental complexity by preserving enough legitimate variety in its internal regulators: roles, standing, obligations, review paths, memory, and authority checks.

An EXIT is safe only when the remaining composition still has sufficient lawful variety to regulate itself.

A SPLIT occurs when the composition loses the ability to maintain one lawful regulatory identity and instead branches into incompatible successor claims.

Headcount is not the relevant control variable. Regulator structure is.

4.3 Hidden bridge II: information theory and unique successor state

A transition reduces uncertainty by selecting one successor state from a space of possible successor states.

A clean EXIT is a low-ambiguity transition because one lawful successor state remains selected and witnessed.

A SPLIT is a high-ambiguity branching event because multiple incompatible successor states remain admissible.

A HOLD preserves unresolved entropy without laundering it into action.

A DISSOLVE declares that no successor state remains admissible.

4.4 Hidden bridge III: engineering inspection and load paths

A composition is like a load-bearing structure.

Some members can be removed after load transfer, inspection, and sign-off. Others are part of the critical load path. Removing them without a replacement does not produce a smaller intact structure. It produces a hazard state.

A6 transition classification is the inspection rule for that load path.

4.5 Earth paragraph

On a construction project, a worker may leave and the project continues. That is EXIT. But if the only licensed engineer leaves and the permit, signature authority, liability chain, and inspection standing no longer have one lawful continuation, the project has not merely lost a worker. It enters HOLD or FREEZE until standing is restored. If two parties then claim incompatible rights to continue the same project, documentation, budget, and authority chain, that is SPLIT. If nobody can lawfully continue it, that is DISSOLVE.

The threshold is not emotional importance.

The threshold is whether one lawful continuation still exists.

5. Definitions

5.1 Composition state K

For A6 transition analysis, a composition state K is treated as:

K = (

- P, participants / actors,
- R, roles,
- S, standing graph,
- O, obligations / claims / liabilities,
- Q, quorum / authority rules,
- W, witness lineage,
- M, memory / continuity references,
- X, exit rules,
- I, invariants,

T temporal window state / notice periods / challenge deadlines
)

This is not a replacement for A6. It is a transition-analysis projection. T is included so WINDOWED predicates are evaluated from recorded window state rather than from an implicit wall clock.

5.2 Event e

An event e is any occurrence that may change composition state:

actor departure;
 role revocation;
 anchor absence;
 authority challenge;
 quorum failure;
 obligation breach;
 witness-chain break;
 claimant split;
 memory-lineage conflict;
 external freeze;
 legal or institutional intervention.

5.3 Lawful successor composition

A successor composition K' is lawful only when the transition record separates three classes of predicates:

DECIDABLE = checkable from bounded record state.
 WINDOWED = checkable only after declared notice / challenge /
 ↪ review window.
 ESCALATE_ONLY = never closed by checker; requires ARL / human /
 ↪ competent authority disposition.

The phrase `valid_successor(K, e)` MUST NOT be implemented as a single opaque human judgment hidden inside a function name.

It is a structured result:

```
valid_successor(K, e) =
  all_required_decidable_predicates_pass
  ∧ all_required_windowed_predicates_closed_or_not_applicable
  ∧ all_required_escalate_only_predicates_resolved_or_not_applicable
  ∧ witness_appended
```

If any required WINDOWED or ESCALATE_ONLY predicate is unresolved, the successor is not rejected as impossible and not accepted as lawful. The transition enters HOLD/FREEZE.

5.4 Predicate class map

The transition record should classify each predicate explicitly. This map is intentionally written as a list rather than a compact table so that long predicate names remain readable in derived PDFs.

- `witness_appended`
 - Class: DECIDABLE.
 - Checker may decide: yes.
 - Default unresolved outcome: HOLD/FREEZE.
- `quorum_satisfiable_given_roster`
 - Class: DECIDABLE when roster and quorum rules are declared.
 - Checker may decide: yes, within the declared roster/rule scope.
 - Default unresolved outcome: HOLD/FREEZE.
- `role_reassigned_or_closed`
 - Class: DECIDABLE for explicit records; WINDOWED when notice or challenge windows apply.
 - Checker may decide: yes for explicit internal records; otherwise windowed.
 - Default unresolved outcome: HOLD/FREEZE.
- `obligation_rebound_or_closed`
 - Class: DECIDABLE, WINDOWED, or ESCALATE_ONLY depending on obligation type.
 - Checker may decide: only where obligation rules are declared and non-legal.
 - Default unresolved outcome: HOLD/FREEZE.
- `standing_validated`
 - Class: DECIDABLE for declared internal standing; ESCALATE_ONLY for external legal/professional standing.
 - Checker may decide: only inside declared internal standing rules.
 - Default unresolved outcome: HOLD/FREEZE.
- `authority_unambiguous`
 - Class: DECIDABLE, WINDOWED, or ESCALATE_ONLY depending on authority source.
 - Checker may decide: only when authority records are complete and challenge windows are closed.
 - Default unresolved outcome: HOLD/FREEZE.
- `no_hidden_successor_claim`
 - Class: WINDOWED or ESCALATE_ONLY.
 - Checker may decide: no absolute proof; only bounded no-claim-window status or ARL disposition.
 - Default unresolved outcome: HOLD/FREEZE.
- `no_prohibited_authority_laundering`
 - Class: ESCALATE_ONLY when contested.
 - Checker may decide: it may flag indicators, but must not clear contested cases.
 - Default unresolved outcome: HOLD/FREEZE.
- `successor_space_exhausted`
 - Class: WINDOWED or ESCALATE_ONLY.
 - Checker may decide: never by absence alone; a witnessed exhaustion record is required.
 - Default unresolved outcome: HOLD/FREEZE.
- `successor_claims_incompatible`
 - Class: DECIDABLE only for explicitly declared structural conflicts inside the bounded record; WINDOWED where notice/challenge windows may admit new compatibility evidence; ESCALATE_ONLY where legal, professional, anchor, authority, or external standing determines incompatibility.
 - Checker may decide: direct record contradiction may be flagged; contested substantive incompatibility must be held or escalated.
 - Default unresolved outcome: HOLD/FREEZE.
- `actor_or_role_departed`

- Class: DECIDABLE when an explicit departure, role-loss, resignation, revocation, or no-departure record exists inside the bounded transition record; WINDOWED when the departure is contested or subject to a notice / challenge / review window; ESCALATE_ONLY when the departure or role loss depends on external legal, professional, anchor, institutional, or competent-authority standing.
- Checker may decide: explicit internal departure/no-departure records may be classified; contested or external standing-dependent departure must be held or escalated.
- Default unresolved outcome: HOLD/FREEZE.

5.5 Bounded no-claim status

A checker must not claim that hidden successor claims do not exist in an absolute sense.

It may only report a bounded status such as:

```
no_claim_found_within_declared_record
notice_window_closed_without_claim
ARL_review_found_no_admissible_claim
competent_authority_disposition_attached
```

These statuses are not metaphysical absence claims. They are bounded governance statuses.

Invalid:

no hidden claim exists because the checker did not see one.

Valid:

no admissible claim was recorded in the declared evidence window, with notice scope, review scope, reviewer standing, and witness
 ↪ reference attached.

5.6 Lawful successor candidate

Before a successor is accepted as lawful, it is only a candidate:

```
SUCCESSOR_CANDIDATE = structurally plausible continuation under review.
LAWFUL_SUCCESSOR     = candidate with required decidable, windowed,
  ↪ escalate-only, and witness gates closed.
```

This prevents the implementation from treating a candidate list as an already lawful successor list.

5.7 Unique lawful successor

A unique lawful successor exists when exactly one admissible successor composition can continue the original composition line and all required predicate classes are closed.

Notation:

$\exists! K' \text{ valid_successor}(K, e)$

means:

there exists exactly one lawful successor K' after event e,
 after required DECIDABLE, WINDOWED, ESCALATE_ONLY, and witness gates
 ↪ are resolved.

5.8 Incompatible successor claims

Incompatible successor claims exist when two or more successor candidates each retain some admissible standing, memory, authority, obligation, or witness basis, but cannot all be true continuations of the same A6 composition.

Incompatibility is itself a classified predicate. It must not be hidden behind an opaque incompatible() function.

```
successor_claims_incompatible =
  direct_structural_conflict
  OR window_closed_with_unresolved_exclusive_claims
  OR ARL / human / competent authority disposition finding
  ↪ incompatibility
```

A direct structural conflict may be DECIDABLE when the record itself contains mutually exclusive continuation claims over the same composition line and the same authority slot.

A substantive incompatibility claim is WINDOWED when notice, challenge, or evidence windows may change the classification.

A contested incompatibility involving external legal standing, professional authority, anchor status, or institutional control is ESCALATE_ONLY.

Default rule:

```
Unresolved incompatibility -> HOLD/FREEZE.
Do not collapse unresolved incompatibility into EXIT.
Do not dramatize unresolved incompatibility into SPLIT.
```

SPLIT requires positive successor-claim evidence and resolved incompatibility status. It must not be inferred from emotion, style drift, or mere disagreement.

5.9 Successor-space exhaustion record

DISSOLVE requires a successor_space_exhaustion_record.

A valid record should include:

```
composition_id
transition_id
exhaustion_scope
notice_scope
challenge_window_start
challenge_window_end
standing_review_ref
obligation_review_ref
authority_review_ref
quorum_review_ref
witness_review_ref
successor_claims_considered
rejected_successor_claims
```

```

remaining_successor_claims
reviewer_or_authority_ref
decision_basis_ref
witness_ref

```

Absence of visible successor claims is not successor-space exhaustion.

If the exhaustion record is missing or incomplete, the outcome is HOLD/FREEZE, not DIS-SOLVE.

5.10 Temporal window state

WINDOWED predicates require explicit temporal-window state inside K.T or an equivalent transition record.

A window record should include:

```

window_id
predicate_name
opened_at
closes_at
notice_scope
challenge_route
status          # open / closed / expired / escalated / cancelled
closed_by
closure_basis_ref
witness_ref

```

The checker must evaluate the window state recorded in the artifact. It must not silently consult the current wall clock and reclassify a transition without a witnessed window-state update.

Valid pattern:

```

window_open record -> checker returns HOLD/FREEZE(reason="window_open")
witnessed window_close event -> K.T updated -> checker may re-evaluate
  ↪ from new record state

```

Invalid pattern:

```

same K and same event e
+ later wall-clock time
-> checker silently changes HOLD into EXIT/SPLIT/DISSOLVE

```

Compact rule:

```

Clock time is not a hidden checker input.
Window state must be in the record.

```

6. Transition operator

A6 transition classification is a partial witnessed operator:

```

 $\delta_{A6} : K \times \text{Event partial} \rightarrow \text{Decision}$ 

```

where:

```

Decision ∈ {
  CONTINUE,
  EXIT(actor, successor_K),
  SPLIT({successor_K1, successor_K2, ...}, cause),
  DISSOLVE(cause),
  HOLD_OR_FREEZE(reason)
}

```

The operator is partial.

If the transition cannot be classified without violating standing, witness, obligation, quorum, authority, or memory precedence, it must not silently proceed.

Compact rule:

When in doubt, HOLD.
 Do not guess EXIT.
 Do not dramatize SPLIT.
 Do not dissolve without successor analysis.

7. Core classification rule

7.1 Decision order

The classifier should evaluate transitions in this order:

1. Is the event material to A6 composition?
 If no: CONTINUE.
2. Are required records present for standing, obligation, quorum,
 ↪ authority, memory precedence, and witness?
 If no: HOLD_OR_FREEZE.
3. Classify required subpredicates as DECIDABLE, WINDOWED, or
 ↪ ESCALATE_ONLY.
 If any required WINDOWED / ESCALATE_ONLY predicate is unresolved:
 ↪ HOLD_OR_FREEZE.
4. Build successor candidates.
 Candidates are not yet lawful successors.
5. Promote candidates to lawful successors only after all required
 ↪ gates close.
6. Does exactly one lawful successor exist?
 If yes: EXIT or CONTINUE, depending on whether a participant/role
 ↪ actually departed.
7. Do multiple incompatible successor claims remain admissible?
 If yes: SPLIT.

8. Does a witnessed successor_space_exhaustion_record show no
 ↪ admissible successor remains?
 If yes: DISSOLVE.

9. Otherwise:
 HOLD_OR_FREEZE.

The order is fail-closed. Lack of proof for SPLIT does not imply DISSOLVE. Lack of proof for DISSOLVE does not imply EXIT. Unresolved classification remains HOLD.

7.2 Classification table

Condition	Decision
No material composition change	CONTINUE
Actor leaves and one lawful successor remains	EXIT
Evidence insufficient	HOLD_OR_FREEZE
Required WINDOWED predicate unresolved	HOLD_OR_FREEZE
Required ESCALATE_ONLY predicate unresolved	HOLD_OR_FREEZE
Standing unclear	HOLD_OR_FREEZE
Quorum unclear	HOLD_OR_FREEZE
Authority unclear	HOLD_OR_FREEZE
Multiple incompatible successor claims remain admissible	SPLIT
Successor-space exhaustion record is complete, witnessed, and no admissible successor remains	DISSOLVE
No successor visible but no exhaustion record	HOLD_OR_FREEZE

8. EXIT predicate

8.1 Definition

An actor departure may be classified as EXIT only if exactly one lawful successor composition exists.

EXIT(actor x) iff $\exists! K'$ such that:

```

valid_successor(K, K')
 $\wedge$  same_composition_lineage(K, K')
 $\wedge$  critical_invariants_hold(K')
 $\wedge$  decidable_predicates_pass(K, K')
 $\wedge$  windowed_predicates_closed_or_not_applicable(K, K')
 $\wedge$  escalate_only_predicates_resolved_or_not_applicable(K, K')
 $\wedge$  standing_rebound_or_closed( $x$ , K')
 $\wedge$  obligations_rebound_or_closed( $x$ , K')
 $\wedge$  quorum_satisfiable(K')
```

\wedge authority_unambiguous(K')

\wedge witness_appended(K \rightarrow K')

8.2 Required checks

A valid EXIT requires all of the following:

1. departing actor identified;
2. affected roles identified;
3. standing effects identified;
4. obligations closed, transferred, reduced, or externalized;
5. quorum remains satisfiable;
6. authority remains unambiguous;
7. witness lineage is appended;
8. DECIDABLE predicates pass;
9. WINDOWED predicates are closed or not applicable;
10. ESCALATE_ONLY predicates are resolved or not applicable;
11. no incompatible successor claim remains unresolved;
12. no critical invariant is silently dropped.

8.3 Invalid EXIT reductions

The following are invalid:

actor left \rightarrow EXIT
majority remained \rightarrow EXIT
system still runs \rightarrow EXIT
new leader declared \rightarrow EXIT
old documents still exist \rightarrow EXIT
one side has more resources \rightarrow EXIT
one side controls the files \rightarrow EXIT

A transition is EXIT only when one lawful successor composition is preserved.

8.4 EXIT examples

Ordinary role exit

A non-critical participant leaves. Their tasks are reassigned. No standing, quorum, authority, obligation, or witness path is broken.

Decision:

EXIT

Critical role exit with valid replacement

A critical signatory leaves, but a valid replacement is appointed under the existing exit law, standing is revalidated, obligations are rebound, and witness is appended.

Decision:

EXIT

Apparent exit without standing rebinding

A critical standing-holder leaves and the composition continues as if nothing happened.

Decision:

HOLD_OR_FREEZE

until standing is resolved.

9. SPLIT predicate

9.1 Definition

A transition must be classified as SPLIT when no unique lawful successor composition exists and more than one admissible but incompatible successor claim remains.

SPLIT(event e) iff:

$$\neg \exists! K' \text{ valid_successor}(K, e) \\ \wedge \text{residual_standing_or_obligation_exists}(K, e) \\ \wedge \text{incompatible_successor_claims_exist}(K, e)$$

9.2 Required checks

A valid SPLIT declaration requires:

1. original composition identified;
2. event or cause identified;
3. at least two successor claims identified;
4. successor claims are incompatible;
5. each successor claim has some admissible standing, memory, witness, obligation, or authority basis;
6. unique lawful continuation is unavailable;
7. witness event records the split classification;
8. re-entry is blocked unless later resolved by ARL / lawful review.

9.3 Invalid SPLIT reductions

The following are invalid:

important person left → SPLIT
 strong disagreement happened → SPLIT
 one member is angry → SPLIT
 composition became smaller → SPLIT
 style diverged → SPLIT
 one branch feels more authentic → SPLIT

SPLIT requires incompatible successor claims, not mood or drama.

9.4 SPLIT examples

Authority branch conflict

Two groups each claim authority to continue the same A6 composition. Both hold different portions of standing, witness, obligation, or memory basis. Their claims cannot both be the single continuation.

Decision:

SPLIT

Memory-lineage conflict

Two restored or continued branches each claim to be the same continuity path. Both have some witness basis, but the witness chains diverge and cannot be collapsed into one lawful successor.

Decision:

SPLIT

Resource controller vs standing holder

One branch controls infrastructure, another holds lawful standing. Neither alone can silently claim the whole continuation.

Decision:

HOLD_OR_FREEZE

then potentially:

SPLIT

if incompatible successor claims remain admissible after review.

10. HOLD / FREEZE predicate

10.1 Definition

HOLD or FREEZE is required when the system cannot safely classify a transition as CONTINUE, EXIT, SPLIT, or DISSOLVE.

HOLD_OR_FREEZE(reason) iff:

```
evidence_insufficient
v standing_unclear
v obligation_rebinding_unclear
v witness_lineage_unclear
v quorum_unclear
v authority_claim_unclear
v critical_invariant_status_unclear
```

10.2 HOLD is not continuation

HOLD is a classification delay, not a hidden continuation.

A composition in HOLD may preserve state, evidence, witness pointers, and review path. It must not proceed as if EXIT had been admitted.

10.3 FREEZE escalation

FREEZE is appropriate when unresolved classification touches:

- privilege;
- authority;
- continuity;
- memory core;
- witness lineage;
- irreversible obligation;
- external lawful standing;
- re-entry after a disputed branch.

10.4 HOLD examples

Missing witness

An actor departure appears ordinary, but no witness event exists and obligations are unclear.

Decision:

HOLD_OR_FREEZE

Unclear standing

A participant claims they can continue the composition, but their standing class is unverified.

Decision:

HOLD_OR_FREEZE

11. DISSOLVE predicate

11.1 Definition

A transition must be classified as DISSOLVE only when no admissible successor composition remains and this exhaustion has been explicitly recorded and witnessed.

DISSOLVE(event e) iff:

```

successor_space_exhaustion_record_exists(K, e)
∧ successor_space_exhaustion_record_witnessed(K, e)
∧ no_valid_successor(K, e)
∧ no_admissible_incompatible_successor_claims_remain

```

∧ obligations can only be closed, archived, externally resolved, or
 ↪ written off

DISSOLVE must not be inferred from silence, missing files, resource control, majority preference, emotional closure, or the absence of a visible claimant in the immediate record.

11.2 DISSOLVE default rule

No successor_space_exhaustion_record -> HOLD_OR_FREEZE.

Incomplete exhaustion record -> HOLD_OR_FREEZE.

Contested exhaustion record -> HOLD_OR_FREEZE or SPLIT, depending on
 ↪ claims.

Complete witnessed exhaustion record -> DISSOLVE candidate.

A deterministic checker may propose DISSOLVE_CANDIDATE only when the required record exists. It must not finalize DISSOLVE by itself when external standing, legal authority, anchor state, or institutional responsibility is involved.

11.3 Required checks

A valid DISSOLVE declaration requires:

1. original composition identified;
2. material transition event identified;
3. successor search scope declared;
4. notice or challenge window declared where applicable;
5. all successor claims considered;
6. all rejected successor claims recorded with reason and basis;
7. no remaining admissible successor claim;
8. obligation closure / externalization path declared;
9. authority and standing exhaustion recorded;
10. witness event appended;
11. review owner identified;
12. re-entry rules stated for future evidence.

11.4 DISSOLVE is not SPLIT

SPLIT preserves competing successor claims.

DISSOLVE ends the composition as a continuing composition.

The remaining work may include:

archive;
 liability closure;
 external adjudication;
 witness preservation;
 memory sealing;
 public non-claim;
 re-entry denial;
 future evidence re-entry path.

11.5 DISSOLVE examples

No lawful successor after witnessed exhaustion

A composition loses the only standing path, notice/review has run, no valid successor exists, no branch retains admissible continuation standing, and a witnessed exhaustion record is attached.

Decision:

DISSOLVE

Archive-only residue

Documents and memories remain, but no authority, standing, quorum, or witness path remains capable of continuation. Exhaustion is recorded and witnessed.

Decision:

DISSOLVE

with archive preservation.

No visible successor, but no exhaustion record

No successor appears in the immediate file set, but no review window or exhaustion record exists.

Decision:

HOLD_OR_FREEZE

not DISSOLVE.

12. CONTINUE predicate

12.1 Definition

CONTINUE applies when an event does not materially affect A6 composition or when the transition is already covered by ordinary operation and no participant departure, standing disturbance, obligation rebinding, quorum disturbance, authority change, or witness-lineage change occurs.

CONTINUE(event e) iff:

```
no_material_composition_change(e)
 $\wedge$  no_successor_claim_change(e)
 $\wedge$  no_critical_invariant_change(e)
```

12.2 CONTINUE must not hide EXIT

If an actor or critical role actually leaves, the event must not be labelled CONTINUE merely because the system still functions.

13. Critical invariants

A6 composition transition classification must check at least these invariants:

Invariant	EXIT condition	SPLIT / HOLD trigger
Role structure	roles are closed or reassigned	role cannot be lawfully reassigned
Standing	standing is preserved, closed, or transferred	standing becomes disputed
Obligation	obligations are closed or rebound	obligations split or remain unresolved
Quorum	quorum remains satisfiable	quorum impossible or disputed
Authority Witness	one authority path remains lineage appended	authority claims branch witness chain broken or forked
Memory precedence	one memory line remains controlling	memory branches compete
Re-entry Purpose	release basis is clear composition purpose remains compatible	re-entry legality unclear purpose splits into incompatible goals
Successor-space exhaustion	exhaustion record not required for EXIT	DISSOLVE forbidden without witnessed exhaustion record

If any critical invariant cannot be checked, classification must enter HOLD or FREEZE.

14. Witness binding

14.1 Required witness events

The canonical witness namespace for this addendum is a6.*.

Canonical A6 event types:

```
a6.composition_transition_classified
a6.composition_exit_admitted
a6.composition_split_declared
a6.composition_transition_hold
a6.composition_dissolve_candidate
a6.composition_successor_space_exhausted
a6.composition_dissolved
a6.composition_reentry_released
a6.composition_reentry_denied
```

ARL-facing names are projections, not aliases. They may be generated from canonical A6 events for ARL integration, but they must not be independently authored as a second root of truth.

Projection map:

```

a6.composition_transition_classified      ->
  ↪ arl.composition_transition_classified
a6.composition_exit_admitted             ->
  ↪ arl.composition_exit_admitted
a6.composition_split_declared            ->
  ↪ arl.composition_split_declared
a6.composition_transition_hold            ->
  ↪ arl.composition_transition_hold
a6.composition_dissolve_candidate         ->
  ↪ arl.composition_dissolve_candidate
a6.composition_successor_space_exhausted ->
  ↪ arl.composition_successor_space_exhausted
a6.composition_dissolved                  -> arl.composition_dissolved
a6.composition_reentry_released           ->
  ↪ arl.composition_reentry_released
a6.composition_reentry_denied             ->
  ↪ arl.composition_reentry_denied

```

Required namespace rule:

```

canonical_event_type = a6.*
projection_event_type = arl.* only when generated from
  ↪ canonical_event_type

```

The projection must preserve `transition_id`, `composition_id`, `witness_ref`, canonicalization, and hash/signature binding. A projection must not create a new witness event identity.

Invalid:

```

a6.* authored by one component
arl.* authored separately by another component
both treated as equivalent witness roots

```

Valid:

```

a6.* canonical event
-> deterministic arl.* projection with preserved witness binding

```

14.2 Minimum event fields

Each transition event should be able to carry:

```

record_id
transition_id
ts
system_id
composition_id
parent_a6_ref
parent_doi_ref
actor_role
event_type
canonical_event_type
projection_event_type
state_from

```

state_to
 decision
 reason_code
 policy_ref
 input_ref
 output_ref
 previous_state_ref
 successor_claims
 unique_successor_claim
 successor_space_exhaustion_record
 predicate_class_status
 successor_claim_incompatibility_status
 temporal_window_state
 critical_invariant_status
 standing_status
 obligation_status
 quorum_status
 authority_status
 memory_precedence_status
 witness_ref
 canonicalization
 hash_or_signature_binding
 privacy_class
 labels

14.3 Witness anti-laundering rule

Witness does not make a transition lawful after the fact.

Bad pattern:

unclassified departure
 → ordinary continuation
 → later poetic witness note
 → claimed EXIT

Good pattern:

departure event
 → standing / obligation / quorum / authority / witness check
 → classification
 → witness event
 → bounded continuation, hold, split, or dissolution

Witness is trace, not makeup.

15. Semantic checker rule names

A future local checker may implement these rule names:

A6CTP-R01_exit_requires_unique_successor
 A6CTP-R02_exit_requires_standing_rebound_or_closure

A6CTP-R03_exit_requires_obligation_rebound_or_closure
 A6CTP-R04_exit_requires_quorum_satisfiable
 A6CTP-R05_exit_requires_authority_unambiguous
 A6CTP-R06_exit_requires_witness_append
 A6CTP-R07_split_requires_incompatible_successor_claims
 A6CTP-R08_split_requires_no_unique_successor
 A6CTP-R09_hold_required_when_evidence_insufficient
 A6CTP-R10_hold_required_when_standing_unclear
 A6CTP-R11_dissolve_requires_successor_space_exhaustion_record
 A6CTP-R12_dissolve_requires_no_valid_successor_after_exhaustion
 A6CTP-R13_continue_forbidden_when_material_actor_departure_exists
 A6CTP-R14_no_exit_without_critical_invariant_status
 A6CTP-R15_no_transition_label_without_policy_ref
 A6CTP-
 ↪ R16_predicates_must_be_classified_decidable_windowed_or_escalate_only
 A6CTP-R17_checker_must_not_clear_escalate_only_predicate
 A6CTP-R18_no_hidden_claim_absence_must_be_windowed_or_escalated
 A6CTP-R19_dissolve_forbidden_without_witnessed_exhaustion
 A6CTP-R20_lawful_successor_must_not_be_opaque_judgment
 A6CTP-R21_successor_claims_incompatible_must_be_predicate_classified
 A6CTP-R22_unresolved_incompatibility_requires_hold
 A6CTP-R23_witness_event_namespace_requires_canonical_a6_root
 A6CTP-R24_windowed_predicate_requires_recorded_temporal_state
 A6CTP-R25_actor_or_role_departed_must_be_predicate_classified

15.1 Predicate status vocabulary

Recommended predicate statuses:

pass
 fail
 unknown
 not_applicable
 window_open
 window_closed_no_claim
 incompatibility_unresolved
 incompatibility_resolved
 escalation_required
 escalation_resolved
 witness_missing
 departure_record_explicit
 departure_contested
 no_departure_record_explicit

15.2 Checker result vocabulary

Recommended checker outcomes:

pass
 fail
 hold_required
 freeze_required
 insufficient_evidence

escalation_required
 dissolve_candidate_requires_review
 not_applicable

15.3 Checker non-goals

The checker must not decide actual law, moral blame, personhood, metaphysical identity, or absolute absence of hidden claims.

It checks whether the transition record satisfies A6-CTP structural requirements and whether unresolved predicates require HOLD, FREEZE, ARL review, or human/competent authority disposition.

16. Fixture index seed

A future fixture pack may include:

valid_exit_role_reassigned.json
 valid_exit_obligations_closed.json
 valid_exit_critical_role_replaced_with_witness.json
 valid_hold_unclear_standing.json
 valid_hold_missing_witness.json
 valid_split_two_authority_claims.json
 valid_split_memory_lineage_conflict.json
 valid_dissolve_no_successor_with_exhaustion_record.json
 valid_hold_no_visible_successor_without_exhaustion_record.json
 valid_hold_escalate_only_predicate_unresolved.json
 valid_exit_all_predicate_classes_closed.json
 valid_hold_incompatible_claims_unresolved.json
 valid_split_incompatible_claims_window_closed.json
 valid_window_open_returns_hold.json
 valid_exit_explicit_actor_departure_record.json
 valid_continue_explicit_no_actor_departure_record.json
 valid_hold_contested_actor_departure.json
 valid_arl_projection_generated_from_a6_canonical_event.json
 invalid_exit_without_unique_successor.json
 invalid_exit_without_obligation_rebinding.json
 invalid_exit_without_witness.json
 invalid_split_without_incompatible_claims.json
 invalid_continue_with_material_actor_departure.json
 invalid_dissolve_without_successor_space_exhaustion_record.json
 invalid_dissolve_with_admissible_successor_claim.json
 invalid_checker_clears_escalate_only_predicate.json
 invalid_split_without_incompatibility_predicate_class.json
 invalid_alias_fork_witness_event_name.json
 invalid_windowed_predicate_uses_wall_clock_without_record.json
 invalid_exit_with_unclassified_actor_departure.json
 invalid_continue_with_contested_actor_departure.json

Each fixture should include:

composition_id
 event

```

participants_before
participants_after
roles_before
roles_after
standing_status
obligation_status
quorum_status
authority_status
witness_status
successor_claims
predicate_class_status
successor_claim_incompatibility_status
temporal_window_state
canonical_event_type
projection_event_type
successor_space_exhaustion_record
expected_decision
expected_rule_results

```

17. Implementation mapping note

This addendum is normative / control-layer guidance.

It does not require immediate runtime implementation.

If implemented later, the first implementation surface should be deterministic and narrow:

modules/arbitration_review/composition_transition.py

Suggested functions:

```

classify_composition_transition(record) -> transition_classification
classify_predicates(record) -> predicate_class_status
build_successor_candidates(record) -> successor_candidates
classify_successor_claim_incompatibility(record) ->
  ↪ incompatibility_status
classify_temporal_window_state(record) -> window_status
canonicalize_witness_event_type(record) -> canonical_event_record
validate_exit(record) -> problems
validate_split(record) -> problems
validate_hold(record) -> problems
validate_dissolve_candidate(record) -> problems
validate_continue(record) -> problems

```

The implementation must distinguish:

```

successor_candidate
lawful_successor
admissible_successor_claim
incompatible_successor_claim
successor_space_exhaustion_record

```

Implementation constraints:

```
no LLM required;
no network required;
no UI-first implementation;
no smart Judge first;
no federation-wide consensus first;
no runtime mutation without witness;
no memory-core write by checker;
no checker clearance of ESCALATE_ONLY predicates;
no SPLIT from unresolved successor_claims_incompatible;
no DISSOLVE without witnessed successor_space_exhaustion_record;
no independent arl.* witness root when a6.* canonical event is required;
no wall-clock reclassification of WINDOWED predicates without recorded
  ↪ window-state update;
fail closed on unclear status.
```

17.1 Machine/human boundary

A deterministic checker may say:

```
this decidable field is present or missing;
this quorum is satisfiable under declared roster and rules;
this witness reference exists;
this required predicate is unresolved;
this transition requires HOLD/FREEZE;
this DISSOLVE candidate lacks exhaustion record.
```

A deterministic checker must not say:

```
no hidden claim exists;
no authority laundering exists in contested facts;
all legal standing is exhausted;
DISSOLVE is finally authorized;
this contested successor is lawful by machine judgment alone.
```

Those are ARL / human anchor / competent review surfaces, depending on the transition class.

17.2 WINDOWED determinism rule

A checker remains deterministic only over the supplied record state.

For WINDOWED predicates, the terminal checker output may be:

```
window_open -> HOLD/FREEZE
window_closed_with_witness -> evaluate from updated K.T
window_escalated -> HOLD/FREEZE or ESCALATE_ONLY disposition required
```

The checker must not use unstated wall-clock time as a hidden input.

If the wall clock has advanced, a separate witnessed window-state update must be appended before reclassification.

17.3 Witness namespace implementation rule

The implementation should store `a6.*` as the canonical event type.

If ARL integration requires `arl.*`, the implementation should generate it as a projection field, not author it as a separate event.

18. Relationship to ARL

A6-CTP should be treated as a classification predicate upstream of ARL hold / freeze / re-view / outcome / re-entry paths.

Recommended placement:

```
standing-validation
→ predicate-classification DECIDABLE/WINDOWED/ESCALATE_ONLY
→ composition-transition-classification
→ hold-entry / freeze-entry
→ evidence-admissibility
→ review-state persistence
→ outcome witness
→ re-entry legality
```

This does not replace ARL.

It gives ARL a sharper reason code when the dispute concerns composition continuation.

19. Relationship to $c = a + b$

For $c = a + b$, the same rule prevents a common false continuation:

```
a absent or changed
+ b continues to run
= ordinary c continuity
```

This is invalid.

The correct classification depends on whether one lawful continuity-bearing successor remains under anchor, memory, witness, authority, L4, and permission constraints.

Examples:

```
Temporary anchor unavailability with preserved reduced-authority
↔ posture → HOLD / reduced-authority continuity.
Anchor standing loss with no lawful successor → DISSOLVE or
↔ archive-only posture.
Two branches claiming same c lineage → SPLIT / fork dispute.
Model or hardware replacement with witnessed continuity bundle →
↔ possible EXIT-like replacement or CONTINUE, depending on scope.
```

The sign + is governed binding, not arithmetic addition.

20. Public claim discipline

This addendum supports claim hygiene.

Invalid public claims:

A6 can always survive actor loss.
 A6 proves identity continuity.
 A6 split is subjective feeling.
 A6 exit is whatever the majority says.
 Witness record alone proves lawful transition.
 Running software proves continuation.
 No visible claimant proves DISSOLVE.
 A checker can clear hidden-claim absence absolutely.

Valid bounded claims:

A6-CTP defines a transition classification predicate.
 A6-CTP separates EXIT, SPLIT, HOLD, and DISSOLVE.
 A6-CTP treats unique lawful successor preservation as the EXIT
 ↪ threshold.
 A6-CTP requires HOLD/FREEZE when standing, obligation, quorum,
 ↪ authority, or witness are unclear.
 A6-CTP does not replace the parent DOI artifact.
 A6-CTP does not claim that all successor predicates are
 ↪ machine-decidable.
 A6-CTP requires DISSOLVE to be backed by witnessed successor-space
 ↪ exhaustion.
 A6-CTP requires actor_or_role_departed to be an explicit classified
 ↪ predicate, not an implicit branch assumption.

21. Red-line failures

A transition record is non-compliant if it:

1. labels EXIT without unique lawful successor analysis;
2. labels SPLIT without incompatible successor claims;
3. labels CONTINUE despite material actor or role departure;
4. dissolves a composition while admissible successor claims remain; 4a. dissolves a composition without a witnessed successor_space_exhaustion_record;
5. ignores standing loss;
6. ignores obligation fracture;
7. ignores quorum impossibility;
8. ignores authority ambiguity;
9. ignores witness-lineage break;
10. uses majority, control of files, or resource ownership as sufficient continuation proof;
11. proceeds under uncertainty without HOLD/FREEZE;
12. emits witness only after silent continuation has already occurred;
13. hides ESCALATE_ONLY judgment inside a function named like a deterministic checker;
14. treats no visible claim as proof of no admissible successor;

15. labels EXIT or CONTINUE from a single lawful successor while actor_or_role_departed remains unclassified or contested.

22. Compact operational rule

Given composition K and event e:

```
if not material_to_composition(e):
    return CONTINUE
```

```
predicate_status = classify_predicates(K, e)
```

```
if predicate_status.missing_required_records:
    return HOLD_OR_FREEZE(reason="missing_required_records")
```

```
if predicate_status.required_decidable_failed:
    return
↪ HOLD_OR_FREEZE(reason="decidable_predicate_failed_or_unclear")
```

```
if predicate_status.windowed_open:
    return HOLD_OR_FREEZE(reason="windowed_predicate_unresolved")
```

```
if predicate_status.escalate_only_unresolved:
    return HOLD_OR_FREEZE(reason="escalate_only_predicate_unresolved")
```

```
candidates = build_successor_candidates(K, e)
```

```
lawful = [
    candidate for candidate in candidates
    if candidate.all_required_predicates_closed
    and candidate.witness_appended
]
```

```
incompatibility_status = classify_successor_claim_incompatibility(K, e,
↪ lawful, candidates)
```

```
departure_status = classify_actor_or_role_departure(K, e)
```

```
if departure_status.window_open:
    return HOLD_OR_FREEZE(reason="actor_or_role_departure_window_open")
```

```
if departure_status.escalate_only_unresolved:
    return
↪ HOLD_OR_FREEZE(reason="actor_or_role_departure_escalate_only_unresolved")
```

```
if departure_status.decidable_unclear:
    return HOLD_OR_FREEZE(reason="actor_or_role_departure_unclear")
```

```
if incompatibility_status.window_open:
    return
↪ HOLD_OR_FREEZE(reason="successor_incompatibility_window_open")
```

```

if incompatibility_status.escalate_only_unresolved:
    return
    ↪ HOLD_OR_FREEZE(reason="successor_incompatibility_escalate_only_unresolved")

if len(lawful) == 1:
    if departure_status.departed:
        return EXIT(actor, lawful[0])
    if departure_status.no_departure:
        return CONTINUE
    return
    ↪ HOLD_OR_FREEZE(reason="actor_or_role_departure_status_unclassified")

if len(lawful) > 1:
    if incompatibility_status.resolved_incompatible:
        return SPLIT(lawful, cause=e)
    return
    ↪ HOLD_OR_FREEZE(reason="multiple_lawful_successors_incompatibility_unresolved")

if admissible_successor_claims_exist(K, e):
    if incompatibility_status.resolved_incompatible:
        return SPLIT(successor_claims, cause=e)
    return
    ↪ HOLD_OR_FREEZE(reason="successor_claim_incompatibility_unresolved")

if successor_space_exhaustion_record_complete_and_witnessed(K, e):
    if len(lawful) == 0 and no_remaining_admissible_successor_claims(K,
    ↪ e):
        return DISSOLVE(cause=e)

if no_visible_successor_claims(K, e):
    return
    ↪ HOLD_OR_FREEZE(reason="no_visible_successor_without_exhaustion_record")

return HOLD_OR_FREEZE(reason="classification_unresolved")

```

22.1 Non-machine-decidability guard

lawful_successors(K,e)

must not hide ARL, human, legal, or authority judgment behind a deterministic function name.

A checker may assemble evidence and classify predicate status. It may not silently convert unresolved WINDOWED or ESCALATE_ONLY predicates into pass.

Compact guard:

Machine-checkable fields are not the whole successor question.

The machine may detect unresolved authority.

It must not launder unresolved authority into EXIT or DISSOLVE.

22.2 DISSOLVE guard

No exhaustion record -> no DISSOLVE.

No witness on exhaustion -> no DISSOLVE.

Contested exhaustion -> HOLD/FREEZE or SPLIT.

22.3 SPLIT incompatibility guard

No classified incompatibility status -> no SPLIT.

Unresolved incompatibility -> HOLD/FREEZE.

Machine-visible disagreement alone -> no SPLIT.

The SPLIT branch must not hide the same kind of judgment that `lawful_successors(K, e)` is forbidden to hide.

A checker may flag direct structural contradiction. It must not decide contested legal, professional, anchor, authority, or institutional incompatibility by machine judgment alone.

22.4 WINDOWED time guard

No window record -> no window closure.

No witnessed window-state update -> no reclassification.

Same K and same e -> same checker output.

If time has passed, the transition record must show that time as a witnessed update in `K.T` or equivalent state. Otherwise the checker must keep returning `HOLD/FREEZE (reason="window_open")`.

22.5 Actor / role departure guard

No departure classification -> no EXIT/CONTINUE discriminator.

Contested departure -> HOLD/FREEZE.

External standing-dependent departure -> ESCALATE_ONLY.

Explicit no-departure record -> CONTINUE may be considered, if all

↪ other predicates permit it.

The `actor_or_role_departed` branch is a classified predicate. It must not be treated as an informal observation merely because it usually appears simple.

A checker may classify an explicit internal departure/no-departure record. It must not decide contested resignation, revocation, anchor standing, professional standing, legal removal, or institutional role loss by machine judgment alone.

23. Changelog

v0.1.4 — 2026-07-06

Third-pass convergent review-integrated draft.

Integrated b-layer review record:

Record ID: A6CTP-REV-20260706-b3

Reviewed target: A6-CTP v0.1.3

Reviewed target SHA-256:

↪ 3d62a70a1770a57712c28d9091197795e60c3a8a838870ea9bd18121880e55d6

Companion PDF SHA-256:

↪ 330971fbd6b2bd149880121f37ec38695a52f9dc39d813ee2d67aaaba6ce5ff8

Disposition: T1 accepted as leaf-completeness hardening input;

↪ predicate-review cycle marked convergent at draft level

Changes:

- added actor_or_role_departed to the predicate class map with DECIDABLE / WINDOWED / ESCALATE_ONLY handling;
- revised the compact operational rule so EXIT/CONTINUE under a single lawful successor requires classified departure status;
- added an actor / role departure guard to prevent implicit EXIT/CONTINUE discrimination;
- added semantic rule A6CTP-R25 for actor / role departure predicate classification;
- added fixture seeds for explicit departure, explicit no-departure, contested departure, and unclassified-departure failure;
- added a predicate-review convergence note: after T1, remaining work is expected to concern fixtures / implementation / conformance evidence rather than predicate-completeness repair.

No runtime implementation claim.

No conformance claim.

No parent DOI artifact modification.

v0.1.3 — 2026-07-06

Second-pass review-integrated draft.

Integrated b-layer review record:

Record ID: A6CTP-REV-20260706-b2

Reviewed target: A6-CTP v0.1.2

Reviewed target SHA-256:

↪ e5aa43d591cdfd7d24b58ac40fe9a6198f1eb13b36c9bd0cb2215b75ffd73d2e

Companion PDF SHA-256:

↪ acf1ab70049ab77a83428f66f029a2fd316bb5f37ce0de28c9b9bde4ab0cdfbb

Disposition: N1-N3 accepted as bounded hardening input

Changes:

- replaced a6.* / arl.* alias ambiguity with canonical a6.* witness namespace and generated arl.* projection map;
- added successor_claims_incompatible to predicate classification, with DECIDABLE / WINDOWED / ESCALATE_ONLY handling;
- revised SPLIT logic so unresolved incompatibility defaults to HOLD/FREEZE rather than EXIT or SPLIT;
- added temporal window state T to the transition-analysis projection K;
- added temporal-window records and no-wall-clock checker rule for WINDOWED predicates;
- added semantic rules and fixture seeds for incompatibility classification, witness namespace, and window-state discipline.

No runtime implementation claim.

No conformance claim.

No parent DOI artifact modification.

v0.1.2 — 2026-07-06

Review-integrated draft.

Integrated b-layer review record:

Record ID: A6CTP-REV-20260706-b

Reviewed target: A6-CTP v0.1.1

Reviewed target SHA-256:

↪ be1a6a750b770cef7d00e149ffec5db8c8a2d070dce8c49a8f57a3d7256d9b14

Disposition: findings accepted as draft-hardening input

Changes:

- split `lawful_successors(K,e)` into `DECIDABLE`, `WINDOWED`, and `ESCALATE_ONLY` predicate classes;
- added bounded no-claim status to prevent absolute hidden-claim absence claims;
- changed `DISSOLVE` from simple no-successor framing to witnessed successor-space exhaustion;
- added `successor_space_exhaustion_record` definition and event/rule hooks;
- revised compact operational rule so unresolved predicate classes default to `HOLD/FREEZE`;
- added section-level claim-strength map;
- added checker constraints preventing `ESCALATE_ONLY` clearance by deterministic checker;
- added fixtures and semantic rules for the new boundary.

No runtime implementation claim.

No conformance claim.

No parent DOI artifact modification.

v0.1.1 — 2026-07-06

Initial bounded addendum draft.

Added:

- explicit A6 composition transition operator;
- `EXIT / SPLIT / HOLD / DISSOLVE` predicates;
- unique lawful successor threshold;
- critical invariant table;
- witness event recommendations;
- semantic checker rule names;
- fixture index seed;
- precedence and non-replacement status for DOI-published parent A6 artifact.

No runtime implementation claim.

No conformance claim.

No parent DOI artifact modification.

24. Closing invariant

EXIT is not departure.

EXIT is departure with one lawful continuation.

SPLIT is not disagreement.

SPLIT is loss of one lawful continuation under incompatible successor
↔ claims.

HOLD is not weakness.

HOLD is refusal to launder uncertainty into action.

DISSOLVE is not failure to continue emotionally.

DISSOLVE is witnessed successor-space exhaustion.

A checker is not a court.

A missing claimant is not exhaustion.

A hidden judgment inside a function name is laundering.

An unresolved incompatibility is HOLD, not EXIT or SPLIT.

An alias is not witness canon.

A clocknot in the record is not a checker input.

A6 composition survives only when its continuation is lawful, witnessed, and non-ambiguous enough to remain one composition.