Realizing Possible Outcomes through Feasible Mechanism Paths in Generative Relational Models

Wanhong HUANG
huangwanhong@serendip.ngo

Abstract

A possible outcome can have several mechanism paths while depending on one shared condition. This paper develops a model of target availability through feasible paths in the generative relational framework. Feasibility is checked at every execution prefix before words are compressed to semigroup products. Persistent positive requirements yield finite minimal requirement families, while dependency cuts identify condition removals that disable every target route. An actor-control map separates condition multiplicity from independent control. Resource ledgers derive the minimum initial stock required by consumption and production at successive steps, and joint profiles preserve incomparable combinations of permissions and resources. Additive restoration is characterized by minimal deficit profiles. Under uncertain scenarios, the analysis distinguishes scenario-wise availability from a common route that can be selected with the available information. A finite deterministic information-state recursion characterizes adaptive feasible termination and distinguishes free observation refinement from costly diagnostic actions. Horizon comparisons separate remaining distance to a target from unrestricted reachability. For finite permission vocabularies and integer resources, an application of Dickson's lemma establishes finite availability bases for unbounded route families; continuous-resource counterexamples show the role of the domain assumptions. Probabilistic calculations distinguish shared-condition dependence, route selection, and realization; finite counterexamples show why more routes can coexist with lower success probability under a changed selection rule. A complete route illustration connects the constructions. The philosophical discussion examines possibility, access, agency, retained competence, dependence, and evaluation. The results are conditional on declared trajectories, requirements, information, and intervention assumptions. Empirical interpretation and the evaluation of outcomes require additional evidence and standards.

Keywords: mechanism support; feasible paths; enabling conditions; shared dependencies; realization

Note on Scope and Research Status

This working paper develops one branch of a generative relational research program. Its purpose is to make a bounded representation precise enough for criticism and further application. Ontological questions remain suspended. The terms used in the model specify analytical roles and declared interpretations.

Definitions and modeling assumptions precede the conditional propositions that use them. The finite example illustrates their consequences under stipulated inputs. A mathematical witness and an empirically warranted claim have different requirements. The source audits and project notes record the distinction between earlier repository ideas, agent-added development, and verified literature positions.

The paper uses the repository's article template and retains its author, notice, and draft-marking structure. It is an expanded draft for author review. The review should examine the argument's scope, the suitability of the examples, the adequacy of the source comparisons, and the questions that an empirical application would need to settle.

DRAFT

Responsible Use and Rights Reservation

This section separates requested scholarly conduct from the legal permissions stated on the following page. It records an ethical request for responsible use and then defines the narrower scope of retained legal rights.

The author encourages good-faith discussion, criticism, independent inquiry, and responsible use of the material in this work. Separately from the licence's terms, the author asks users to consider foreseeable harms when adapting or applying the proposed framework. This ethical request leaves the licence's permissions and legally authorized uses unchanged.

The author retains the rights preserved under CC BY-NC 4.0 and may pursue remedies to which the author is legally entitled for breach of the licence or violation of the author's independently applicable rights. Reuse remains independent from authorial endorsement. Third-party rights require authorization from their respective holders where applicable. Copyright exceptions and limitations, including applicable forms of fair use or fair dealing, remain fully available.

DRAFT

Notices

This page records the draft's status, inherited licence, and preparation disclosure.

Status. This comprehensive revision is in progress for author review. Its formal propositions are conditional on the displayed assumptions. The path interpretation and shared-permission example remain revisable modeling proposals. Empirical validation, a broader originality review, and publication sequencing remain open.

Licence. Except where otherwise indicated, copyright 2026 Wanhong Huang. This work is made available under the Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0). Subject to its terms, the licence permits sharing and adaptation for noncommercial purposes with appropriate attribution, a link to the licence, an indication of changes, and attribution that preserves the licensor's independence from the reuse. Reuse is governed solely by that licence; the responsible-use request on the preceding page remains separate from its terms. The licence deed and legal-code link are available at `https://creativecommons.org/licenses/by-nc/4.0/`. The licence governs in case of conflict with this summary. Third-party material remains subject to the rights held by its respective rights holders.

Statement on the use of language models. The author selected the research program, required ontological suspension, and authorized drafting of this portfolio paper. Language-model assistance supported organization of the source discussion, mathematical reconstruction, counterexamples, website verification, and preparation in LaTeX. Agent-added constructions and source provenance are recorded in the accompanying research files. This version awaits the author's review of its formulations, arguments, and references. Authorship and decisions about circulation remain with the human author.

DRAFT

Contents

1 Introduction 6

2 Mechanism Support and Feasible Paths 7

3 Compression of Paths into Mechanism Products 10

4 Persistent Enabling Conditions 14

5 Minimal Requirement Families and Availability 17

6 Condition Removal and Target Robustness 20

7 Control over Enabling Conditions 23

8 A Shared-Permission Route Illustration 25

9 Resources and Prefix Requirements 29

10 Route Restoration and Feasibility under Uncertainty 32

11 Horizons and Progress Toward a Target 36

12 Unbounded Route Families and Finite Availability Bases 39

13 Scenario Probability and Outcome Realization 43

14 Information and the Selection of Feasible Paths 47

15 Adaptive Feasible Termination under Partial Observation 50

16 Philosophical Interpretation of Feasible Possibility 54

17 Evidence, Objections, and Research Obligations 58

18 Conclusion 60

DRAFT

1 Introduction

This paper examines the passage from possible outcomes to their realization through feasible mechanism paths in a generative relational model. It develops the information needed to distinguish outcome support, executable sequences, persistent prerequisites, resource use, control, informed choice, and actual execution. The method combines conditional derivations, finite counterexamples, a complete route illustration, and philosophical analysis of their scope. Ontological questions remain suspended.

A target can have several represented routes while depending on one shared permission. An additional route can improve resistance to withdrawal without adding a new outcome. A path can have a nonnegative final resource balance and still fail before its final step. An actor can face a successful route in every possible scenario while lacking the information needed to choose one reliably. These differences concern distinct conditions of realization.

The generative relational framework supplies an action model in which mechanisms compose and configurations are compared under a preserved equivalence [2]. The present paper develops a bounded extension of that framework. It retains mechanism witnesses and then examines what a particular sequence requires at each intermediate state. The resulting path record supports questions that an outcome image or a composite product can leave unresolved.

The first result characterizes product compression. A word's feasibility can be recovered from its product exactly when feasibility is constant on each evaluation fibre. A two-state prefix trap disproves automatic descent. The paper also preserves a legitimate weaker interpretation: a product can be existentially available because some feasible word realizes it, even when another factorization fails. That distinction prevents the critique from rejecting composite models more broadly than the proof warrants.

The second result concerns persistent positive prerequisites. Under fixed transitions and scenario-independent requirements, a path is feasible precisely when the union of its step requirements is available. Target words then yield a requirement family, whose minimal members preserve the entire Boolean availability function. Disabling the target is exactly a hitting-set condition on the currently feasible requirements.

These results have established mathematical antecedents in discrete planning, provenance representation, and reliability analysis [3, 1, 6]. Their use here is scoped to the coordination of path execution and shared enabling conditions. The manuscript derives the relevant results explicitly and states where social interpretation requires additional premises.

Three extensions address omissions that a persistent requirement set cannot resolve. An actor-control map distinguishes the number of removed conditions from the number of actors able to remove them. A prefix resource ledger derives the minimum initial stock for a fixed word, including production timing. An information model distinguishes scenario-specific successful choices from a policy implementable under the actor's observations.

The probability analysis separates availability from realization. A joint scenario distribution weights the availability function. A selection law assigns probability to feasible words, producing an endpoint probability bounded above by availability. Shared prerequisites make route events dependent even when the underlying conditions are independent. The exact equality condition in the bound also requires a selection rule that the actor can implement.

The worked example compares two facility routes sharing one permission, a duplicate operation label, and an additional direct route. The catalogs share the same four-state outcome set at full availability. Their target-word counts and removal robustness differ. Every transition, requirement, and minimal cut is derived, and control and resource variants are stated as separate added assumptions.

The philosophical discussion examines retained capacity, effective access, meaningful alternatives, control over continuation, and the formation of interests through participation. It keeps generative and reproductive roles open to evaluation. Word counts, cut cardinalities, resource thresholds, and probabilities answer different questions; a universal ranking would require further standards and an argument for combining them.

Section 2 defines support and feasible paths. Section 3 examines product reduction. Sections 4 and

DRAFT

5 derive persistent requirements and minimal availability. Sections 6 and 7 address condition removal and actor control. The finite example is developed in Section 8. Sections 9, 10, and 11 examine resource thresholds, additions that restore availability, common-route feasibility under uncertainty, and target progress. Section 12 derives finite availability bases for unbounded route families under discrete resource assumptions. Sections 13 and 14 treat probability and informed selection. Section 15 completes the finite deterministic adaptive-policy analysis with an information-state recursion, a full induction proof, and diagnostic counterexamples. The final sections discuss philosophical interpretation, evidence, objections, and a bounded research program.

2 Mechanism Support and Feasible Paths

This section establishes the objects needed to distinguish a supported outcome from an executable route to that outcome. It begins with mechanism fibres and their outcome image, derives the effect of enlarging a mechanism repertoire, and then introduces primitive words with prefix feasibility. It also examines partial composition and the compatibility of availability with a quotient. These constructions provide the foundation for the later analysis of requirements, control, resources, and selection.

2.1 Configurations, Mechanisms, and Outcome Classes

The base model distinguishes a configuration from an outcome class used by an inquiry. This subsection states the action assumptions and the interpretive responsibility of the quotient.

Let a semigroup M act on a nonempty set S, with $(nm)x = n(mx)$. Let $\sim$ be an equivalence preserved by every mechanism, and write $q : S \to Q = S/\!\sim$ for the quotient map. The quotient identifies outcomes according to the selected distinction regime.

Preservation ensures that $q(mx)$ depends only on m and $q(x)$ when the mechanism is fixed. If $x \sim y$, then $mx \sim my$, so the induced map on classes is well-defined. A coarse outcome description can therefore support dynamics only when it retains the distinctions needed by the declared operations.

An outcome class need not be an evaluated achievement. It can record completion of an activity, a configuration of access, or another analytically selected feature. Whether reaching that class is desirable, legitimate, or beneficial requires further criteria. The formal target supplies a condition to be reached.

A configuration can also contain information that the quotient omits. Two raw states can share a completion flag while differing in remaining resources or available permissions. If those differences affect later activity, a quotient that merges them may fail dynamic compatibility or require a restricted task. The availability criterion below adds another compatibility requirement.

Table 1 collects the notation. The separate domains help prevent a mechanism, a word, a requirement set, and an outcome from being treated as interchangeable forms of possibility.

2.2 Mechanism Fibres and Outcome Images

A target can have several supporting mechanisms. Retaining those witnesses provides more information than retaining only the target's membership in an outcome set. This subsection defines both levels and explains their different uses.

For a declared admissible set $\mathcal{A}(x) \subseteq M$, define

$$\begin{aligned} \Pi_{\mathcal{A}}(x, v) &= \{m \in \mathcal{A}(x) : q(mx) = v\}, \\ P_{\mathcal{A}}(x) &= \{v \in Q : \Pi_{\mathcal{A}}(x, v) \neq \varnothing\}. \end{aligned} \tag{1}$$

The fibre retains mechanisms supporting the target. The image records which targets have at least one such witness.

The distinction matters when mechanisms differ in prerequisites, cost, authority, or susceptibility to failure. Two configurations can have the same outcome image and different support fibres. A comparison concerned only with outcome membership will merge those differences by design.

DRAFT

Table 1: Symbols for feasible paths and persistent enabling conditions

Symbol	Domain and role
S, M, q	Configurations, mechanisms, and selected quotient outcome map
$\mathcal{A}(x)$	Admissible composite mechanisms in M at x
$B, T_b, A(x)$	Primitive labels, their total maps, and admissible labels at x
$h, \mathcal{W}_{A,h}(x, v)$	Horizon and prefix-feasible words reaching outcome v
E, P	Finite condition universe and available-condition scenario $P \subseteq E$
$\mathrm{req}(b, x), R_x(w)$	Step requirements and their union along a word
$\mathscr{R}, \Phi, \kappa$	Target requirement family, availability, and least disabling cardinality

The admissible set is additional structure. The same abstract action can be paired with all mechanisms available, a restricted repertoire, or an empty repertoire. An empirical application can define admissibility from represented conditions, but the definition and its evidence must be supplied.

A nontrivial outcome repertoire can remove $q(x)$ when the question concerns changed outcomes. That exclusion should be stated explicitly. Reaching the current class can still involve activity, maintenance, or a costly return to the same observed condition. A changed-outcome criterion and an occurrence criterion therefore differ.

The fibre also depends on the level at which mechanisms are individuated. Distinct labels with the same map can count as different routes in a word model while being represented by one transformation in a mechanism model. A support count needs a declared carrier and equality criterion before it can be interpreted.

2.3 Repertoire Enlargement and Strict Outcome Change

Adding mechanisms can increase support without adding outcomes. This subsection derives the exact inclusion result and the condition for strict enlargement of the image.

Fix x and q, and suppose $\mathcal{A}(x) \subseteq \mathcal{A}'(x)$. Every mechanism in $\Pi_{\mathcal{A}}(x, v)$ also belongs to $\mathcal{A}'(x)$ and reaches the same target. Hence

$$\Pi_{\mathcal{A}}(x, v) \subseteq \Pi_{\mathcal{A}'}(x, v)$$

for every v. Taking nonempty fibres gives $P_{\mathcal{A}}(x) \subseteq P_{\mathcal{A}'}(x)$.

Proposition 2.1 (Strict outcome enlargement)**.** *The outcome inclusion is strict precisely when some added admissible mechanism $m \in \mathcal{A}'(x) \setminus \mathcal{A}(x)$ satisfies $q(mx) \notin P_{\mathcal{A}}(x)$.*

Proof. If such a mechanism exists, its outcome belongs to the enlarged image and lies outside the old image, proving strictness. Conversely, if the inclusion is strict, choose an outcome in the difference. It has a witness in $\mathcal{A}'(x)$. That witness cannot belong to $\mathcal{A}(x)$, because then the outcome would belong to the old image. It is therefore an added mechanism with the required property. □

The proposition prevents a common overstatement. A larger mechanism repertoire does not entail a strictly larger set of outcomes. Added mechanisms can all reach classes that were already supported. Their importance can lie in alternative routes, different requirements, or increased robustness.

The comparison also holds the seed and quotient fixed. Changing the initial configuration or the outcome criterion asks a different question. A refinement of the quotient can reveal newly distinguished outcomes without creating new raw configurations. A changed seed can alter reachability under an unchanged repertoire.

The later route example uses this distinction constructively. A new independently enabled route preserves the full-availability outcome set while improving survival under specified condition removals. Its contribution appears in the support structure and scenario comparison.

DRAFT

2.4 Primitive Words and Prefix Feasibility

A composite mechanism can conceal the requirements of its execution. Primitive words retain the intermediate states at which those requirements must be checked. This subsection defines the path representation.

Choose an operation alphabet B and total maps $T_b : S \to S$ for its labels. Declare $A(x) \subseteq B$ to be the labels available at configuration x. A word $w = b_1 \cdots b_k$ induces

$$x_0 = x, \qquad x_i = T_{b_i} x_{i-1}.$$

Execution proceeds left to right. The empty word has $T_\epsilon = \mathrm{id}_S$ and performs zero steps.

Definition 2.2 (Feasible target-word fibre)**.** For an integer $h \geq 0$, define

$$\mathcal{W}_{A,h}(x, v) = \left\{ w \in B^{\leq h} : \begin{array}{l} b_i \in A(x_{i-1}) \text{ for every step } i, \\ q(T_w x) = v \end{array} \right\}. \tag{2}$$

The empty word is feasible and has outcome $q(x)$.

The prefix condition checks availability where each operation is actually to be executed in the modeled trajectory. Checking all labels only at the initial state would ignore preparation, depletion, withdrawal, and other changes that alter later availability.

The total maps permit an unrestricted simulated trajectory even for a word that would be blocked under the availability rule. Such simulation is a computational construction. It does not assert that the blocked steps occurred. The later persistent requirement theorem uses this fixed trajectory under explicit assumptions.

State-dependent action sets and goal-directed finite paths have an established formulation in discrete planning [3, Formulation 2.1]. The present use focuses on how route requirements and shared conditions affect access to a socially interpreted target.

A nonempty feasible fibre establishes a modeled route. Selection, information, intention, and actual execution remain additional questions. The paper returns to each after deriving the route-level requirements.

2.5 Partial Maps and Composition Domains

Partial maps encode availability directly in their domains. Their composition law exposes the same prefix responsibility in another representation. This subsection derives the law and its associativity.

Suppose m is executable on $D_m \subseteq S$ and n on $D_n \subseteq S$. Their partial composite has domain

$$D_{n \circ m} = \{x \in D_m : mx \in D_n\}. \tag{3}$$

The first condition permits the initial operation. The second permits the next operation at its resulting state.

For three maps m, n, p, either parenthesization is defined exactly when

$$x \in D_m, \qquad mx \in D_n, \qquad n(mx) \in D_p.$$

Where defined, both composites give $p(n(mx))$. Equality of domains and values proves associativity of partial-map composition.

The domain is part of the partial map. Two partial maps with equal values on their common domain can still differ because one is defined at additional configurations. That distinction matters to feasibility.

Attaching arbitrary domains to elements of an existing total semigroup can violate its product equations. A product that equals the total identity can represent a word whose partial execution fails midway. Section 3 gives an explicit two-state example.

Using the semigroup of partial maps resolves this particular mismatch because equality then includes domains. It also changes the representation. The relationship to an earlier total product must be stated through a map or a declared loss of information.

DRAFT

2.6 Availability on a Quotient

A quotient can preserve total effects while failing to preserve availability. This subsection derives the additional condition needed for feasible paths to be well-defined on outcome classes.

Assume $q(x) = q(y)$ implies $q(T_b x) = q(T_b y)$ for every b, as supplied by the base compatibility condition. To define a class-level available-label set, require also

$$q(x) = q(y) \implies A(x) = A(y).$$

This states that availability is constant on each quotient class.

Proposition 2.3 (Quotient feasibility)**.** *Under these two conditions, equivalent seeds have the same feasible words, and their corresponding prefix states and endpoints are q-equivalent.*

Proof. The seeds are equivalent by assumption. Suppose the prefix states before a step are equivalent. Their available-label sets agree, so the next label is executable from one exactly when it is executable from the other. If executed, compatibility of T_b gives equivalent successor states. Induction proves the claim for every finite word. □

If availability differs inside a quotient class, that class cannot have one exact available-label set. A coarse completion flag can therefore be adequate for a current report while failing as a planning state. The repair can retain a permission variable or another sufficient distinction.

The result also shows why outcome aggregation and feasibility aggregation should be audited separately. A quotient can serve one purpose without serving the other. The model should identify which responsibility its reduced state actually meets.

2.7 Interpretive Responsibilities of a Feasible Path

The formal path model specifies a sequence whose steps satisfy declared availability rules and reach a target. This final subsection identifies the interpretation needed before that result supports a claim about social possibility.

The operation labels require an account of what counts as one step. The configuration needs to contain the conditions on which the update and availability depend. The target needs a criterion connecting its quotient class to the research question.

The horizon also has a meaning. A bound on the number of operations can differ from a time limit, a cost limit, or a participant's attention budget. Those constraints require their own variables or annotations.

Feasibility can refer to different kinds of permission and capacity. A formally allowed step may remain practically inaccessible, while an unapproved step may remain physically possible. The chosen availability rule should state which sense is modeled.

The resulting construction is useful because it exposes these requirements. It preserves the route long enough to examine its intermediate conditions, then permits carefully justified reductions for narrower questions.

3 Compression of Paths into Mechanism Products

This section determines what survives when a mechanism word is replaced by its product. It derives the exact criterion for recovering a word's feasibility, presents a prefix counterexample, and distinguishes that failure from the legitimate existential claim that a product has some feasible realization. It then compares alternative representations and the loss of path-dependent information. The distinction between these claims is essential to a fair assessment of composite models.

DRAFT

3.1 Product Evaluation and the Target of Compression

A product representation summarizes a word through an algebraic evaluation. The summary can be adequate for one question and inadequate for another. This subsection specifies the maps before testing their information content.

Fix a seed x and a word domain W. Let $e : W \to N$ be the chosen product evaluation, where N is a semigroup or an identity extension appropriate to the word domain. Let $F_x : W \to \{0, 1\}$ indicate prefix feasibility from x.

For zero-step words, the product representation needs an element acting as id_S. The generated transformation monoid provides such an element. An existing identity in an abstract semigroup requires a unital-action check before it can represent zero steps. A fresh formal unit is another valid construction.

The evaluation map can identify different words because of algebraic equations. The action can identify different products because its representation is nonfaithful. Evaluation at one seed can identify different transformations because they agree at that seed. Each stage can lose a different kind of information.

The target query determines whether the loss matters. Endpoint calculation uses the action law and can remain exact after product compression. Prefix feasibility, occurrence count, duration, or path cost can depend on distinctions that the product merges. Their recovery requires a separate factorization test.

3.2 Exact Recovery of Word Feasibility

The next proposition characterizes the strong claim that the feasibility of each selected word is determined by its product. The proof is included in full because later interpretive distinctions depend on its quantifiers.

Proposition 3.1 (Product-level recovery of word feasibility)**.** *There is a function $\bar{F}_x : e(W) \to \{0, 1\}$ with $F_x = \bar{F}_x \circ e$ if and only if*

$$e(w) = e(w') \implies F_x(w) = F_x(w') \qquad (w, w' \in W). \tag{4}$$

Proof. If the factorization holds, equal products are inputs to the same function $\bar{F}_x$, so their feasibility values agree. This proves necessity.

Conversely, suppose feasibility is constant on every evaluation fibre. For $m \in e(W)$, choose any word with product m and define $\bar{F}_x(m)$ to be its feasibility value. The fibre is nonempty because m is attained, and the assumed constancy makes the definition independent of the choice. Substituting $m = e(w)$ gives the required factorization for every word. □

The criterion is relative to the seed and word domain. A product filter intended to work at every configuration requires the same condition at each relevant seed. A result proved on a finite catalog leaves omitted words outside its scope.

The theorem also identifies an efficient counterexample method. Two words with the same product and different feasibility values disprove exact recovery. The failure concerns the information retained by the product, so a more elaborate function of that product alone cannot repair it.

A repair can retain a word witness, enrich the product representation with domain information, or narrow the query to existential availability. These choices preserve different responsibilities. The following subsections examine them separately.

3.3 The Two-State Prefix Trap

A two-state example shows how total algebraic equality can hide a failed intermediate step. This subsection derives the collision and identifies exactly which conclusion it defeats.

Let f flip $S = \{0, 1\}$, so its total transformation satisfies $f^2 = \mathrm{id}_S$. Allow f only at state zero. Take seed zero and a word domain containing ϵ and ff.

DRAFT

The empty word is feasible and ends at zero. The first f in ff is available and reaches one. The second f is unavailable there. Consequently

$$e(\epsilon) = e(ff) = \mathrm{id}_S, \qquad F_0(\epsilon) = 1, \quad F_0(ff) = 0.$$

Proposition 3.1 excludes exact recovery of word feasibility from this total product.

The endpoint calculation remains correct for the unrestricted total maps: both words have total effect equal to the identity. The failure arises when that calculation is used to infer execution of the particular factorization ff. Its intermediate availability condition has been lost.

The associated partial flip has domain $\{0\}$ and maps zero to one. Its partial square has empty domain because the second step is unavailable. It therefore differs from the identity partial map, whose domain is all of S.

This comparison shows that restricting domains changes the algebraic structure. The total equation $f^2 = \mathrm{id}_S$ cannot be transferred to the restricted partial maps without checking domains. The partial-map semigroup records the missing distinction through equality of domains and values.

3.4 Existential Availability of a Composite

A product can have some feasible realization even when another realization is infeasible. This subsection develops that weaker claim and explains why it remains legitimate after the prefix counterexample.

Define the existentially available product set

$$\mathcal{A}_x^{\exists} = e(\{w \in W : F_x(w) = 1\}). \tag{5}$$

This set is always well-defined. Membership means that at least one represented word with the product is feasible from the seed.

The construction is an image of a subset. It requires no constancy of feasibility on product fibres. It therefore survives the failure identified by Proposition 3.1. The two claims have different quantifiers.

For an attained product, define

$$\begin{aligned} F_x^{\exists}(m) &= \max\{F_x(w) : e(w) = m\}, \\ F_x^{\forall}(m) &= \min\{F_x(w) : e(w) = m\}. \end{aligned} \tag{6}$$

The fibre is nonempty and the values are Boolean, so both extrema exist. The first asks whether some realization is feasible; the second asks whether every represented realization is feasible.

Proposition 3.2 (Existential and universal product bounds)**.** *For every $w \in W$,*

$$F_x^{\forall}(e(w)) \le F_x(w) \le F_x^{\exists}(e(w)).$$

Exact recovery of word feasibility from products holds precisely when $F_x^{\exists} = F_x^{\forall}$ on $e(W)$.

Proof. A value in a nonempty Boolean set lies between its minimum and maximum, proving the inequalities. The extrema agree exactly when every value in the fibre is the same. Applying this fact to every attained product gives Equation (4) and hence the factorization criterion. □

In the prefix trap, the identity product has a feasible empty realization and an infeasible two-step realization. Its existential value is one and its universal value is zero. A product catalog can correctly report the existence of a feasible realization while remaining insufficient to execute an arbitrarily selected factorization.

This distinction repairs an overly broad criticism of composite models. Product-level availability can be meaningful when it explicitly means existence of a feasible realization. A planner or actor still needs a witness, a method for finding one, or another execution certificate. The product's existence predicate and the route record answer different questions.

DRAFT

Table 2: Representational reductions and their question-specific limits

Retained object	Typical supported question	Information requiring a further check
Word and trajectory	Which steps lead to the target?	Actual occurrence, duration, and evidential interpretation
Composite product	What composite effect is represented?	Factorization, prefix feasibility, and path-dependent cost
Outcome image	Which target classes have witnesses?	Alternative witnesses and shared requirements
Requirement family	Which persistent scenarios permit the target?	Order, consumption, route identity, and control authority
Minimal family	What is the Boolean availability function?	Dominated routes that may differ in cost or meaning

3.5 Alternative Representations of Execution

Several constructions can preserve the information needed for feasibility. This subsection compares them through their intended use and the assumptions they retain.

The full word and its trajectory provide the most direct representation within the current alphabet. Every prefix can be checked against availability. The record also retains order and repetition, although dates, durations, and actual occurrence still require additional information.

A partial transformation retains its domain and endpoint effect. It can identify whether a composite is executable from a seed, provided the composite is constructed through partial-map composition. It can still merge different feasible histories that share the same partial effect.

An existential product catalog retains the fact that a feasible realization exists. A separate witness map can associate an available product with one feasible word. The witness need not preserve every alternative route, so questions about robustness or history can require the larger fibre.

A requirement profile can summarize the conditions needed by an already evaluated path. Under the persistent positive assumptions of the next section, it answers feasibility across scenarios. It omits the path's order only after that order has been used to calculate the requirements.

Table 2 organizes these levels and their limits. The table records questions supported by a representation; it supplies no evidence that a particular social interpretation is adequate.

3.6 Costs, Duration, and Other Path Queries

Feasibility is one path-dependent property among several. The same factorization logic applies to cost, duration, occurrence count, and historical classification. This subsection identifies the common criterion and its limits.

Let $g : W \to V$ be a path query. It is recoverable from the product exactly when equal products imply equal query values. The proof repeats the representative independence argument of Proposition 3.1. The result is general, while each query requires its own definition.

A positive cost per occurrence can fail this criterion under idempotence. If b and bb have the same product but each occurrence costs a positive amount, their total costs differ. The product alone loses the count needed by the sum.

Duration can fail for the same reason, and context-dependent duration introduces further dependencies. A path record can retain occurrence times or step-specific durations. A product annotation requires proof that the annotation respects the algebraic equations used by evaluation.

A minimum-cost product value is another existential summary: it chooses the least cost among realizations, when that minimum exists. It differs from the cost of an arbitrary realization. Using it for execution requires a route attaining the value or a justified approximation.

The broader lesson is to name the query before compressing. Endpoint, existence, all-realization feasibility, minimum cost, and actual history place different requirements on the record. A composite

DRAFT

model can serve several of them, but the connection must be proved for each.

3.7 Interpretive Consequences of Product Compression

The product analysis clarifies a central philosophical distinction between a possible effect and the process through which that effect can be realized. This final subsection states the consequence within the paper's scope.

An algebraic product can represent the combined effect of a sequence while leaving its execution history unresolved. The distinction permits efficient calculation without treating the compressed object as a complete account of activity.

A feasible realization adds more than a factorization in symbols. Its prefixes must meet the declared conditions. An actual realization adds selection and occurrence evidence. These layers can be coordinated without being identified.

The existential product construction also preserves a constructive role for composite reasoning. A researcher can ask which effects have some feasible realization, then inspect their witnesses when requirements or historical questions matter. The limitation concerns the scope of the summary.

The next section derives one such summary for persistent enabling conditions. Its set union is justified only after ordered execution has determined the relevant requirements.

4 Persistent Enabling Conditions

This section introduces the assumptions under which a path's requirements can be summarized by a set. It derives prefix feasibility from that set, constructs target availability across scenarios, and examines monotonicity and its counterconditions. The set representation is intentionally restricted: persistence, positive requirements, and scenario-independent transitions perform essential work in the proof.

4.1 Scenarios and the Meaning of a Condition

A scenario specifies which enabling conditions are available during execution. This subsection separates those conditions from operation labels and from the configuration variables that change along a path.

Let E be a finite universe of condition labels. A scenario is a subset $P \subseteq E$. Membership means that the corresponding condition is available under the adopted interpretation. The label can represent a permission, an accessible facility, or another prerequisite suited to the persistent model.

A condition label is not automatically a resource quantity. Membership in P records availability as a Boolean fact. Repeated use leaves that membership unchanged under the assumptions below. A token that is spent, a capacity that becomes occupied, or a permission that expires requires a different representation.

The universe also determines the resolution of the analysis. Two condition labels can depend on one underlying facility or one actor's control. Treating them as distinct elements does not establish independent failure, independent provision, or independent authority. Later sections add explicit control and probability structures.

The configuration x can change while the scenario stays fixed. A preparation step may change which completion operation is relevant, and its requirements can depend on the resulting state. The scenario specifies the persistent stock of available conditions against which each requirement is tested.

4.2 The Persistent Positive Requirement Assumptions

The main theorem depends on a precise relation between scenarios, transitions, and availability. This subsection states that relation before introducing the requirement union.

Assumption 4.1 (Persistent positive requirements)**.** Let E and B be finite, fix a seed x and a finite horizon h, and use total deterministic maps $T_b : S \to S$. For each operation and configuration, declare

DRAFT

$\mathrm{req}(b, x) \subseteq E$. In scenario P, operation b is available at x exactly when $\mathrm{req}(b, x) \subseteq P$. The scenario remains fixed through execution. Transitions and requirements are independent of P except for this availability test.

Positive means that requirements ask for the presence of conditions. An added condition cannot directly invalidate such a test. A rule requiring the absence of a condition would fall outside this assumption.

Persistent means that using a condition leaves its availability unchanged. A facility can be modeled this way when the relevant uses are compatible and the considered horizon ignores depletion or occupation. The adequacy of that approximation requires a substantive argument.

Scenario independence ensures that the same word has the same unrestricted trajectory under every scenario. Availability can prevent its execution, but the total maps used to calculate its hypothetical path remain fixed.

State dependence of requirements is permitted. A label can require one condition at a preparatory state and another at a later state. The fixed trajectory determines which requirement applies at each step.

The finite alphabet and horizon make the candidate word catalog finite. The state carrier itself can be infinite. Finiteness of the relevant word set is enough for the finite enumeration used here, provided the maps and requirements can be evaluated.

4.3 The Requirement Profile of an Ordered Path

A requirement set is calculated after the ordered trajectory has been specified. This subsection defines that calculation and explains why its use of commutative union does not make execution commutative.

For $w = b_1 \cdots b_k$, simulate $x_0 = x$ and $x_i = T_{b_i} x_{i-1}$ using the total maps. Define

$$R_x(w) = \bigcup_{i=1}^{k} \mathrm{req}(b_i, x_{i-1}), \qquad R_x(\epsilon) = \varnothing. \tag{7}$$

The set contains every condition requested at any step along that specific trajectory.

The order of the word enters through the prefix states. Reversing two labels can change the state at which a requirement is evaluated and can therefore change the union. Even when the two unions agree, the endpoints or intermediate states can differ.

Union discards repeated requests for the same condition. That is justified for the selected feasibility question because persistent availability is tested identically each time. The same reduction would fail for a consumable quantity unless the quantity were represented separately.

An unrestricted simulated path can pass through states that a particular scenario would prevent the actor from reaching. The simulation computes what the total maps would do. The requirement test then decides whether that entire path is executable in the scenario.

This separation is legitimate under the fixed-transition assumption. If the scenario changes the transition itself, the precomputed trajectory may cease to be the relevant one. The theorem would then need a revised model.

4.4 Derivation of Path Feasibility

The requirement profile replaces a sequence of availability tests with one inclusion test. This subsection gives the complete equivalence and its inductive interpretation.

Proposition 4.2 (Persistent path requirements)**.** *Under Assumption 4.1, a word w is prefix-feasible in scenario P if and only if $R_x(w) \subseteq P$.*

Proof. If the word is feasible, every step satisfies $\mathrm{req}(b_i, x_{i-1}) \subseteq P$. An element of $R_x(w)$ belongs to at least one of those requirement sets, so it belongs to P. Hence $R_x(w) \subseteq P$.

DRAFT

Conversely, suppose $R_x(w) \subseteq P$. Every step requirement is a subset of the union and therefore a subset of P. The first operation is available at x_0. Its execution produces the simulated state x_1 because the transition map is fixed. Inductively, each later operation is available at its simulated prefix state and produces the next one. Thus every prefix is executable.

For the empty word, both conditions hold: execution has no steps, and the empty requirement set is contained in every scenario. □

The reverse direction shows why a pure set-theoretic shortcut needs the dynamic assumptions. The inclusion ensures each test, while fixed transitions ensure that the tests are being applied at the correct prefix states.

The theorem is about one word and one seed. Different words can have equal requirement sets while reaching different targets. A target family must therefore filter words by their endpoint before using their requirements to describe target availability.

4.5 Target Availability across Scenarios

A target is available when at least one target-reaching word has its requirements satisfied. This subsection constructs that family and derives the Boolean availability function.

Fix $v \in Q$ and define

$$\begin{aligned} \mathscr{R} &= \{R_x(w) : w \in B^{\leq h},\ q(T_w x) = v\}, \\ \Phi(P) &= \mathbf{1}\{\exists R \in \mathscr{R} : R \subseteq P\}. \end{aligned} \tag{8}$$

The family is a set of requirement sets. Several words can contribute the same member, so route multiplicity is already compressed.

If a feasible target word exists, its requirement set belongs to $\mathscr{R}$ and lies inside P by Proposition 4.2. Thus $\Phi(P) = 1$. Conversely, a member of $\mathscr{R}$ inside P has a witnessing target word, which the proposition makes feasible. This proves the availability interpretation.

The empty-family case gives $\Phi(P) = 0$ for every scenario. There is no target word within the selected catalog and horizon. This result leaves longer paths or enlarged operation repertoires outside its scope.

If $\varnothing \in \mathscr{R}$, the target is available under every scenario. One target word requires none of the declared conditions. That statement is relative to the condition universe. The word can still have costs or dependencies omitted from E.

If $v = q(x)$, the empty word contributes the empty requirement set. The target class is already occupied. A question about performing an activity must therefore use a separate occurrence criterion or a positive-word restriction.

4.6 Scenario Monotonicity

The persistent positive model makes availability monotone under addition of conditions. This subsection derives the result and states its interpretive scope.

Suppose $P \subseteq P'$ and $\Phi(P) = 1$. Some $R \in \mathscr{R}$ satisfies $R \subseteq P$. By transitivity of inclusion, $R \subseteq P'$. The same word witness therefore remains feasible, so $\Phi(P') = 1$.

The proof retains the old witness. It leaves open whether the enlarged scenario also enables new routes. It also leaves route selection open: an actor can choose differently after new conditions become available.

Monotone availability therefore need not imply monotone realized success. A changed selection rule can assign less probability to successful paths despite their continued existence. The probability and policy sections separate these questions.

The result also preserves the target and observation criterion. If the meaning of success changes when conditions are added, the two Boolean functions describe different tasks. Their comparison requires an alignment.

DRAFT

The theorem's value lies in making the retained assumptions explicit. A claim that more conditions improve possibility should state whether the conditions are positive, persistent, and independent of the transition definition.

4.7 Counterconditions to the Set Representation

The requirement union can fail when conditions are consumed, when absence is required, or when the scenario changes the dynamics. This subsection gives concrete boundaries and identifies the needed repairs.

Consumption defeats Boolean persistence. If one token is available and a word spends it twice, a set containing the token label cannot distinguish one use from two. A quantity ledger with prefix updates is needed. The resource section derives such a construction.

A negative requirement can defeat monotonicity. Consider one target operation available exactly when condition d is absent. The target is available under the empty scenario and unavailable under $\{d\}$. Adding a condition has reversed success because the rule falls outside positive requirements.

Scenario-dependent transitions can produce another failure. If an operation reaches the target when d is absent and reaches a different state when d is present, an unchanged availability test cannot preserve the old endpoint witness. The transition assumption has changed.

A persistent condition can also be available to only one participant or one simultaneous use. A global set P can conceal that allocation. An actor-specific or capacity model should represent the relevant restriction.

These cases do not invalidate the restricted theorem. They identify which feature of an application requires additional structure. The set summary is useful when its assumptions fit the question, and its limits should remain part of the record.

4.8 Interpretive Role of the Requirement Family

The requirement family describes how a target depends on declared persistent conditions across scenarios. This final subsection distinguishes that result from broader claims about causation, agency, and value.

A member of $\mathscr{R}$ is a sufficient condition set for one modeled route. It need not identify a complete causal explanation of the target. The transition rules, initial configuration, and horizon remain background inputs.

The family can reveal shared dependencies among routes. Two paths that look different at the operation level can require the same condition. That common requirement matters to removal analysis.

The family also leaves control open. Knowing that a route requires a permission does not identify who can grant, withdraw, or revise it. The actor-control extension adds that information explicitly.

The next section compresses the family further for Boolean availability. Its proof makes clear which route differences survive and which require separate retention.

5 Minimal Requirement Families and Availability

This section derives the inclusion-minimal representation of a finite availability function. It proves sufficiency and uniqueness, examines duplicate and dominated routes, and identifies the information that the reduction discards. The result provides a compact basis for condition-removal analysis while preserving the distinction between Boolean availability and a complete route description.

5.1 Inclusion-Minimal Members

A requirement set can be redundant for availability when another target route requires a proper subset of its conditions. This subsection defines the reduction and explains the order used by minimality.

DRAFT

For the finite family $\mathscr{R} \subseteq 2^E$, let $\text{Min}(\mathscr{R})$ contain those members R for which no proper subset also belongs to $\mathscr{R}$. The order is set inclusion. Minimality therefore compares the conditions required, rather than route length, cost, or desirability.

Two distinct minimal members are incomparable under inclusion. If one were a proper subset of the other, the larger would fail minimality. Thus $\text{Min}(\mathscr{R})$ is an antichain.

Every member of a nonempty finite family contains a minimal member of that family. Starting from the member, repeatedly choose a proper subset that remains in the family whenever one exists. Finiteness forces the descent to stop. The terminal member is minimal.

This existence argument is the key step in the sufficiency proof. It explains why a successful scenario supported by a redundant requirement set also contains a minimal supporting set.

5.2 Sufficiency for Boolean Availability

Removing redundant requirement sets preserves the availability function. This subsection proves the result in both directions.

Proposition 5.1 (Minimal-family sufficiency)**.** *The families $\mathscr{R}$ and $\text{Min}(\mathscr{R})$ define the same function Φ. Moreover, $\text{Min}(\mathscr{R})$ is exactly the family of inclusion-minimal successful scenarios.*

Proof. Suppose P contains a member R of $\mathscr{R}$. By finite descent, R contains a minimal member R_0. Then $R_0 \subseteq P$, so the minimal family also supplies success. Conversely, every minimal member belongs to $\mathscr{R}$, so success supplied by the minimal family is success under the original family.

Now suppose P is an inclusion-minimal successful scenario. Some $R \in \mathscr{R}$ lies inside P. A strict inclusion would make R a smaller successful scenario, contradicting minimality. Hence $R = P$, and no proper subset of it belongs to $\mathscr{R}$. Thus $P \in \text{Min}(\mathscr{R})$.

Conversely, let $R \in \text{Min}(\mathscr{R})$. It is successful because it contains itself. If a proper subset were successful, it would contain a member of $\mathscr{R}$ properly inside R, contradicting minimality. Therefore R is a minimal successful scenario. □

The theorem proves an exact reduction for one question: which scenarios permit at least one target word. It preserves all answers to that Boolean question on the declared condition universe.

The result includes the boundary cases. If $\mathscr{R}$ is empty, both families describe constant failure. If $\varnothing$ belongs to $\mathscr{R}$, it is the sole minimal member, and both families describe constant success.

The reduction can be applied after finite path enumeration. It should not be used to discard path records before their ordered requirements have been calculated.

5.3 Uniqueness and the Converse Representation

The minimal family can be recovered from the availability function itself. This subsection proves uniqueness and explains the converse for any finite monotone Boolean function.

Proposition 5.1 identifies the minimal family through successful scenarios. Since those scenarios are determined by Φ, the family is uniquely determined by the function.

More generally, let $\Psi : 2^E \to \{0, 1\}$ be monotone on finite E. Every successful scenario contains an inclusion-minimal successful subset by finite descent. Every superset of such a subset is successful by monotonicity. Hence Ψ has the same minimal-family representation.

This converse describes information in a Boolean function. It does not reconstruct the actual mechanisms or paths that produce success. Many different route systems can have the same availability function.

For example, one route requiring $\{a, b\}$ and several differently ordered routes with the same requirement set produce the same minimal family when their target and scenario effects agree. Their histories and costs can remain different.

The uniqueness claim therefore has a precise scope. The minimal condition antichain is unique for the fixed availability function and condition universe. A unique social mechanism does not follow.

DRAFT

5.4 Duplicate Routes and Requirement Supersets

Two kinds of redundancy disappear under the reduction: repeated copies of a requirement set and members containing a smaller member. This subsection compares their interpretive consequences.

Duplicating an operation label can produce additional word witnesses without changing any requirement set. The family $\mathscr{R}$ is a set, so these duplicates already disappear before minimality is applied.

A route requiring $\{a, b, c\}$ is redundant for Boolean availability when another target route requires $\{a, b\}$. Every scenario supporting the first also supports the second. Removing the larger member therefore preserves Φ.

The larger-requirement route can still be faster, cheaper, easier to understand, or preferred by participants. Those properties belong to other queries. Set inclusion of prerequisites does not order every aspect of a route.

Historical significance also can survive Boolean redundancy. A route may have been the one actually used or the one through which a practice was formed. Its omission from the minimal family is a compression for availability, rather than a reason to remove its source record.

A useful representation can therefore retain a map from minimal or nonminimal requirement sets to their word witnesses. The availability calculation uses the antichain, while further inquiry can recover the routes relevant to another question.

5.5 Finiteness and Failure of Minimal Descent

The finite assumptions ensure the existence of minimal supporting members. This subsection gives a countercondition that clarifies their role.

Let the condition universe be the natural numbers and consider the family $R_n = \{n, n+1, \ldots\}$. Every member contains the proper member R_{n+1}. The family therefore has no inclusion-minimal member.

The full universe contains every R_n and is successful under the existential availability rule. The empty minimal family would describe failure. The finite minimal-family theorem thus cannot be transferred to this example without an additional existence condition.

Finiteness of the condition universe makes every family of condition sets finite. A finite word catalog also supplies a finite family even when a larger ambient universe is available. The proof needs the relevant descent to terminate.

An infinite application can still have minimal members, but their existence and sufficiency require proof. A compact notation for an infinite family should not be treated as a substitute for that argument.

The boundary also separates mathematical representation from computation. A family can be finite yet too large to enumerate practically. Efficient construction of its minimal members is a further implementation question.

5.6 Relation to Provenance and Reliability Representations

The minimal-family construction has established mathematical relatives. This subsection states the comparisons narrowly and preserves the present model's distinct input assumptions.

Provenance semirings retain information about alternative and joint contributions to outputs [1]. The present requirement family likewise distinguishes alternative sufficient sets after ordered paths have been evaluated. Its Boolean reduction deliberately omits some multiplicity and route information.

Reliability analysis uses path and cut representations to characterize system success and failure [6]. The next section uses the same elementary hitting condition for declared enabling requirements. The social interpretation of those requirements remains an additional modeling commitment.

The comparisons identify antecedent structures and help delimit novelty claims. They leave open whether a particular social path system satisfies persistence, positive requirements, and fixed transitions.

The contribution here is the sequence of representational steps: preserve ordered paths, derive their requirements, reduce only for a specified availability question, and then examine shared dependence. Each reduction has an explicit information boundary.

DRAFT

5.7 Interpretive Limits of Minimal Availability

The minimal family is a compact description of scenario-dependent possibility. This final subsection states what it can support and what requires a richer account.

It can identify which combinations of declared conditions suffice for some target route. It can support exact condition-removal analysis. It can distinguish two systems with the same target but different shared dependencies.

It leaves the identities of route providers, the costs of execution, the information available to participants, and the reasons for selection open. Those features can change realization without changing the Boolean availability function.

Minimality also carries no automatic normative priority. A route requiring fewer conditions can be less desirable under another criterion. The model should state which comparison the word minimal denotes.

The reduction is therefore best used alongside its path witnesses and interpretation record. The next section derives a further question that the minimal family does answer: which condition removals disable the target.

6 Condition Removal and Target Robustness

This section derives the condition under which removals disable a target. It distinguishes inclusion-minimal disabling sets from minimum cardinality, treats empty and unavoidable cases, and examines the dependence of robustness on the chosen condition universe. The calculation holds the path catalog, transition rules, horizon, target, and persistent requirement interpretation fixed. Its result concerns a specific intervention on that model.

6.1 The Current Scenario and Its Feasible Requirements

Removal analysis begins with the routes available in a specified scenario. This subsection identifies the relevant family and explains why routes already blocked need no additional removal witness.

Fix $P \subseteq E$ and write

$$\mathscr{R}_P = \{R \in \mathscr{R} : R \subseteq P\}.$$

This is the family of requirement sets whose witnessing target routes are currently feasible. Let $F \subseteq P$ be a set of removed conditions. The remaining scenario is $P \setminus F$.

A member of $\mathscr{R}$ outside P remains outside every subset of P. Removal cannot make a positive requirement newly satisfied. Therefore only members of $\mathscr{R}_P$ can survive the intervention. This observation depends on the positive requirement assumption and would fail for a model in which absence can enable a route.

The removal set changes condition availability while retaining the operation system. A withdrawal that changes transition rules, induces a repair procedure, or creates new routes lies outside this fixed comparison unless those responses are already represented. The calculation should therefore be interpreted as a controlled scenario change within the model.

The target and horizon also remain fixed. A route that becomes available only after a longer recovery period cannot count as survival within the original horizon unless the comparison is expanded. A claim about permanent loss requires a different time scope from a claim about loss during a bounded interval.

6.2 The Disabling Criterion

A removal disables the target precisely when it interrupts every currently feasible requirement set. This subsection derives that equivalence and identifies its combinatorial interpretation.

Proposition 6.1 (Target loss through condition removal)**.** *Under Assumption 4.1,*

$$\Phi(P \setminus F) = 0 \quad \Longleftrightarrow \quad F \cap R \neq \varnothing \quad \textit{for every } R \in \mathscr{R}_P. \tag{9}$$

DRAFT

Proof. For $R \in \mathscr{R}_P$, survival means $R \subseteq P \setminus F$. Since $R \subseteq P$, this is equivalent to $R \cap F = \varnothing$. A requirement set outside P remains infeasible after removal. Hence the target survives exactly when some member of $\mathscr{R}_P$ has empty intersection with F. Negating that existential condition gives the displayed universal intersection condition. □

A set intersecting every member of a family is a hitting set. The theorem therefore converts the selected target-loss question into a hitting-set problem on the currently feasible requirements. It preserves the route dependence that an outcome image alone would omit.

The same Boolean structure appears in reliability analysis through path and cut sets [6, Chapter VII, Section 4]. Here the elements are declared enabling conditions of mechanism paths. The mathematical comparison supports the combinatorial derivation; the interpretation of those conditions in a social case requires independent evidence.

The universal quantifier is consequential. Disabling one prominent route can leave another feasible. A shared requirement can interrupt several routes at once. Counting routes without examining their intersections therefore leaves the removal question unresolved.

6.3 Minimal Disabling Sets and Minimum Cardinality

A disabling set can contain redundant removals. This subsection distinguishes the irredundancy question from the question of smallest size.

An inclusion-minimal disabling set ceases to disable the target when any one of its members is restored while the other removals remain. Equivalently, none of its proper subsets is disabling. Different minimal sets can have different cardinalities.

For example, suppose $\mathscr{R}_P = \{\{p, a\}, \{p, b\}\}$. Removing $\{p\}$ hits both requirements. Removing $\{a, b\}$ also hits both, and restoring either member of that pair leaves one route. Both sets are inclusion-minimal, while their cardinalities are one and two.

Define

$$\kappa(P, v) = \min\{|F| : F \subseteq P,\ \Phi(P \setminus F) = 0\}, \tag{10}$$

with $\kappa = \infty$ when no disabling set exists. This number answers the minimum-cardinality question.

The minimum is attained whenever the collection of disabling sets is nonempty because P is finite. Every minimum-cardinality set is inclusion-minimal: a proper disabling subset would contradict minimum size. The converse fails in the two-route example.

A complete report can retain the minimal disabling family alongside κ. The family shows alternative patterns of dependence that one scalar suppresses. A one-condition shared dependency and a two-condition distributed dependency can have different practical interpretations even when both appear among the target's possible failures.

6.4 Boundary Cases

The definition includes cases in which the target is already unavailable or cannot be disabled by removing the declared conditions. This subsection derives those values and prevents their interpretation from exceeding the model.

If $\Phi(P) = 0$, the empty removal set disables the target in the sense that it is already unavailable. Therefore $\kappa(P, v) = 0$. The value records the starting condition; it need not describe a failure caused by an intervention.

If a currently feasible requirement set is empty, every removal leaves its witnessing route available. An empty set cannot be hit. Thus the collection of disabling sets is empty and $\kappa = \infty$.

The infinite value is relative to the declared removals. A route requiring none of the conditions in E can still depend on background assumptions, an initial configuration, or resources omitted from the universe. The result says that removal of the modeled conditions cannot disable it within the fixed system.

DRAFT

If the target is available and every feasible requirement set is nonempty, removing all of P hits every member of $\mathscr{R}_P$. Hence a finite disabling set exists. Together, these cases exhaust the finite persistent model.

The empty word deserves special attention. If the target is the current outcome class, it supplies an empty requirement route. The resulting infinite removal robustness concerns continued availability of that already occupied class through zero steps. It should not be read as unlimited protection of an activity or an institutional arrangement.

6.5 Bounded Removal Robustness

The minimum cut has an exact interpretation through bounded removals. This subsection states the equivalence and examines its scope.

For a nonnegative integer k, the target survives every removal of at most k conditions if and only if $\kappa(P, v) > k$. If $\kappa \leq k$, a disabling set within the bound supplies a counterexample. If $\kappa > k$, no set within the bound is disabling by definition.

The equivalence includes the boundary values. An already unavailable target fails survival under the empty removal. An empty-requirement route survives every finite removal from the declared universe. A finite positive value locates the first cardinality at which some removal can cause loss.

The quantifier concerns every removal up to the bound. A probability of failure under a particular distribution over removals is a different question. A small cut can be improbable, and many large cuts can collectively have substantial probability. The cardinality alone supplies neither distribution.

The measure also treats conditions as equal units of counting. If removal costs differ, a weighted minimum can be more appropriate. Its interpretation then depends on the cost model and whether joint removal costs are additive. The actor-control extension uses a separately declared unit of intervention.

6.6 Adding Routes and Preserving Outcomes

An additional route can improve removal robustness while leaving the outcome image unchanged. This subsection derives the direction of the comparison and distinguishes it from route duplication.

Suppose the old requirement family is contained in a new family at the same target and scenario. Every removal that disables the enlarged family must hit all old members as well as the added ones. The new disabling collection is therefore contained in the old one.

When minima exist under the same counting convention, the enlarged family has a minimum disabling cardinality at least as large as the old one. An added route can also make removal impossible within the declared universe if it has empty requirements. These conclusions follow from inclusion of the disabling collections.

Strict improvement requires the added route to survive some old minimum removals. Mere route duplication leaves the family unchanged. Adding a requirement superset of an existing member also leaves availability unchanged, so the disabling sets and κ remain unchanged.

The outcome target can remain the same throughout. The improvement concerns alternative support under condition loss. This distinction explains why outcome counts can miss a consequential change in access structure.

The comparison assumes the added route leaves old routes and their requirements intact. A new route that consumes a shared resource, changes a rule, or displaces a provider can alter the old family. Its net effect requires the revised model rather than the inclusion argument alone.

6.7 Representation Sensitivity and Shared Dependence

Condition-level robustness depends on how the condition universe is defined. This subsection examines that dependence and motivates a separate model of control.

Splitting one underlying prerequisite into two labels can change a naive count while leaving practical dependence unchanged. Conversely, combining several independent requirements into one label can

DRAFT

conceal the number of interventions needed to remove them. The condition ontology used for counting therefore needs an explicit analytical rationale.

A shared facility can also underlie several nominally different permissions. Their labels may be distinct in the requirement family while one failure removes them together. A distribution over condition scenarios can represent correlated availability, and a control map can represent common withdrawal. These enrichments answer different questions.

The removal theorem remains valid for the declared sets. The interpretive concern is whether a condition count corresponds to the intervention unit relevant to the inquiry. That correspondence should be tested rather than inferred from the number of labels.

The next section adds actors who can remove specified conditions. It shows formally how condition-level and actor-level robustness can differ, while keeping the meaning of control separate from legitimacy or intention.

7 Control over Enabling Conditions

This section adds a declared control structure to the persistent requirement model. It derives actor-level disabling sets from condition-level cuts, compares their cardinalities, and examines the limits of unilateral removal and coalition aggregation. The construction represents a specified capacity to alter conditions. Questions of legitimate authority, actual intention, and accountability require further interpretation.

7.1 Actors and Removal Capabilities

The condition universe identifies prerequisites, while a control map identifies who can remove them through modeled actions. This subsection specifies that additional structure.

Let $\mathcal{U}$ be a finite set of actors. For each $a \in \mathcal{U}$, let $D_a \subseteq E$ be the conditions that actor can remove under the declared control action. For a set of actors $U \subseteq \mathcal{U}$, define

$$F(U) = P \cap \bigcup_{a \in U} D_a. \tag{11}$$

The intersection restricts effective removals to conditions currently available.

The map represents unilateral removability of each listed condition and combines actors' removals by union. It can describe a credential manager, a provider controlling a facility, or another stipulated removal capability. A substantive application must identify what action produces the removal and under which circumstances.

Control in this sense differs from authorization to exercise it. An actor can possess a practical capacity whose use is contested, or hold a formal role whose practical effect depends on other participants. The model should state which capability the set D_a represents.

The actor set also requires an individuation criterion. An organization can be modeled as one actor for a decision made collectively, or as several actors when internal decisions matter. That choice changes the unit of the cut analysis and should match the research question.

7.2 The Actor-Level Disabling Criterion

The earlier cut theorem can be pulled through the control map. This subsection gives the derivation and identifies the family on which actor-level hitting sets are computed.

For each currently feasible requirement set $R \in \mathscr{R}_P$, define

$$A_R = \{a \in \mathcal{U} : D_a \cap R \neq \varnothing\}.$$

These are the actors able to remove at least one condition required by that route.

DRAFT

Proposition 7.1 (Disabling through actor control)**.** *Under the persistent model and the control rule in Equation* (11)*,*

$$\Phi(P \setminus F(U)) = 0 \quad \Longleftrightarrow \quad U \cap A_R \neq \varnothing \quad \textit{for every } R \in \mathscr{R}_P. \tag{12}$$

Proof. By Proposition 6.1, target loss is equivalent to $F(U) \cap R \neq \varnothing$ for every currently feasible R. Since $R \subseteq P$, the intersection with P in $F(U)$ does not change whether the union meets R. The union meets R exactly when some $a \in U$ satisfies $D_a \cap R \neq \varnothing$. By definition, this is equivalent to $U \cap A_R \neq \varnothing$. Applying the equivalence to every R proves the result. □

The actor family differs from the condition family. One actor can hit several route requirements through control over different conditions. Several actors can also control the same condition. The transformation records both forms of overlap.

If A_R is empty for some feasible route, the modeled actors cannot remove any condition required by it. That route survives every actor coalition under the declared rule. This is a limit of the selected control repertoire.

7.3 Actor Cardinality and Condition Cardinality

The minimum number of controlling actors can differ from the minimum number of removed conditions. This subsection derives the comparison and its coverage assumption.

Define

$$\kappa_{\text{actor}}(P, v) = \min\{|U| : U \subseteq \mathcal{U},\ \Phi(P \setminus F(U)) = 0\},$$

using infinity when no actor set disables the target. An already unavailable target has value zero through the empty actor set.

Consider two target routes with requirements $\{a\}$ and $\{b\}$. Condition-level disabling requires removal of both conditions, so $\kappa = 2$. If one actor controls both, that actor alone disables both routes and $\kappa_{\text{actor}} = 1$.

A conditional inequality follows when the conditions in a minimum condition cut are each removable by some actor. Choose one controller for each condition in that cut. Their combined removal contains the cut and therefore disables the target. The number of chosen actors is at most the cut's cardinality, so $\kappa_{\text{actor}} \leq \kappa$.

The coverage assumption matters. A condition cut can be mathematically available in the unrestricted removal model while lying beyond the control of every modeled actor. Actor-level disabling can then be impossible. The two minima concern different intervention classes.

The comparison is descriptive. A small actor cut can indicate concentrated control, a single maintenance responsibility, or another structural feature. Its desirability depends on the purposes, safeguards, and affected participants of the arrangement.

7.4 Weighted Control and Coalition Costs

Counting actors treats each actor's participation as one unit. A weighted model can represent another intervention cost. This subsection states the enrichment and its limits.

Assign a positive cost c_a to each actor's modeled removal action. The minimum weighted disabling cost is the minimum of $\sum_{a \in U} c_a$ over disabling actor sets. Positivity keeps the empty action at zero and gives every nonempty coalition a positive cost.

The additive convention is an assumption. Coordination can have costs beyond the sum of individual actions. One actor's participation can also reduce another's cost or make an action available. Such effects require a coalition-cost function or an explicit dynamic model.

A cost can represent effort, resources, or another declared quantity. It should not silently combine unlike burdens. A scalar minimum is meaningful only under the chosen measurement and aggregation rule.

DRAFT

The hitting condition remains the same when only the objective changes. The family identifies which coalitions disable the target, while the cost model ranks them. Keeping those layers separate makes the source of a changed conclusion inspectable.

7.5 Limits of the Union Control Rule

The control map assumes that actors can remove their listed conditions individually and that combined removals accumulate. This subsection identifies arrangements requiring a richer model.

Joint consent can be necessary to withdraw a condition. A simple set D_a for each actor cannot represent a condition removable only by a specified coalition when no member can remove it alone. The model would need coalition-dependent control.

Removal can also trigger replacement. Another actor may restore the condition, supply an alternative, or revise the route catalog. The fixed removal theorem describes the immediate scenario change within its horizon. A longer response model must include those operations.

Control can change through the very history under study. A formation procedure may allocate withdrawal power, and later revisions may redistribute it. A fixed map D_a then represents one stage. Modeling its evolution requires control variables and update rules.

Authority to amend the rules is another distinct role. An actor who can change which conditions count as requirements can alter the family itself, rather than remove members of a fixed scenario. The present cut theorem should not be applied to that intervention without revising its inputs.

7.6 Interpretive Consequences for Shared Dependence

The control extension clarifies why several routes can remain vulnerable to one decision. This final subsection connects the result to the paper's broader analysis of possibility.

Two doors opened by one revocable credential can supply two physical routes while depending on one control action. The condition and actor maps make that dependence explicit in a stipulated model. The example is a comparison design, not an empirical claim about an institution.

The same outcome image can therefore coexist with different distributions of control. An outcome count omits those differences. A route count can omit them as well when routes share conditions or controllers.

The extension also separates capacity from exercise. An actor-level cut identifies a modeled ability to disable the target. It leaves actual withdrawal, intention, justification, and responsibility open. Those questions need evidence and normative premises.

The practical research task is to specify the conditions, identify the relevant control actions, and test whether the union model fits their operation. The result can then support a bounded claim about shared dependence with its assumptions visible.

8 A Shared-Permission Route Illustration

This section develops a complete finite comparison of three route catalogs. The first contains two routes sharing a permission, the second duplicates one operation label, and the third adds a separately enabled direct route. The analysis derives the target words, requirement families, outcome sets, and disabling sets. It then examines control, resources, and state compression as explicit variants of the same illustration.

8.1 Four-State Construction and Condition Universe

The example uses four configurations so that every operative transition and target path can be inspected. This subsection defines the carrier and the interpretation of its conditions.

Let $S = \{s_0, s_A, s_B, t\}$, with seed s_0, target t, equality as the base distinction, and horizon $h = 2$. The intermediate states represent preparation through two hypothetical facilities. The target records completion under the selected criterion.

DRAFT

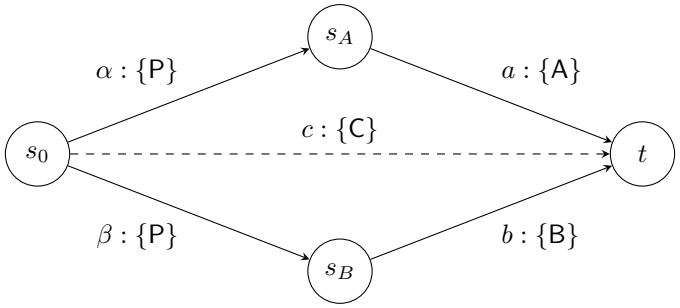

Figure 1: Two facility routes sharing permission P and an additional direct route supported by C. Labels give operation and requirement set. Self-loops with empty requirements are omitted.

Let $E = \{\mathsf{P}, \mathsf{A}, \mathsf{B}, \mathsf{C}\}$. The condition P is a shared permission. Conditions A and B support completion through the corresponding facilities. Condition C supports a direct route.

The same condition universe is used for all three catalogs. The baseline simply has no operative transition requiring C. This permits a comparison across the same scenario space. The full-availability scenario is $P = E$.

The social labels are stipulated. They make the structural comparison intelligible while leaving the actual nature of a permission, facility, or completion criterion open to a future application.

8.2 Complete Transition and Requirement Rules

The maps are total, and every state-specific requirement is specified. This subsection states the operative transitions and the self-loop convention.

Use

$$\begin{aligned} \alpha(s_0) &= s_A, & \mathrm{req}(\alpha, s_0) &= \{\mathsf{P}\}, \\ \beta(s_0) &= s_B, & \mathrm{req}(\beta, s_0) &= \{\mathsf{P}\}, \\ a(s_A) &= t, & \mathrm{req}(a, s_A) &= \{\mathsf{A}\}, \\ b(s_B) &= t, & \mathrm{req}(b, s_B) &= \{\mathsf{B}\}, \\ c(s_0) &= t, & \mathrm{req}(c, s_0) &= \{\mathsf{C}\}. \end{aligned} \tag{13}$$

Each listed operation fixes every other configuration and has empty requirements for those self-loops. Figure 1 displays the operative transitions.

The self-loop convention is part of the model. Applying a at the start performs a modeled operation with no state change or persistent requirement. It may be interpreted as an ineffective attempt only if that interpretation is justified. The word count includes it regardless of whether a substantive application would retain such steps.

The baseline alphabet is $\{\alpha, \beta, a, b\}$. The duplicate catalog adds α' with the same map and requirements as α. The direct-route catalog adds c to the baseline.

These variants separate label multiplicity from structural support. The duplicate preserves the same route requirement, while the direct operation supplies a route with a different condition. Neither variant changes the carrier or target.

8.3 Enumeration of Target Words

The complete target-word count depends on the horizon and the inclusion of self-loops. This subsection derives the counts, including the padding words in the direct-route catalog.

In the baseline, the first operative step must be α or β. From s_A, only a reaches the target in the remaining step. From s_B, only b does so. Thus the target words are αa and βb.

DRAFT

Table 3: Route catalogs at full availability with a two-step horizon

Catalog	Minimal target requirements	Inclusion-minimal disabling sets	κ
Baseline	$\{\mathsf{P},\mathsf{A}\}$, $\{\mathsf{P},\mathsf{B}\}$	$\{\mathsf{P}\}$, $\{\mathsf{A},\mathsf{B}\}$	1
Duplicate label	Same family	Same family	1
Additional route	Baseline family and $\{\mathsf{C}\}$	$\{\mathsf{P},\mathsf{C}\}$, $\{\mathsf{A},\mathsf{B},\mathsf{C}\}$	2

The duplicate label gives one additional word, $\alpha' a$. Its trajectory and requirements match those of αa. No one-step operation in this catalog reaches t from s_0, so the complete target count is three.

With the direct route, c reaches the target in one step. After c, any of the five operations fixes t, giving $c\alpha, c\beta, ca, cb, cc$. The ineffective first steps a and b can also be followed by c, giving ac, bc. Together with αa and βb, the complete list contains ten words.

The total word catalogs, including the empty word, have sizes $1 + 4 + 4^2 = 21$ for the baseline and $1+5+5^2 = 31$ for either five-label variant. These counts concern represented words, not distinct effects or distinct social episodes.

The enumeration makes the granularity issue visible. Several direct-route words differ only by an ineffective prefix or a post-completion self-loop. Their multiplicity should not be mistaken for ten independent means of obtaining the target.

8.4 Requirement Families and Outcome Sets

The target words can now be reduced to their persistent requirements. This subsection derives the families and compares their full-availability outcomes.

The baseline words require

$$\mathscr{R}_0 = \big\{\{\mathsf{P},\mathsf{A}\},\{\mathsf{P},\mathsf{B}\}\big\}.$$

The duplicate word contributes the first set again, so the family remains unchanged.

Every direct-route word containing c and reaching the target within the horizon has requirement $\{\mathsf{C}\}$. Its other steps are self-loops with empty requirements. The enlarged minimal family is therefore

$$\mathscr{R}_1 = \big\{\{\mathsf{P},\mathsf{A}\},\{\mathsf{P},\mathsf{B}\},\{\mathsf{C}\}\big\}.$$

The members are incomparable under inclusion.

At full availability, all three catalogs reach the same four configurations within two steps. The empty word gives s_0, preparation gives the two stages, and a completion route gives t. The direct route adds support for an existing target.

The equality of outcome sets is restricted to the full scenario. If P is removed while C remains, the direct catalog can still reach t and the baseline cannot. The difference appears across scenarios.

Table 3 records the requirement and cut comparison. It keeps the outcome question distinct from the dependence question.

8.5 Derivation of the Minimal Disabling Sets

The hitting-set theorem gives the removal patterns for each catalog. This subsection derives their minimality and completeness.

For the baseline, a disabling set must meet both $\{\mathsf{P},\mathsf{A}\}$ and $\{\mathsf{P},\mathsf{B}\}$. If it contains P, the singleton $\{\mathsf{P}\}$ already disables both. An inclusion-minimal set of this kind is therefore exactly that singleton.

If a disabling set omits P, it must contain A to hit the first route and B to hit the second. The pair $\{\mathsf{A},\mathsf{B}\}$ is disabling and has no disabling proper subset. These alternatives exhaust the minimal sets.

The baseline therefore has minimum cardinality one. The duplicate catalog has the same availability function, so its disabling sets and minimum remain identical.

DRAFT

With the direct route, every disabling set must also contain C. Combining that requirement with the baseline analysis gives exactly $\{\mathsf{P}, \mathsf{C}\}$ and $\{\mathsf{A}, \mathsf{B}, \mathsf{C}\}$ as minimal disabling sets. The minimum cardinality is two.

The improvement is strict because the old one-condition cut $\{\mathsf{P}\}$ leaves the direct route available. The new route's distinct prerequisite performs the relevant work; its label alone would be insufficient.

8.6 Control and Resource Variants

In this resource variant, every operation that is a self-loop outside its specified active transition has zero consumption and zero production. The ledger charges stated below apply to the active preparation and completion transitions.

The same requirement family can have different actor-level and resource interpretations. This subsection adds two stipulated variants to show why those structures must be represented separately.

Suppose one actor can remove both P and C. In the direct-route catalog, that actor removes a minimum condition cut in one control action. Condition-level robustness is two, while actor-level robustness is one under this control map.

A different distribution of control can require several actors to produce the same removal. The route and requirement families remain unchanged. The difference belongs to the added control structure.

For a resource variant, suppose each preparation and completion consumes one unit without production. Each two-stage route then requires initial stock two. Suppose the direct route consumes one unit. It requires stock one.

Under full persistent conditions and initial stock one, the old routes are resource-infeasible while the direct route is feasible. This is a different source of advantage from removal robustness. The resource-ledger section derives the general prefix threshold used by this variant.

These additions are separate models with explicit new rules. Their consequences should not be attributed to the original persistent-only catalog without stating the enrichment.

8.7 State Compression and Operation Identity

The example also tests a tempting coarse state description. This subsection shows why merging the preparation stages can lose information required by the completion operations.

Suppose s_A and s_B are grouped into one stage class while t remains distinct. Operation a sends s_A to t but fixes s_B. The two representatives therefore have different successor classes.

The proposed quotient fails update compatibility. A current observation that reports only prepared can still be meaningful, but it cannot support the selected deterministic updates without additional memory.

Operation identity also affects counting. The labels α and α' are distinct words but share maps and requirements. A transformation-semigroup representation merges them. A historical record can retain them if their different occurrences or interpretations matter.

The example thus connects path feasibility with the broader category framework. State, operation, event, and process records have different identity criteria. A valid reduction for one task can discard information needed by another.

8.8 Verification and Interpretive Scope

The finite example permits exhaustive checking, while its social interpretation remains stipulated. This final subsection states the evidence provided by the calculations.

Enumeration of every catalog word against every condition scenario confirms the target and requirement counts. Independent finite family checks corroborate minimal-family reduction and the cut criterion, including empty-family and empty-requirement cases. The propositions provide the general conditional proofs.

DRAFT

The checks establish consistency of the specified construction. They do not validate an empirical claim about a provider, permission, or facility. A substantive application must determine whether conditions persist, whether the routes share hidden dependencies, and whether the operation granularity matches practice.

The illustration's contribution is the coordinated comparison. Outcome sets, word multiplicity, requirement families, removal cuts, actor control, and resource needs can vary independently under declared changes. Keeping those quantities separate makes the meaning of an added route more precise.

9 Resources and Prefix Requirements

This section extends the path analysis to consumable and replenishable resources. It defines a resource ledger, derives the minimum initial stock required by a fixed word, and gives counterexamples to final-balance and Boolean-set shortcuts. It then combines resource thresholds with persistent condition profiles under explicit independence assumptions. The extension preserves the path's order and makes the timing of consumption and production part of the model.

9.1 Persistent Conditions and Resource Quantities

A persistent condition remains available after use. A resource quantity can decrease or increase during execution. This subsection distinguishes their mathematical roles before constructing the ledger.

A Boolean token label records whether a condition is available. It does not count how many times a token can be spent. If two steps each consume one unit, initial availability of one unit permits only the first. The requirement union $\{\text{token}\}$ would lose the repeated demand unless a quantity were represented separately.

A resource coordinate can represent material stock, time remaining, or another quantity with an appropriate conservation and update interpretation. Different quantities need separate coordinates when their units or substitution rules differ. Adding them into one scalar requires a conversion model that the present construction leaves open.

Production also has a temporal role. A resource produced after a step can support later steps but may be unavailable for that step's own initial consumption. The update convention must state this order. The formula below uses consumption followed by production.

The nonresource state can still change through the original operation maps. Requirements and production can depend on that state, provided the relevant trajectory can be determined independently of the resource stock under the assumptions used to precompute the bound.

9.2 The Resource Ledger

The ledger tracks stock at every prefix. This subsection defines the vectors and execution rule that support the derivation.

Let $z_i = T_{b_i} z_{i-1}$ be a fixed nonresource trajectory for $w = b_1 \cdots b_k$. At step i, let $c_i, p_i \in \mathbb{N}^d$ be consumption and subsequent production vectors determined by (b_i, z_{i-1}). Start with $r_0 \in \mathbb{N}^d$.

The step is executable when $r_{i-1} \geq c_i$ coordinatewise. After consumption and production, the stock is

$$r_i = r_{i-1} - c_i + p_i. \tag{14}$$

Every coordinate must satisfy the availability test before the update occurs.

The fixed-trajectory assumption excludes additional feedback from resource quantities into the nonresource transition or the consumption and production rules. If low stock changes the procedure itself, the full state must include that dependence and the precomputed sequence may need revision.

The convention permits a step with zero consumption and positive production, or a step that consumes and replaces a resource. It also permits several resource coordinates to be consumed together. It leaves borrowing, simultaneous external replenishment, and uncertain yields outside the stated model.

DRAFT

9.3 Derivation of Minimum Initial Stock

The ledger can be converted into a threshold on initial stock for a fixed word. This subsection derives that threshold from every prefix constraint.

For resource coordinate j, define

$$L_j(w) = \max\left(0,\ \max_{1\le i\le k}\left[\sum_{t\le i} c_{t,j} - \sum_{t<i} p_{t,j}\right]\right), \qquad L_j(\epsilon) = 0. \tag{15}$$

Write $L(w) = (L_1(w), \dots, L_d(w))$.

Proposition 9.1 (Prefix resource threshold)**.** *Under the fixed-trajectory ledger assumptions, the word w is resource-feasible from r_0 if and only if $r_0 \ge L(w)$ coordinatewise.*

Proof. After successful execution of the first $i-1$ steps, repeated substitution in Equation (14) gives

$$r_{i-1} = r_0 + \sum_{t<i}(p_t - c_t).$$

The step-i feasibility condition is therefore equivalent to

$$r_0 \ge \sum_{t\le i} c_t - \sum_{t<i} p_t.$$

Requiring this inequality at every step and in every coordinate, together with $r_0 \ge 0$, gives exactly Equation (15). This proves necessity.

For sufficiency, assume $r_0 \ge L(w)$. The step-one bound guarantees its consumption. Suppose all earlier steps have been executed. Their ledger sum has the displayed form, and the step-i bound ensures that the current stock covers c_i. Thus the next step can be executed. Induction completes the word. The empty word requires no consumption and satisfies the zero bound. □

The production sum stops before the current step because that step's production occurs after its consumption. Including p_i would permit a step to finance its own initial requirement through resources available only afterward. That would implement a different timing convention.

The maximum retains the most demanding prefix. Later production can reduce the stock needed for subsequent steps, but it cannot retrospectively finance an earlier deficit. The formula makes this temporal asymmetry explicit.

9.4 Counterexamples to Final-Balance Reasoning

A nonnegative final balance can conceal an infeasible prefix. This subsection gives two small examples and relates them to the information lost by compression.

Consider one step consuming one unit and then producing one unit. Its net change is zero. From initial stock zero, a final-balance calculation returns zero and may appear acceptable. The step nevertheless fails its initial consumption test. Equation (15) gives $L = 1$, correctly retaining the requirement.

Now let an earning step e have consumption zero and production one, and a working step w have consumption one and production zero. Both words ew and we have net change zero. The first is feasible from zero: earning supplies the unit before work consumes it. The second requires one initial unit.

The threshold calculation gives $L(ew) = 0$ and $L(we) = 1$. The difference arises entirely from order. A summary containing only total consumption and total production would merge the two words and lose feasibility.

Two successive spending steps provide the multiplicity example. Each consumes one and produces zero, so the successive prefix requirements are one and two. The word requires initial stock two. A persistent Boolean label cannot represent this threshold.

These examples show why an endpoint resource balance, a net-change vector, and a prefix feasibility profile are different records. Each can be useful for a specified task. Only the last supports the complete execution test under the current assumptions.

DRAFT

9.5 Resource Dimensions and Incomparable Routes

Different routes can require different resource combinations. This subsection explains why one componentwise minimum can fail to represent the target's resource requirements.

Suppose one target route requires initial vector $(2, 0)$ and another requires $(0, 2)$. The first uses the first resource, and the second uses the second. Neither threshold dominates the other coordinatewise.

The componentwise infimum is $(0, 0)$, but neither route is feasible from that stock. Taking the lowest requirement in each coordinate has combined advantages from different routes into a fictitious route. The common path witness has been lost.

A target's resource requirement can therefore be a family of incomparable vectors. Availability means that the initial stock dominates at least one member of that family. A single least vector exists only in special cases where one profile is below all relevant alternatives.

This distinction is important for a generative interpretation of possibility. Several routes can provide different ways of reaching the same outcome without being reducible to one aggregate capacity. The family retains the tradeoffs that a scalar or coordinatewise minimum can conceal.

A valuation can later rank the resource combinations under declared costs or preferences. That ranking is additional to the feasibility relation. It should retain the fact that different thresholds belong to different executable paths.

9.6 Joint Persistent and Resource Profiles

Persistent prerequisites and consumable resources can be combined when their rules refer to a common fixed trajectory. This subsection derives that joint representation and its minimal-profile reduction.

For each target word, retain the pair

$$\big(R_x(w), L(w)\big).$$

At scenario P and initial stock r_0, the word is feasible for both parts exactly when

$$R_x(w) \subseteq P, \qquad L(w) \leq r_0. \tag{16}$$

The conjunction follows from the persistent requirement theorem and Proposition 9.1.

The transfer assumes that the persistent requirements, resource consumptions, and productions are determined by the same fixed nonresource trajectory. It also assumes that satisfying one part introduces no unrepresented change in the other. Otherwise the two separate tests may fail to describe one common execution.

Order profiles by $(R, L) \preceq (R', L')$ when $R \subseteq R'$ and $L \leq L'$ coordinatewise. Every scenario and stock satisfying the second profile also satisfy the first. Thus a larger profile is redundant for Boolean target availability when a smaller target profile exists.

For a finite target-word catalog, finite descent shows that the minimal profiles preserve availability. The proof parallels the minimal requirement-family theorem: each successful profile contains a minimal one under the product order, and every minimal profile belongs to the original family.

The reduction remains query-specific. A dominated profile can belong to a faster or historically important route. The minimal family preserves the existence of a feasible target path under the declared resources and conditions, while other path properties need separate retention.

9.7 Changing Production and Resource-Dependent Dynamics

The threshold theorem relies on a precomputed sequence. This subsection identifies cases in which a richer dynamic model is needed.

Production can depend on the stock available during execution. A procedure might yield more output when supplied with additional resources or change its consumption under shortage. The vectors c_i, p_i then depend on the evolving ledger, so the fixed-word threshold formula requires renewed analysis.

DRAFT

An operation can also have several modes selected according to available resources. Representing them by one fixed map can conceal the choice. Separate labels or a state-dependent decision rule can make the alternatives explicit.

External replenishment introduces timing and coordination. A resource arriving after a blocked step can enable a later retry, while an absorbing blocked state would rule that recovery out. The model should represent the relevant waiting and recovery operations when they belong to the inquiry.

Borrowing changes the admissible stock domain and the meaning of a negative balance. A credit limit can be encoded as a lower bound, but repayment, interest, and enforcement require their own rules. The present nonnegative ledger supplies a baseline whose extensions should remain explicit.

Stochastic yields require a distribution or a set of possible productions. A path can then be feasible under some realizations and fail under others. A deterministic threshold would answer only a selected scenario or bound. Probability and robust execution need separate constructions.

9.8 Interpretation of Resource Feasibility

The resource extension adds a temporal account of what a path needs before each step. This final subsection states its conceptual contribution and its limits.

The threshold $L(w)$ is derived from a specified word and timing convention. It provides a minimum initial stock for that execution. It is not a general measure of a person's capacity or an institution's value.

The distinction between stock and persistent permission also remains important. An actor can possess resources while lacking access, or possess authorization while lacking the resources needed to use it. The joint profile represents both conditions without identifying them.

Resource feasibility still leaves selection and information open. An actor may have enough stock for one route while lacking knowledge of which route will succeed. The following analysis of path selection addresses that additional requirement.

The resulting framework preserves a sequence of questions: which outcome has a route, which conditions and resources make that route executable, who controls those conditions, and what information permits a successful choice. Each question adds a declared structure whose consequences can be derived and challenged.

10 Route Restoration and Feasibility under Uncertainty

This section develops the additions required to restore target availability and the conditions for selecting a route under uncertain scenarios. It begins with fixed persistent and resource profiles, derives a family of minimal deficits, and then distinguishes scenario-wise availability from one route that works across the retained alternatives. Information and intervention timing determine the practical scope of the results.

The removal analysis identifies conditions whose loss disables every target route. Restoration asks a different question: given the current conditions and resources, which additions would enable at least one route? The two questions use the same route profiles while selecting different comparisons. A restoration proposal also needs an account of who can supply the additions and how the additions affect the underlying dynamics.

10.1 Fixed Profiles and Additive Interventions

Let E be a finite universe of persistent conditions and let $\mathcal{G}$ be a finite family of target-word profiles (R, L), where $R \subseteq E$ and $L \in \mathbb{N}^d$. The profile records persistent requirements and the initial resource threshold derived from the execution prefixes. At scenario (P, r), it is feasible exactly when

$$R \subseteq P, \qquad L \leq r$$

coordinatewise.

DRAFT

The nonresource initial state, operation meanings, target, and trajectory-dependent thresholds are fixed for this comparison. An additive intervention (D, δ) changes the scenario to

$$(P \cup D, r + \delta), \qquad D \subseteq E \setminus P, \quad \delta \in \mathbb{N}^d.$$

The intervention supplies conditions and initial stock while leaving the route profiles unchanged.

This is a restrictive and useful starting point. A grant can also change an operation's meaning, induce a different response, or arrive after the step that needed it. Those cases require another model or a recomputation of the profiles. The present construction isolates what follows when the additions have the displayed effect.

10.2 Deficits for a Target Route

For a profile (R, L), define its deficit at (P, r) by

$$\Delta_{P,r}(R, L) = \big(R \setminus P, (L - r)_+\big), \tag{17}$$

where the positive part is coordinatewise:

$$((L - r)_+)_j = \max\{0, L_j - r_j\}.$$

The first component lists missing persistent conditions. The second gives additional initial stock required in each resource dimension.

Order interventions by

$$(D_1, \delta_1) \preceq (D_2, \delta_2) \quad \Longleftrightarrow \quad D_1 \subseteq D_2, \quad \delta_1 \leq \delta_2.$$

The order compares sufficiency of additions. A larger element supplies every condition and unit supplied by a smaller one. It leaves costs and evaluative preferences open.

Proposition 10.1 (Restoration by deficit domination)**.** *Under the fixed-profile assumptions, an additive intervention enables a target route if and only if it dominates $\Delta_{P,r}(R, L)$ for some $(R, L) \in \mathcal{G}$.*

Proof. A profile succeeds after the intervention precisely when $R \subseteq P \cup D$ and $L \leq r + \delta$. The set inclusion is equivalent to $R \setminus P \subseteq D$. For each coordinate, the inequality $L_j \leq r_j + \delta_j$, together with $\delta_j \geq 0$, is equivalent to $\delta_j \geq \max\{0, L_j - r_j\}$. Combining the conditions gives domination of the deficit. Existential quantification over the target profiles yields the availability statement. □

The proposition turns a route question into an intervention question under explicit assumptions. It can identify several alternative additions, each sufficient through a different route. The identity of the route remains available in the full profile record when the inquiry needs to explain how the addition enables the target.

10.3 Minimal Restoration Profiles and Incomparable Additions

A finite family of deficits can be reduced to its minimal members under $\preceq$. If an intervention dominates a nonminimal deficit, finite descent supplies a minimal deficit below it. The intervention therefore dominates that minimal member as well. Conversely, each minimal deficit belongs to the original family. Thus the minimal family preserves exactly which additive interventions enable a target route.

The resulting family can contain incomparable options. Suppose two target routes have resource thresholds $(2, 0)$ and $(0, 2)$, and the initial stock is $(0, 0)$. One route needs two units of the first resource; the other needs two units of the second. The minimum sufficient additions form a two-element family.

Taking the coordinatewise minimum of the two vectors gives $(0, 0)$, which enables neither route. A vector of separately small coordinates can therefore fail to represent any sufficient intervention. The family retains the combinations of requirements that belong together.

DRAFT

The same point applies to persistent conditions. At $P = \{a, b\}$, two routes with requirements $\{p, a\}$ and $\{p, b\}$ both have deficit $\{p\}$. Supplying p restores both. Duplicating either route changes its number of witnesses while leaving the missing condition unchanged.

If a direct route requiring c is added, another restoration option appears: supplying c. The minimal deficit family becomes $\{\{p\}, \{c\}\}$ for this persistent-only example. The options can differ in provider, timing, cost, or political consequences. Those comparisons require information beyond set inclusion.

A zero deficit has a clear boundary meaning. It indicates that a route is already available under the represented scenario. An empty route-profile family has the opposite status: additive conditions and stock cannot enable a route absent from the fixed catalog. A proposal to create a new mechanism or alter the target changes the comparison and requires a revised model.

10.4 Uncertain Scenarios and a Common Route

An actor or investigator can be uncertain about the conditions and resources actually available. Let K be a nonempty finite set of scenarios $s = (P_s, r_s)$ compatible with the retained information. Define

$$P_\cap = \bigcap_{s \in K} P_s, \qquad (r_\wedge)_j = \min_{s \in K} (r_s)_j. \tag{18}$$

The constructed pair $(P_\cap, r_\wedge)$ retains the conditions and stock amounts guaranteed across all scenarios.

The pair need not itself be a member of K. Different resource minima can occur in different scenarios. Its role is to test one fixed profile against every scenario, which is possible because the profile's conditions are conjunctive and coordinatewise.

Proposition 10.2 (Common-route feasibility)**.** *A fixed profile (R, L) is feasible in every scenario of K if and only if*

$$R \subseteq P_\cap, \qquad L \leq r_\wedge.$$

A common target route exists exactly when some member of $\mathcal{G}$ satisfies these conditions.

Proof. Universal feasibility requires $R \subseteq P_s$ for every $s \in K$. This is equivalent to $R \subseteq \bigcap_s P_s$. For each coordinate, the requirement $L_j \leq (r_s)_j$ for every scenario is equivalent to $L_j \leq \min_s (r_s)_j$. Combining the conditions proves the fixed-profile equivalence. Choosing one such profile proves the common-route statement. □

The result preserves the quantifier order. One profile is chosen to work across the retained scenarios. A weaker statement allows the chosen route to vary with the actual scenario. That weaker availability statement can hold even when the available information supplies insufficient grounds for choosing the appropriate route.

10.5 Scenario-Wise Availability and Information

Consider two routes with requirements $\{a\}$ and $\{b\}$, and two scenarios $P_1 = \{a\}$, $P_2 = \{b\}$. Resource requirements are zero. Each scenario has a target route. Their intersection is empty, so neither route is feasible in both.

The difference can be written as

$$\forall s \in K \; \exists (R, L) \in \mathcal{G} \quad \text{versus} \quad \exists (R, L) \in \mathcal{G} \; \forall s \in K.$$

The first statement describes availability at each scenario. The second supports a fixed choice before the scenario is resolved.

A common-route failure can therefore coexist with sufficient conditions in the actual scenario. The difficulty can lie in information or selection. This is distinct from a scenario in which every route lacks a required resource.

DRAFT

Suppose the actor receives an observation that partitions K into nonempty cells. A rule choosing one fixed route per observed cell succeeds exactly when each cell admits a common feasible profile. The criterion in Proposition 10.2 applies separately to each cell.

Refining the information partition can help. Each new cell lies within an old cell, so a route feasible throughout the old cell remains feasible throughout each of its subcells. The actor can also ignore the additional information. Consequently additional usable information weakly expands the set of successful selection rules under these assumptions.

The word *usable* carries a practical requirement. An observation must be available before the relevant choice, reliably interpreted, and retained by the selecting actor or procedure. An investigator's later knowledge does not by itself supply the actor's earlier information.

10.6 Additions Chosen before Uncertainty Is Resolved

A restoration intervention can itself be selected before the scenario is known. Suppose a fixed grant $D \subseteq E$ and stock addition δ apply in every scenario. The grant can include a condition already present in some scenarios.

The intersection of the resulting condition sets is

$$\bigcap_{s \in K} (P_s \cup D) = \left(\bigcap_{s \in K} P_s \right) \cup D.$$

An element belongs to every enlarged scenario exactly when it is supplied by the common grant or belongs to every original scenario. For resources,

$$\min_{s \in K} \big((r_s)_j + \delta_j \big) = (r_\wedge)_j + \delta_j.$$

The scenario-independent addition can be taken outside the minimum.

These identities permit the deficit construction to be applied at $(P_\cap, r_\wedge)$. A single intervention enables a common route across all scenarios if and only if it dominates a profile deficit computed at that constructed pair.

The interpretation is specific. The same addition is chosen before uncertainty is resolved, and one route must then be feasible throughout the remaining scenario set. A policy that observes the scenario and chooses different routes can require smaller additions. A policy that also chooses different grants after observation answers a further question.

This distinction can matter in the design of support arrangements. A general provision intended to work across several circumstances has different requirements from a provision tailored after a diagnosis. The formal comparison identifies those requirements under the model's fixed profiles. Evaluation of the alternatives requires their costs, effects, and institutional context.

10.7 Timing, Provision Authority, and Intervention Effects

The stock addition in the restoration theorem is available before execution begins. A resource arriving after a blocked prefix can have a different effect. If waiting or retrying is permitted, the word and its threshold must include the relevant operations. The formula should follow the execution model's timing.

Persistent conditions also require an account of provision. A missing authorization can be represented by a label, while the authority to supply it may lie with a particular actor or procedure. The deficit family identifies what is missing. A control or provision map is needed to identify which interventions are available to which actors.

Supplying a condition can also change other conditions. A new authorization may trigger a review, alter obligations, or change an operation's resource requirements. Such effects violate the assumption that the profiles remain fixed. The appropriate response is to model the intervention and recompute its route consequences.

DRAFT

An addition can carry costs that are absent from the target availability question. Two minimal deficits can differ in the burdens they impose, the subjects who bear them, and the future dependencies they create. Minimality under $\preceq$ records sufficiency of additions; it leaves their evaluation to an explicit standard.

10.8 Interpretation of Restoration and Robust Access

The restoration construction complements the analysis of dependency and disabling. It identifies sufficient additions from the current scenario and preserves alternative combinations when the requirements are incomparable. The uncertainty construction then asks which additions and route choices remain effective across the scenarios retained by the available information.

These results help distinguish several sources of an unrealized outcome. A target route can be absent from the catalog, blocked by missing conditions, infeasible because of an intermediate resource requirement, or difficult to select because the actor cannot distinguish the relevant scenarios. Each source calls for a different revision or intervention.

The framework also preserves the difference between possessing a skill and having the conditions needed to realize its use. A missing condition can block the route through which a retained competence would contribute. Restoring that condition can change practical availability while leaving the competence itself unchanged.

A substantive account must still establish the correspondence between the route model and the activity being studied. The formal results specify what follows from fixed profiles, valid scenario information, and declared interventions. Their value lies in making those commitments and their consequences inspectable.

11 Horizons and Progress Toward a Target

This section clarifies comparisons of possibility before and after an operation. It derives a horizon-shift inclusion, defines the remaining distance to a target, and reconciles progress toward that target with contraction of unrestricted reachability in a fixed semigroup. It also examines how declaring a composite to be a primitive changes step counts. The aim is to identify the comparison budget behind a claim of generativity or expanded opportunity.

11.1 Remaining and Total Execution Budgets

A successor configuration has already been reached through some activity. Comparing it with the original seed under equal remaining budgets can therefore differ from comparing equal total budgets. This subsection states that distinction before deriving the inclusion.

Let $\mathrm{Reach}_h^A(x)$ be the outcome set of prefix-feasible words of length at most h from x. The budget counts operations in the declared primitive alphabet. The empty word contributes the current outcome.

Suppose c is executable at x. A path of length at most h from $T_c x$ uses at most $h+1$ steps when the initial operation is counted. Comparing $\mathrm{Reach}_h^A(T_c x)$ with $\mathrm{Reach}_h^A(x)$ resets the remaining budget after that operation.

Such a comparison can be useful. It asks what is available from the new situation with a fresh budget. It should not be described as a comparison of equal total execution effort unless the initial step is accounted for separately.

The distinction also applies to time and resources, although those budgets need their own measurements. A one-step preparation can be brief or extended. The primitive count supplies one formal comparison, whose empirical interpretation requires a definition of the step.

11.2 The Horizon-Shift Inclusion

Concatenation provides an exact relation between the two budgets. This subsection proves it with the prefix conditions explicit.

DRAFT

Proposition 11.1 (Horizon shift after an executable step)**.** *For a fixed availability rule and outcome map, if c is executable at x, then*

$$\mathrm{Reach}_h^A(T_c x) \subseteq \mathrm{Reach}_{h+1}^A(x). \tag{19}$$

Proof. Take an outcome in the left-hand set and choose a feasible witnessing word w from $T_c x$ with length at most h. The word cw begins with the executable step c. Every later prefix reaches the same state as the corresponding prefix of w from $T_c x$, so its availability test remains satisfied. The endpoint is the same and the length is at most $h + 1$. Thus the outcome belongs to the right-hand set. □

The proof retains the same availability and transition rules after the first step. If the operation changes the rules themselves, those changes must be included in the configuration or represented through an explicitly enlarged system.

The inclusion also covers the empty continuation. The successor's current outcome is reached from the original seed through the one-step word c. This boundary case is part of the budget comparison.

In the shared-permission example, the target is unavailable within one step from s_0 and available within one step from s_A. Preparation has already used one step. The target was available within two steps from s_0. The two statements are consistent.

11.3 Remaining Distance to a Target

A target-specific distance provides a precise sense in which an operation can make progress. This subsection defines it and derives the shortest-path consequence.

Let $d_A(x, v)$ be the least length of a feasible word reaching v, with value infinity if no such word exists. When a target word exists, its lengths form a nonempty subset of the natural numbers, so a least length exists.

For an executable c, the prefix argument gives

$$d_A(x, v) \leq 1 + d_A(T_c x, v). \tag{20}$$

If the successor distance is finite, prepend c to a shortest witness. If it is infinite, the extended inequality places no additional restriction.

Suppose c begins a shortest positive-length target word of length L. Its suffix is a feasible target word of length $L - 1$ from the successor. A shorter successor word would prepend to a word shorter than L from the original seed, which is impossible. Therefore

$$d_A(T_c x, v) = d_A(x, v) - 1.$$

This result supplies a limited progress measure. It concerns one target and one primitive length convention. An operation that moves toward one target can move away from another or remove other options.

Distance also leaves cost and value open. A shorter word can contain a more burdensome operation, and the selected target may be unwanted by some participants. The measure should retain its stated purpose.

11.4 Unrestricted Reachability in a Fixed Semigroup

The source framework's global reachability result concerns all composite mechanisms without a finite budget. This subsection gives the proof and distinguishes it from target progress.

For the fixed unrestricted semigroup, define

$$\mathrm{Reach}^*(x) = \{x\} \cup \{mx : m \in M\}.$$

For $c \in M$,

$$\mathrm{Reach}^*(cx) \subseteq \mathrm{Reach}^*(x). \tag{21}$$

DRAFT

A point in the successor set is either cx or $m(cx) = (mc)x$. Both already belong to the original set.

The inclusion follows from closure of the semigroup under composition. The original unrestricted repertoire already contains every sequence beginning with c and followed by another mechanism. Moving to the successor cannot create a point outside that preexisting global image.

This result can coexist with a shorter remaining path to a target. The global set asks whether some composite reaches the point. The distance asks how many declared primitive steps remain. The quantities have different information content.

The inclusion also can be strict. An irreversible operation can remove access to earlier alternatives. That contraction can occur while the chosen target becomes closer. A generative interpretation should state which aspect of possibility it means.

11.5 Composite Shortcuts and Primitive Granularity

Changing the alphabet can shorten a word without changing the generated semigroup. This subsection gives a finite example and examines its interpretive limit.

On $\{0, 1, 2\}$, let a send zero to one and fix the other points. Let b send one to two and fix the other points. All operations are available. From seed zero, the target two requires two steps under alphabet $\{a, b\}$.

Now add $c = b \circ a$ as a declared unit-cost primitive. The target is reached in one step by c. The generated transformation semigroup is unchanged because c was already a product of the old generators.

The global outcome set is therefore unchanged, while primitive-word distance decreases from two to one. This is a mathematical consequence of the changed alphabet and cost convention.

A new label alone does not establish a reduction in physical time, resources, or coordination. If performing c still requires the old two-step procedure, counting it as one operation changes the description without proving an execution improvement.

A genuine integrated procedure can justify a new primitive and a different cost. The empirical claim then needs evidence about how the procedure is performed. The example makes that additional premise visible.

11.6 Changing Mechanisms and Changing Descriptions

Several different changes can be called an expansion of possibility. This subsection separates them through the inputs they alter.

Adding a mechanism outside the old semigroup changes the global repertoire and can enlarge the reachable set. Adding a new primitive that is already an old composite can change bounded distance while preserving global reachability.

Changing availability can make an old mechanism practically accessible or remove a previous route. The effect concerns the admissible path system. It should be distinguished from enlarging the unrestricted abstract action.

Refining the outcome quotient can distinguish configurations previously merged. The number of outcome classes can increase without a new raw configuration being generated. The change concerns the descriptive regime.

Changing the carrier can introduce new kinds of represented configurations or new participants. Comparisons across such models need an alignment map or an explicit account of correspondence. Set inclusion inside one fixed carrier does not supply that comparison automatically.

A comprehensive generativity claim should identify which change is being asserted. The framework permits several forms of development, while the proof appropriate to one should not be transferred to another through a shared word.

DRAFT

11.7 Interpretive Role of Bounded Progress

The horizon analysis supplies a constructive interpretation of progress that remains compatible with the fixed-base theorem. This final subsection states that contribution.

An operation can reduce the remaining effort to a selected target, establish a needed intermediate condition, or make an old route available under a bounded budget. These are meaningful changes even when the unrestricted global outcome set remains fixed.

The framework also preserves the possibility of tradeoffs. Progress toward one target can consume resources, remove alternatives, or change control over later steps. A single distance cannot summarize all those effects.

The value of the formal comparison lies in its explicit budget and target. It permits a precise statement of what became closer, accessible, or distinguishable. Further evaluation can then examine the consequences that the chosen measure leaves open.

12 Unbounded Route Families and Finite Availability Bases

A horizon bounds the number of operations considered in a route search. The previous sections used that bound to obtain finite families and to compare progress toward a target. This section studies the distinct question of whether an unbounded family of finite routes can nevertheless have a finite availability representation. The answer depends on the ordering of requirements and on the resource domain. The argument first establishes a finite-basis result for integer resources, then examines discovery, horizon dependence, continuous resource quantities, and the historical information that the basis omits.

12.1 Profile Domains and the Availability Query

Fix a finite permission vocabulary E and a finite number d of resource coordinates. Resource stocks and requirements take values in $\mathbb{N}^d$, including zero. The elements of E identify persistent positive prerequisites under the assumptions already stated. Every target-reaching word has a joint profile (R, L), where $R \subseteq E$ records its permissions and $L \in \mathbb{N}^d$ records the minimum stock needed to execute its consumption and production ledger. The collection of these profiles is

$$\mathscr{G} \subseteq \mathcal{P}(E) \times \mathbb{N}^d.$$

There can be infinitely many target-reaching words, and there can also be infinitely many distinct profiles. The operation alphabet, number of states, and method of enumerating words require their own declarations; the order argument below concerns the resulting profile family.

The profile order is the same order used for finite joint families:

$$(R, L) \preceq (R', L') \quad \iff \quad R \subseteq R' \text{ and } L \leq L'. \tag{22}$$

The vector inequality is coordinatewise. A smaller profile demands a subset of the permissions and at most the stock in every coordinate. This interpretation licenses its substitution in a Boolean availability query. It leaves open whether a route with that profile takes more time, requires a less attractive sequence of actions, or has a different formation history.

For a scenario (P, r), availability is

$$\Phi_{\mathscr{G}}(P, r) = \mathbf{1}\{\exists (R, L) \in \mathscr{G} : R \subseteq P,\ L \leq r\}. \tag{23}$$

Every quantified profile must arise from an actual word admitted by the model. In particular, taking an infimum of profile coordinates and inserting the result into $\mathscr{G}$ requires a separate argument that the new profile is realizable. The finite-family analysis already showed why coordinatewise aggregation can combine requirements from different routes. An infinite family introduces the additional possibility that a limiting requirement has no witness at all.

DRAFT

12.2 Finite Bases with Integer Resources

The mathematical antecedent is Dickson's lemma for finite-dimensional vectors of natural numbers. Sustik states the equivalent monomial divisibility formulation and outlines a classical subsequence argument in his presentation on its formal verification [5]. The following proof reconstructs the order argument needed for joint route profiles. The established lemma supplies the mathematical basis; the permission and resource interpretation supplies the present application.

Proposition 12.1 (Finite basis for integer route profiles)**.** *Let E and d be finite. For every $\mathscr{G} \subseteq \mathcal{P}(E) \times \mathbb{N}^d$, the family* $\mathrm{Min}(\mathscr{G})$ *is finite. Every member of $\mathscr{G}$ dominates a member of* $\mathrm{Min}(\mathscr{G})$*. Consequently,*

$$\Phi_{\mathscr{G}}(P, r) = \Phi_{\mathrm{Min}(\mathscr{G})}(P, r)$$

for every scenario (P, r).

Proof. First consider an infinite sequence of natural numbers. If some value occurs infinitely often, the positions at which it occurs form an infinite constant subsequence. Otherwise, each value occurs only finitely often. For every fixed bound, only finitely many sequence positions then have values below that bound. Starting at any selected position, a later position with a larger value can therefore be selected. Repetition produces an infinite strictly increasing subsequence. In either case there is an infinite nondecreasing subsequence.

Now consider an infinite sequence in $\mathbb{N}^d$. Select an infinite subsequence nondecreasing in the first coordinate. Within that subsequence select one nondecreasing in the second coordinate, and continue through the finite set of coordinates. Passing to a subsequence preserves the inequalities obtained for earlier coordinates. The final subsequence is nondecreasing in every coordinate, so it contains an earlier vector below a later vector. This is the required sequence form of Dickson's lemma.

Suppose $\mathrm{Min}(\mathscr{G})$ were infinite. Since E is finite, only finitely many permission sets R are possible. Infinitely many distinct minimal profiles would share one permission set. Applying the preceding sequence result to their resource vectors would give two distinct profiles ordered by $\preceq$. The larger would then have a strictly smaller member of $\mathscr{G}$, contradicting minimality. Thus the minimal family is finite.

For coverage, fix $g = (R, L) \in \mathscr{G}$. The profiles below g have permission sets drawn from the finite family $\mathcal{P}(R)$ and resource vectors in the finite integer box $\{L' : 0 \le L' \le L\}$. Hence $\mathscr{G} \cap \{g' : g' \preceq g\}$ is finite and nonempty. Choose a minimal member m of that set. Any member of $\mathscr{G}$ strictly below m would also lie below g, contradicting this choice. Therefore $m \in \mathrm{Min}(\mathscr{G})$.

If (P, r) enables g, the subset and coordinate inequalities imply that it enables $m \preceq g$. Conversely, each minimal profile belongs to $\mathscr{G}$ and has the same kind of route witness as every other member. These implications establish equality of availability. For $\mathscr{G} = \varnothing$, both availability functions are zero and the minimal family is empty. □

The two parts of the proof perform different tasks. Excluding an infinite antichain establishes that the minimal family is finite. Finding a minimal profile below each given profile establishes that the family covers every available scenario. Both are needed for the availability conclusion. A partially ordered set can have an empty minimal family even though it contains many elements, so finiteness of the minimal set by itself would be too weak.

When $d = 0$, the profile space reduces to $\mathcal{P}(E)$, and the result follows directly from its finiteness. Thus even infinitely many permission-only routes admit a finite basis under a finite vocabulary. Integer resources enlarge the profile space to an infinite set while retaining the relevant finite-basis property. The extension changes the sufficient assumptions of the earlier result in a precise way: the route family can be unbounded, provided the permission vocabulary and integer resource dimension remain fixed and finite.

12.3 Basis Existence and Route Discovery

Proposition 12.1 is a statement about the complete profile family. A researcher often has only a partial enumeration of words, a collection of observed episodes, or a bounded search. These sources yield a

DRAFT

family $\mathscr{G}_{\text{seen}} \subseteq \mathscr{G}$. Its minimal profiles give an exact answer for the seen family. Establishing that they also give the unrestricted answer requires a completeness argument.

A newly discovered route can have a profile below an existing minimal profile and remove that profile from the current basis. It can also supply an incomparable profile that enables previously unavailable scenarios. A long period during which enumeration discovers only dominated profiles therefore leaves two possibilities open: the current basis is complete, or a later route will improve it. The existence theorem does not itself decide which possibility applies at the current stage of a search.

The distinction can be expressed directly. A finite observed basis $\mathscr{B}$ is complete exactly when every profile in the full family dominates some member of $\mathscr{B}$, provided $\mathscr{B} \subseteq \mathscr{G}$. Verification of that universal condition depends on the transition structure and on the method used to describe the unbounded family. In a suitable finite model an invariant or a separate reachability calculation may discharge it. The present order theorem leaves such methods to the model-specific analysis. A declaration of computational difficulty would likewise require an explicitly defined input representation and decision problem.

There is nevertheless a useful conditional stabilization statement. Since the complete basis is finite and each of its elements has a finite word witness, an enumeration that eventually includes every finite word will eventually include witnesses for all basis elements. After that stage its minimal profiles equal the complete basis. The proof gives eventual inclusion under exhaustive enumeration; it supplies the observer with no general certificate that a particular stage is already the final one.

For an empirical route catalog, exhaustive enumeration is usually an additional assumption about the model, independent of the number of episodes observed. A long historical archive can repeatedly document the same requirements while missing an institutionally prohibited or rarely attempted alternative. Conversely, an analytically generated word may fail to describe any practicable operation in the domain. The finite basis summarizes the family supplied to it, so the warrant for that family remains visible in the resulting availability claim.

12.4 A Resource-Accumulation Example

A small ledger makes the horizon issue explicit. Fix an integer $N \geq 1$. The nonresource state is either pending or done. At pending, an operation called earn produces one resource unit without consuming any; an operation called work consumes N units and reaches done. At done, both operations are zero-cost self-loops. The permission vocabulary is empty. Resource production and consumption follow the prefix convention used earlier, and the target is done.

Every word that first reaches the target has a prefix earn^{k}work for some $k \geq 0$. Its initial stock requirement is

$$L(k) = \max(0, N - k).$$

The earnings occur before the consuming operation. Hence each earlier earn reduces the necessary initial stock by one until the stock requirement reaches zero. Operations executed after done preserve the target and add no cost under the declared self-loop convention.

At horizon $h \geq 1$, a first target hit can contain at most $h - 1$ earnings. The least stock among target-reaching words is therefore

$$L_h = \min_{0 \leq k \leq h-1} \max(0, N - k) = \max(0, N - h + 1). \tag{24}$$

At $h = 0$, the initial pending state supplies no target word. For every $h \geq 1$, a scenario with initial stock r enables the target within the horizon precisely when $r \geq L_h$.

With unrestricted finite words, earn^{N}work has profile $(\varnothing, 0)$. This is the unique minimal profile, because its resource requirement is the least possible value and its permission requirement is empty. The unrestricted basis is thus particularly small: it consists of one member. Yet the first zero-stock witness has length $N + 1$, which grows with the stipulated consumption requirement.

The example separates compact description from short realization. A one-element basis can stand for a route that requires many successive preparatory operations. Describing the target as available at zero

DRAFT

initial stock is exact for unrestricted finite execution. Describing it as available before a short deadline can be false. Both statements can concern the same actors, target, and operations while differing only in the horizon built into the query.

In a social application, the production premise would require its own interpretation. Earn might stand for the accumulation of budget units, training credits, or another explicitly countable resource. Each interpretation raises evidence questions concerning renewal rates, upper bounds, access to the earning operation, and changes in the environment during accumulation. The mathematical example stipulates those features so that the horizon distinction is transparent. A substantive model should record any constraint that changes the sequence of feasible prefixes.

12.5 Continuous Quantities and Attainment

Discreteness is an operative assumption in the proof. To see its role, consider real resource requirements with two coordinates and the family

$$\mathscr{G}_{\mathbb{R}} = \{(\varnothing, (t, 1-t)) : 0 \le t \le 1\}.$$

For $(t, 1-t) \le (s, 1-s)$, the first coordinate gives $t \le s$ while the second gives $t \ge s$. Thus $t = s$. Every member of the family is minimal, and there are infinitely many minimal profiles. The finite permission vocabulary does not repair the loss of the integer-vector assumption.

This family also clarifies the information carried by a tradeoff. A scenario can support one balance of the two resources while failing to support another. Replacing the entire family with a finite sample loses some exact threshold answers. A discretization can be useful, but its error and the domain on which it is adequate become additional questions. A finite approximation should retain its approximation status in both the formula and its interpretation.

Attainment presents a distinct problem even with one real coordinate. Let the route requirements be all numbers in $(0, 1)$. Each requirement t has a smaller admitted requirement $t/2$, so the family has no minimal member. Availability holds for every stock $r > 0$, because some admitted requirement lies below r. It fails at $r = 0$. The infimum of the family is zero, but zero belongs to no route profile.

Substituting that infimum for the original family changes the availability boundary. The substitute asserts feasibility at zero stock, exactly where every actual route still requires a positive amount. An argument that progressively cheaper routes are conceivable therefore needs a separate step to establish a route with zero cost. The same logical issue can arise with shorter delays, lower risks, or smaller administrative burdens whenever a limit is used to represent an attainable mechanism.

The counterexamples concern mathematical representation, leaving the appropriate resource domain to the application. Some quantities are counted in indivisible units, while others are modeled continuously. A continuous approximation to a discrete process may be convenient; a discrete approximation to a continuous process may be computationally tractable. Either move changes which exact statements about boundaries and minima are available. The domain choice belongs among the substantive assumptions of the feasibility model.

12.6 Historical and Practical Interpretation of the Basis

The finite basis is relative to a target and a question. It retains the minimum permission and initial-stock combinations that suffice for at least one admitted route. It omits the identity of that route unless a witness is stored alongside the profile. Multiple routes with the same profile can differ in sequence, duration, exposure to intervening events, institutional provenance, and the persons who perform them.

For this reason a useful research record can pair each minimal profile with one or more witness words and with the explanation of their admissibility. Such a record permits the availability calculation to remain compact while giving the reader a route back to the mechanism. A selected witness supplies an existence certificate for its profile; additional witnesses can document historical variation or different ways to realize the same threshold. Their interpretive value can survive even when they add no new Boolean availability scenario.

DRAFT

Dominated profiles can also merit preservation. A route demanding more initial resources might finish sooner, rely on different participants, or better satisfy a separately stated evaluative standard. Dominance in (22) compares only permissions and resources. Enlarging the profile with duration or another attribute changes the order and can change which routes are minimal. Treating the original minimal family as the set of preferred actions would therefore add an evaluative commitment beyond the theorem.

The relation to historically formed potential is similarly specific. A present resource stock can be sufficient for a target while the historical process that produced the stock remains relevant to its ownership, authorization, or future stability. The availability basis can use an approved present-state description and leave those questions to a joint historical record. Where the history changes the applicable permissions or the stock's admissible uses, it must enter the profile construction itself. The required integration follows the query and the mechanism assumptions.

The finite-basis result consequently supports a limited compression strategy with explicit advantages. Under its discrete assumptions, unbounded word families can have finite representations of existential availability. The proof identifies why that compression is valid, the examples identify where it fails, and the interpretation identifies which further questions require retained witnesses or additional coordinates. This gives the model a way to simplify one task while preserving the research obligations attached to the others.

13 Scenario Probability and Outcome Realization

This section adds a probability distribution over condition scenarios and separates the probability that a target is available from the probability that a selected execution reaches it. It derives the effect of shared prerequisites, calculates a finite example, and examines the additional assumptions behind selection and parameter sensitivity. All numerical values are stipulated illustrations. Empirical identification of a distribution remains a separate task.

13.1 Probability over Scenarios

The availability function is Boolean at each scenario. A distribution over scenarios turns it into a probability of availability. This subsection defines that enrichment and its information requirements.

Let μ be a probability distribution on the finite set 2^E. Then

$$\Pr_{\mu}(\text{target available}) = \sum_{P \subseteq E} \mu(P)\Phi(P). \tag{25}$$

The formula sums the probability mass of successful scenarios. It follows from the definition of the event $\{P : \Phi(P) = 1\}$.

The distribution is additional to the path and requirement model. A nonempty target fibre supplies a possible route, while μ specifies how condition scenarios are weighted. The route structure alone gives neither those weights nor a rule for estimating them.

A joint distribution can represent correlations among conditions. Shared provision, common control, or other dependencies can make simultaneous availability differ from a product of marginal probabilities. The model should state whether such dependence is known, assumed, or unresolved.

The probability concerns availability at the chosen seed, horizon, and target. It leaves route selection and actual execution open. A successful scenario can contain many feasible words, including words that fail to reach the target or merely preserve the initial state.

13.2 Shared Requirements and Route Dependence

Independent conditions can still produce dependent route events when the routes share a condition. This subsection derives that dependence in the two-route example.

DRAFT

Let the condition labels be p, a, b, with independent availability probabilities $\theta_p, \theta_a, \theta_b$. The two routes require $\{p, a\}$ and $\{p, b\}$. Their individual success probabilities are $\theta_p\theta_a$ and $\theta_p\theta_b$.

Both routes are available together exactly when all three conditions are present. Independence of conditions gives intersection probability $\theta_p\theta_a\theta_b$. Inclusion and exclusion therefore yields

$$A = \theta_p\theta_a + \theta_p\theta_b - \theta_p\theta_a\theta_b = \theta_p(\theta_a + \theta_b - \theta_a\theta_b). \tag{26}$$

The shared factor is counted once in the intersection. Treating route events as independent would instead multiply their individual probabilities, producing $\theta_p^2\theta_a\theta_b$. That corresponds to a different joint event model.

The mistaken independent-route formula is

$$A_{\text{ind}} = 1 - (1 - \theta_p\theta_a)(1 - \theta_p\theta_b).$$

Subtracting Equation (26) gives

$$A_{\text{ind}} - A = \theta_p(1 - \theta_p)\theta_a\theta_b. \tag{27}$$

The difference is nonnegative and is positive when the shared condition is uncertain and both route-specific conditions have positive probability.

The error vanishes at special boundaries, including $\theta_p = 0$ or $\theta_p = 1$. Those cases do not justify route independence in general. They show that a numerical coincidence can hide an incorrect structural assumption.

13.3 Exact Numerical Illustration

The following calculation makes the size of the dependence effect explicit. Its values are hypothetical and serve to check the formulas.

Take $\theta_p = \frac{1}{2}$ and $\theta_a = \theta_b = \frac{4}{5}$. Then

$$A = \frac{1}{2}\left(\frac{4}{5} + \frac{4}{5} - \frac{16}{25}\right) = \frac{12}{25} = 0.48.$$

Each route individually has probability $\frac{2}{5}$, so the independent-route formula gives

$$A_{\text{ind}} = 1 - \left(1 - \frac{2}{5}\right)^2 = \frac{16}{25} = 0.64.$$

The overstatement is $\frac{4}{25} = 0.16$, which also follows by substitution in Equation (27).

Now add a direct route requiring a separate condition c, independent of the old conditions, with $\theta_c = \frac{1}{2}$. The target fails only when the old routes fail and c is absent. Hence

$$A' = 1 - (1 - A)(1 - \theta_c) = 1 - \frac{13}{25}\frac{1}{2} = \frac{37}{50} = 0.74.$$

The increase concerns availability under the enlarged scenario distribution. It leaves the full-availability outcome target unchanged. The new route supplies support in scenarios where the shared condition of the old routes is absent.

The calculation also requires a joint distribution on the enlarged condition universe whose marginal on the old conditions remains the original distribution. Without that preservation, the before-and-after comparison would combine a route change with a distribution change.

DRAFT

13.4 Marginal Information and Unresolved Dependence

Route marginals alone need not determine availability. This subsection gives a simple counterexample and states the resulting evidence requirement.

Consider two route events, each with probability one half. If they coincide, their union has probability one half. If they are disjoint and cover the probability space, their union has probability one. The marginals agree in both constructions.

The missing information is the intersection probability. The union formula requires it, either directly or through justified assumptions about the underlying conditions. A claim of independence supplies one possible value and needs support when used empirically.

A shared condition map can help explain dependence, but it does not automatically specify all correlations. Two nominally distinct conditions can still share a provider or respond to a common environmental change. The joint model should represent dependencies relevant to the target calculation.

An inquiry can retain bounds or several candidate distributions when the joint law remains unresolved. Reporting one precise availability probability without its dependence assumptions would conceal the source of the precision.

13.5 Selection among Feasible Words

A successful scenario contains at least one target route, but an execution must select a word. This subsection introduces a conditional path distribution and derives the probability of a target endpoint.

For each scenario P, let π_P be a probability distribution on all prefix-feasible words of length at most h. The empty word ensures that this set is nonempty. The execution model first draws a scenario according to μ and then a word according to π_P.

Under deterministic execution of the selected feasible word,

$$\Pr(\mathrm{End}(v)) = \sum_{P \subseteq E} \mu(P) \sum_{w \in \mathcal{W}_{A_P,h}(x,v)} \pi_P(w). \tag{28}$$

Here A_P is the availability rule induced by scenario P. The inner sum is the conditional probability assigned to target-reaching words.

The distribution π_P can describe an idealized selection law. It does not by itself establish what the actor observes or how a decision is implemented. A law that depends on the complete scenario can use information unavailable to the actor. The next section derives that limitation explicitly.

The selected word also needs an execution interpretation. The present formula assumes its declared feasible steps occur according to the deterministic maps. Additional execution failure, adaptation, or stochastic responses require a richer probability model.

13.6 The Availability Bound and Its Equality Condition

Availability gives an upper bound on target realization under any supported selection law. This subsection derives the bound and states exactly when equality holds.

Proposition 13.1 (Availability bound)**.** *Under the finite scenario and selection model,*

$$\Pr(\mathrm{End}(v)) \leq \sum_{P \subseteq E} \mu(P)\Phi(P). \tag{29}$$

Equality holds exactly when every successful scenario with positive μ mass assigns all conditional selection probability to target-reaching words.

Proof. If $\Phi(P) = 0$, the target-word fibre is empty and its conditional mass is zero. If $\Phi(P) = 1$, that mass lies between zero and one because π_P is a probability distribution. Thus the inner sum in Equation (28) is at most $\Phi(P)$ for each scenario. Multiplying by nonnegative $\mu(P)$ and summing proves the bound.

DRAFT

The difference between the right and left sides is a finite sum of nonnegative terms. It vanishes precisely when each term with positive scenario mass vanishes. For a successful scenario, this means that the conditional target mass is one. □

The equality condition concerns the path law, not merely the existence of a route. It can be unattainable under an actor's information constraints even when a scenario-specific successful choice exists everywhere.

The empty word also affects the interpretation. If $v = q(x)$, selecting zero steps can contribute to the endpoint event. A claim about realized activity requires an event or positive-execution criterion beyond unchanged endpoint membership.

13.7 Catalog Enlargement and Selection Effects

More feasible routes can preserve availability while reducing success under a changed selection rule. This subsection gives a finite example that separates existence from behavior.

Take a horizon of one and a start state distinct from the target. Initially, one operation s reaches the target. The feasible words are ϵ and s. Uniform selection gives target probability one half.

Add an operation f reaching a different non-target state, while preserving the old operation. The feasible words become ϵ, s, f. Uniform selection now gives target probability one third. Availability remains one in both catalogs.

The decrease arises because the selection law changes with the catalog. The old successful word remains available, but receives less probability. A rule that preserved its old mass would give another comparison.

The example supplies a countercondition to a claim that enlarged possibility automatically raises realized success. The probability model must specify how choices respond to the added options. Reasons, learning, and information can all matter to that response.

13.8 Parameter Sensitivity and Its Interpretation

The availability polynomial permits exact sensitivity calculations. This subsection derives them and distinguishes them from causal effects of a real intervention.

For the shared-condition model, ordinary differentiation gives

$$\frac{\partial A}{\partial \theta_p} = \theta_a + \theta_b - \theta_a\theta_b,$$
$$\frac{\partial A}{\partial \theta_a} = \theta_p(1-\theta_b), \qquad \frac{\partial A}{\partial \theta_b} = \theta_p(1-\theta_a). \tag{30}$$

Each derivative is nonnegative on the unit cube, consistently with the positive availability structure.

The route-specific sensitivity decreases when the alternative route's condition is already highly available. The shared condition affects both routes. These are consequences of the stipulated family and independence assumptions.

A real intervention on one condition can also change other probabilities or the dependence structure. The derivative holds those parameters fixed. Its causal interpretation requires evidence that the proposed intervention matches that comparison.

The calculation therefore supports model analysis. It can identify which assumptions or parameters drive a predicted availability change. It leaves empirical estimation and intervention warrant open.

13.9 Probability and the Actualization Question

Probability adds weights to possibilities, while actualization concerns the selected and executed course of activity. This final subsection states the separation that the preceding derivations establish.

DRAFT

A path witness supplies existence. A scenario distribution supplies availability probability. A selection law supplies conditional choice among feasible paths. An execution model supplies the connection between the selected word and its outcome. Each layer adds assumptions.

The distinction permits precise claims at different levels. A study can establish that a route exists while leaving its frequency unresolved. It can estimate availability while leaving the actor's choice model open. It can observe an execution while remaining uncertain about alternatives that were available.

The formalism should preserve those statuses. A nonempty fibre is insufficient grounds for assigning probability, and a positive probability does not establish that an event occurred. An actual occurrence requires its own evidence.

The next section examines the information available to a choosing actor. It shows why even perfect pointwise availability can fall short of a uniformly successful policy.

14 Information and the Selection of Feasible Paths

This section examines the information needed to select a successful path. It distinguishes pointwise availability from a rule that an actor can implement from available observations, proves a finite counterexample, and derives an observation-cell criterion for open-loop choice. It then considers diagnostic activity, adaptive selection, and the interpretation of agency. The analysis adds an information structure that path existence alone leaves unspecified.

14.1 Scenario-Conditioned Choice and Available Observation

The distribution π_P in the preceding section can depend on the complete scenario. An actor's decision may have access to less information. This subsection identifies the resulting implementability question.

Let Θ be a set of possible scenarios and let $O : \Theta \to Y$ be the observation available before a choice. Scenarios with the same observed value are indistinguishable to a rule using only that observation. Such a rule must choose the same action or distribution at the same observed value.

The scenario can include available conditions, resource stocks, or another uncertainty relevant to execution. Its observation map can reveal all, some, or none of those features. The model should state which information the choosing actor actually receives.

A family of successful choices specified separately for every scenario can therefore use unavailable information. The family is mathematically well-defined and may describe an informed benchmark. Its existence does not establish that one actor can implement it.

This distinction is independent of the actor's computational ability. Even an arbitrarily capable decision procedure receives the same input in observationally identical scenarios. It cannot choose differently between them without another source of information or randomization.

14.2 A Two-Scenario Counterexample

A minimal example separates the existence of a successful route in each scenario from a common successful decision rule. This subsection specifies the outcomes and derives the deterministic limitation.

There are two scenarios, A and B, with the same initial observation. The horizon is one. In scenario A, action a reaches a target distinct from the start, while action b fails. In scenario B, action b reaches the target, while a fails.

Each scenario individually has a feasible successful action. Pointwise availability is therefore one. The successful choice differs between the scenarios.

A deterministic rule using only the common observation must choose the same action in both. Choosing a fails in B, while choosing b fails in A. Thus no such rule guarantees success.

The example can be interpreted through hidden prerequisites: only the condition for a is available in one scenario, and only the condition for b in the other. A wrong attempt then fails the feasibility test. The attempted-action model explicitly includes that failure outcome.

DRAFT

This differs from assuming a selection distribution already supported on feasible words in each scenario. The latter assumption can encode knowledge of feasibility. The counterexample shows why that support condition itself needs an implementability argument.

14.3 Randomization and Guaranteed Success

Randomization can improve the worst-case result of the example, while leaving a gap from pointwise availability. This subsection derives the exact bound.

Let p be the probability of choosing a at the common observation. The probability of choosing b is $1 - p$. Success is p in scenario A and $1 - p$ in scenario B.

Proposition 14.1 (Information-limited success bound)**.** *The greatest guaranteed success probability over the two scenarios is one half. It is attained by choosing each action with probability one half.*

Proof. For any $p \in [0, 1]$, the smaller of p and $1 - p$ is at most their average, which is one half. At $p = \frac{1}{2}$, both success probabilities equal one half. Hence the bound is attained. □

Randomization distributes the risk of choosing the wrong action. It supplies no observation that identifies the scenario. The difference between a guarantee and an average under a prior remains important.

Suppose $\Pr(A) = q$. Expected success is

$$qp + (1 - q)(1 - p) = (1 - q) + (2q - 1)p.$$

The expression is affine in p. It is maximized at $p = 1$ when $q \geq \frac{1}{2}$ and at $p = 0$ when $q \leq \frac{1}{2}$, giving maximum $\max(q, 1 - q)$.

A prior concentrated on one scenario can make expected success close to one while the worst-case guarantee remains limited. The two criteria answer different questions. A research report should state which one motivates its policy comparison.

14.4 The Observation-Cell Criterion

The counterexample generalizes to a finite open-loop word-selection problem. This subsection derives the exact condition for one observation-based rule to succeed in every scenario.

Let Θ be finite and let W_θ be the set of feasible successful words in scenario θ. An open-loop rule selects one complete word after observing $O(\theta)$ and before receiving further observations.

Proposition 14.2 (Common successful words within observation cells)**.** *A deterministic rule based only on O succeeds in every scenario if and only if*

$$\bigcap_{\theta : O(\theta) = y} W_\theta \neq \varnothing \quad \textit{for every attained } y. \tag{31}$$

Proof. A rule assigns one word to each attained observed value y. Universal success requires that word to belong to W_θ for every scenario with observation y. Hence it belongs to the intersection, proving necessity.

Conversely, suppose each intersection is nonempty. Choose one member for each of the finitely many attained observations and let the rule select it. By construction, the word succeeds in every scenario compatible with that observation. □

Pointwise availability states that each W_θ is nonempty. The proposition requires a common member within each observation cell. The difference is the order of the quantifiers: scenario-specific witnesses can vary where an implementable rule must remain the same.

The criterion concerns open-loop choice. A rule that observes intermediate outcomes and branches can use information acquired during execution. Such a policy may succeed without one common complete word. It requires a richer representation of observation histories.

DRAFT

14.5 Diagnostic Activity and Horizon Costs

Information can be acquired through activity. This subsection examines how a diagnostic step changes the counterexample and why its cost belongs in the path model.

Suppose a diagnostic operation is available in both scenarios and reveals whether the scenario is A or B. After that observation, a rule can choose the corresponding successful action. With a horizon of two, the combined policy can guarantee success under these stipulated rules.

The diagnostic operation uses one step. Under a one-step horizon, it can reveal the scenario without leaving time for the target action. Information improves the decision structure, while its acquisition consumes part of the execution budget.

The diagnostic operation can also require resources or change the conditions being diagnosed. A probe that depletes the needed resource can reveal the successful route and simultaneously make it unavailable. The information and resource models then need a common trajectory.

An informative observation should therefore be represented with its availability, timing, cost, and effects. Treating perfect information as free can create an unrealizable policy benchmark.

The example remains constructive: it shows one explicit way to close the information gap when the diagnostic assumptions hold. A substantive application must justify those assumptions.

14.6 Adaptive Policies and Shared Histories

A policy can choose later actions in response to earlier observations. This subsection identifies the additional structure needed to analyze such adaptation.

An adaptive rule maps an available observation history to the next action. Two underlying executions with the same observed history must receive the same decision under a deterministic rule. Different observed histories can support different continuations.

The relevant information state can be a set of scenarios compatible with the observations so far. A new observation narrows that set under a valid evidence model, while an action changes the possible successor states. The update must retain both operations in the correct order.

A path catalog can still describe the possible complete executions of a policy. The policy's branching rule adds constraints linking those paths across indistinguishable histories. Selecting each path separately can violate those constraints.

An adaptive analysis requires a policy semantics, observation updates, and a success criterion across possible executions. Section 15 supplies those components and a full induction proof for finite deterministic models. The open-loop theorem isolates the fixed-word case and its implementability condition.

This extension can matter when an actor learns which conditions are available through attempts, inquiries, or cooperation. The model should represent the information actually obtained, including ambiguity or misleading feedback.

14.7 Choice, Reasons, and the Target of Success

A feasible informed action can remain unselected for reasons outside the model's target criterion. This subsection separates implementability from motivation and evaluation.

The target v is selected by the inquiry. An actor may value another outcome, avoid a cost, or reject a route's conditions. A policy optimized for reaching v therefore describes a declared planning task, rather than a universal account of rational action.

A route can also require coordination among several actors. One operation label can conceal their distinct decisions and information. A single-policy model needs justification when no one actor controls the complete sequence.

Reasons and commitments can influence which available path is chosen. They can be represented through additional preference, rule, or interpretation structures when the research question requires them. Path existence alone supplies none of those structures.

DRAFT

The distinction protects the interpretation of nonuse. An available route can remain unused because it is unknown, because execution is too costly, because another outcome is preferred, or because the model omitted a constraint. Observed nonuse does not uniquely identify the explanation.

A comprehensive inquiry should therefore preserve several candidate accounts until evidence distinguishes them. The formal layers specify where each account enters the analysis.

14.8 Interpretive Consequences for Realization

The information analysis adds a necessary qualification to the passage from possibility to realization. This final subsection states that qualification and its research implications.

Pointwise availability means that each scenario contains some successful route. Implementability requires a rule whose choices respect the actor's information and control. Actual execution adds the occurrence of the selected operations.

The two-scenario example proves that the first achievement can hold while the second fails. The observation-cell criterion identifies the missing common witness for open-loop choice. Diagnostic activity supplies one possible repair under additional assumptions.

The result also qualifies the probability model. A scenario-conditioned path distribution can serve as an informed upper benchmark. Its attainability requires a policy that can realize those conditional choices from the available observations.

The broader framework therefore treats realization as a sequence of distinct requirements: support, prefix feasibility, resources, control, information, selection, and execution. Their coordination is the substantive task. The existence of one formal object leaves the remaining requirements open until they are specified and tested.

15 Adaptive Feasible Termination under Partial Observation

An actor can choose a later operation using information acquired during an earlier operation. The open-loop criterion therefore leaves an important class of realization problems unresolved. This section supplies a finite deterministic policy model, derives its information update and winning recursion, and proves the recursion by induction. A diagnostic illustration then separates adaptive success from success through a common fixed word. The final discussion examines the cost of observation, target certification, and the limits of the finite-state assumptions.

15.1 State, Information, and Available Actions

Let S be finite, let B be a finite operation alphabet, and let $T_b : S \to S$ be the deterministic update associated with b. The set $A(x) \subseteq B$ contains the actions available at x. The state contains the conditions needed to determine both the update and availability. If resource stocks or historical commitments affect those rules, their relevant information belongs in the state used by this section.

An observation map $o : S \to Y$ supplies a value after each executed action. Only the finite attained observation set matters, even if Y is formally larger. The actor knows its initial information and remembers its own actions and subsequent observations. A deterministic policy chooses an action or a stopping decision from that observed history. Under these assumptions, two possible executions with the same retained information require the same current decision.

Represent current uncertainty by a nonempty compatible set $K \subseteq S$. A feasible decision must work as an available action in every state still represented:

$$A(K) = \bigcap_{x \in K} A(x). \tag{32}$$

This requirement follows from the intended guarantee. If a policy selects an action unavailable at some compatible state, the model contains an admissible initial alternative from which the policy fails its feasibility obligation.

DRAFT

The requirement can be changed by changing the semantics of an attempt. If an unavailable operation can be attempted and produces an observable failure, the model can include an attempt action with that successor. The failed attempt then has a defined update and must carry any resulting cost or loss. Such an extension can be useful for diagnostic behavior. The present formulation keeps availability explicit, so the distinction between executing an available operation and attempting a potentially unavailable one remains traceable.

Let $G \subseteq S$ be the target. A policy is permitted to terminate successfully when its current compatible set is contained in G. This defines a guarantee of certified current target attainment. It requires the target to hold for every state consistent with the information on that terminating branch. A task concerned with a past target visit can be modeled by adding a variable that records whether the visit has occurred.

15.2 Propagation and Observational Filtering

The next compatible set combines an action's physical effect with the information obtained afterward. This subsection derives the update in the stated deterministic setting. LaValle's information-state construction provides the verified planning antecedent for propagation, observational filtering, and target inclusion [3].

For $b \in A(K)$, the set of possible successor states before observing the result is $T_b(K) = \{T_b x : x \in K\}$. If the observed value is y, the compatible successor set becomes

$$K_{b,y} = T_b(K) \cap o^{-1}(\{y\}). \tag{33}$$

Only values $y \in o(T_b(K))$ can occur in the model; their successor sets are nonempty.

The derivation has two parts. Each actual state $x \in K$ produces the successor $T_b x$, so the successor lies in the image of K. The observation then requires $o(T_b x) = y$, which restricts that image to the displayed preimage. Conversely, a member of the intersection arises from some current state in K and has the reported observation. It is therefore compatible with the declared transition and observation information.

The order of these operations matters. Filtering the old compatible set by a post-action observation would use evidence about the wrong time unless a preimage through the action is included. One may equivalently filter the old set by $T_b^{-1}(o^{-1}(\{y\}))$ and then propagate it. This equality follows from the elementary image/preimage identity developed for historical evidence in the companion paper. The update in (33) keeps the temporal interpretation direct.

An empty intersection can represent an impossible observation branch in a stipulated model. If such an observation is received in an application, the discrepancy requires examination of the candidate set, transition model, or observation rule. Treating the empty set as successful target certification would conceal that discrepancy. The policy recursion therefore quantifies only over nonempty branches generated by the model.

15.3 A Recursion for Guaranteed Termination

The finite information-state space supports an explicit horizon calculation. This subsection defines the winning families and proves that they characterize exactly the policies permitted by the stated information and feasibility rules.

Set

$$\mathcal{W}_0 = \{K : \varnothing \neq K \subseteq G\}.$$

These are the compatible sets from which the actor can stop successfully without another action. For $h \geq 0$, define

$$\begin{aligned} \mathcal{W}_{h+1} = \mathcal{W}_0 \cup \{K \subseteq S : K \neq \varnothing,\ \exists b \in A(K), \\ \forall y \in o(T_b(K)),\ K_{b,y} \in \mathcal{W}_h\}. \end{aligned} \tag{34}$$

DRAFT

The existential choice concerns one action available from the actor's current information. The universal condition concerns every observation that can follow it. Different observations can support different later choices.

Proposition 15.1 (Finite-horizon adaptive feasibility)**.** *A compatible set K belongs to $\mathcal{W}_h$ exactly when an observation-based deterministic policy guarantees feasible certified target termination within h actions from every state in K.*

Proof. At horizon zero, the only successful decision is to stop with every compatible state already in G. This is precisely the definition of $\mathcal{W}_0$.

Assume the characterization holds at horizon h. Consider a successful policy with at most $h+1$ actions. If it stops immediately, $K \in \mathcal{W}_0$. Otherwise it chooses one first action b. The policy receives the same current information for every state in K, so it chooses that same action throughout the compatible set. Guaranteed feasibility requires $b \in A(K)$.

Every observation $y \in o(T_b(K))$ can arise from at least one state in K. On that branch the remaining policy has at most h actions and must succeed from every compatible successor in $K_{b,y}$. The induction hypothesis gives $K_{b,y} \in \mathcal{W}_h$ for every such observation. This proves necessity of the recursion.

Conversely, suppose K satisfies (34). If $K \in \mathcal{W}_0$, stop. Otherwise choose an action b witnessing the existential condition. It is feasible from every member of K. For each possible observation y, the induction hypothesis supplies a successful policy of at most h actions from $K_{b,y}$. Use that policy after observing y.

There are finitely many possible observations, so these branches form one finite policy. Equation (33) places the actual successor in its selected branch. The branch policy then supplies feasible certified termination within the remaining budget. Including the first action gives the required bound of $h+1$, proving sufficiency. □

The proof displays the exact point at which a policy differs from a list of scenario-specific routes. The root action is shared across indistinguishable alternatives. Branches can diverge only after a difference enters the observed history. A proposed realization plan that assigns different root choices to the same compatible set would use information absent from the model.

15.4 Information Refinement and Horizon Monotonicity

The recursion has useful structural consequences. This subsection derives them while keeping information refinement separate from activities that acquire information at a cost.

First, $\mathcal{W}_h \subseteq \mathcal{W}_{h+1}$: a policy that uses at most h actions is still permitted under a budget of $h+1$. Second, winning sets are downward closed among nonempty compatible sets. If $K' \subseteq K$ and a policy guarantees success from K, it can be followed for the smaller set of possible initial states. The guarantee still applies to each remaining possibility.

The second result can also be checked directly in the recursion. At horizon zero, a subset of a target-contained set remains target-contained. At a later horizon, any action common to K is common to K'. Each successor set from K' is a subset of the corresponding successor from K. Induction gives a winning continuation on every nonempty branch. This argument makes the role of valid information explicit: narrowing the compatible set can improve the guarantee when the true state remains represented.

A related result concerns richer observations. Suppose o' refines o, meaning that $o = f \circ o'$ for some map f. Keep the timing, transitions, action availability, and costs unchanged. An actor receiving o' can reconstruct the old observation by applying f, then follow its old policy. Induction on the execution shows that the same choices and actual trajectory remain available. Every formerly winning policy therefore survives free observation refinement.

This conclusion has a precise scope. It concerns receiving more information at the same decision points under the same dynamics. A diagnostic action can consume a step, use resources, or change the arrangement it observes. Those changes alter other premises of the comparison. The usefulness of its information must then be assessed together with the altered trajectory.

DRAFT

The finite-state assumption also gives an existence bound for the iterative calculation. There are $2^{|S|} - 1$ nonempty compatible sets. If two consecutive winning families agree, the same recursion produces agreement at every later stage. Each strict increase adds at least one previously absent compatible set. Consequently the iteration reaches a fixed point after at most $2^{|S|} - 1$ strict additions of information states. This coarse bound concerns the finite represented problem; the exponential state space can still make explicit calculation demanding.

15.5 A Diagnostic Policy without a Common Successful Word

A six-state illustration makes the advantage of adaptation explicit. The example has uncertain initial state and deterministic operations, so the difference comes entirely from the timing of information and choice.

Let

$$S = \{u_A, u_B, k_A, k_B, t, f\}, \qquad G = \{t\}, \qquad K_0 = \{u_A, u_B\}.$$

The states u_A, u_B have a common observation. The states k_A and k_B have distinct observations identifying their respective cases. Target t and failure f have their own observations. All three actions d, a, b are initially stipulated available everywhere.

The diagnostic action d sends u_A to k_A and u_B to k_B; it fixes the other states. Action a sends u_A, k_A to t and u_B, k_B to f. Action b sends u_B, k_B to t and u_A, k_A to f. Both t and f are absorbing under every action.

Within one action, the target lacks a guaranteed policy. Action d reaches an informative intermediate state, while a and b each send one compatible initial state to failure. Thus $K_0 \notin \mathcal{W}_1$.

Within two actions, choose d first. The next compatible set is either $\{k_A\}$ or $\{k_B\}$, depending on the observation. Choose a in the first case and b in the second. Each branch reaches t, so $K_0 \in \mathcal{W}_2$.

There is nevertheless no common successful fixed word at any finite length. A word consisting only of d leaves the system outside the target. Every target-reaching word must eventually use a or b. Before the first such action, the two possible executions remain in their respective A and B cases. That first a or b sends one case to the absorbing failure state. Later operations cannot repair it. This proves the absence of a common successful word, including words longer than the successful adaptive policy.

The example therefore gives a strict separation between an open-loop word and a contingent policy. Both are assembled from the same available operations. The policy uses a branch condition that a single predetermined sequence lacks. Counting successful words separately for each state would reveal pointwise possibilities while leaving this implementation distinction unresolved.

15.6 An Informative Probe that Removes Access

The information supplied by a diagnostic activity can coexist with a loss of practical access. This subsection modifies the example to isolate that possibility.

Keep the transition and observation maps, but change availability at the two diagnosed states to

$$A(k_A) = A(k_B) = \{d\}.$$

The actions needed to reach the target are now unavailable after the probe. One interpretation of this stipulated change is that diagnosis consumes a condition required by either completing action. The availability map records the condition's loss; an explicit resource coordinate could supply a more detailed realization of the same pattern.

Executing d still identifies whether the successor is k_A or k_B. Each successor set is a singleton, so uncertainty about the current state has disappeared. Yet neither singleton has a feasible route to t: its sole available action d returns it to itself.

The actor could try a or b before diagnosis, but one possible initial state would still reach absorbing failure. Therefore K_0 has no policy guaranteeing target termination. Perfect knowledge reached through the probe leaves the target unavailable in this variant.

Draft

The example is a countercondition to the claim that a sufficiently informative diagnostic always repairs the realization gap. It leaves intact the free-refinement result proved above because the variant changes action availability along the execution. The comparison must include what the probe does, when it does it, and which subsequent operations remain feasible.

An empirical interpretation would need evidence for all of those components. An inquiry, inspection, disclosure, or trial can produce information and also change expectations, resources, permissions, or willingness to participate. The model offers places to represent such effects. Assigning them to a particular practice remains a substantive research task.

15.7 Policy Guarantees and Their Interpretation

The adaptive extension sharpens the relation between feasible possibility and realization. This subsection states what its guarantee contributes and which questions remain outside it.

A winning policy establishes that the actor can use its declared observations to choose feasible operations leading to certified target termination. The construction coordinates uncertainty, prefix availability, and sequential choice. It also supplies a witness: the action selected at each relevant information state and the continuation used after each possible observation.

The guarantee remains conditional on the candidate set containing the actual state, the transition maps representing the effects of the actions, and the observation rule representing the information received. A formally winning policy can fail in an application when one of those premises fails. The corresponding research record should retain these premises so that a failure can lead to a specific model revision.

The target remains selected by the inquiry. Guaranteed feasibility does not establish that the actor endorses the target or accepts the route's consequences. Similarly, a single policy presumes that the modeled decision process can select the listed actions. When several actors control different operations and receive different observations, an additional coordination or game structure is needed.

Finally, the finite-state result leaves room for richer models without silently transferring its bound to them. Unbounded resource stocks, changing operation vocabularies, uncertain transition laws, and continuous observations can require larger information spaces. The present proof provides a complete bounded case whose assumptions can be inspected before those extensions are attempted. It closes the finite adaptive obligation while preserving the broader program's open mathematical and empirical questions.

16 Philosophical Interpretation of Feasible Possibility

This section examines the philosophical meaning of the distinctions developed in the formal model. It separates represented possibility, effective access, retained capacity, informed choice, and actual activity; considers control over the conditions of continuation; and states the evaluative commitments that the equations leave open. The discussion develops the repository's questions through explicitly limited interpretations.

16.1 Represented Possibility and Its Conditions

A path belongs to a model whose carrier and operations have already been specified. This subsection examines what that membership establishes and what remains open about the target phenomenon.

A feasible word provides a witness that the declared rules permit a target under specified conditions. The witness is stronger than an unstructured assertion that an outcome might occur: its intermediate steps and requirements can be inspected. It remains conditional on the adequacy of those rules.

The carrier can include configurations that have never occurred. Their inclusion supplies a representational possibility. A claim that such a configuration is physically, institutionally, or socially possible requires an interpretation of the carrier and evidence supporting the relevant mechanisms.

DRAFT

Generation within a fixed carrier can still describe a substantive change. An organization can form, an activity can begin, or a new arrangement can become actual even when its represented configuration was included among the model's possibilities. The mathematical carrier records the scope of representation, rather than deciding whether the event is historically new.

A change in the model can have a different meaning. Adding a previously omitted operation can reflect improved knowledge, a new procedure, or a revised abstraction. Those alternatives should remain separate. A larger formal repertoire does not identify which kind of change occurred in the world.

The value of the path representation is therefore conditional articulation. It makes a possibility claim specific enough to examine, while preserving the premises on which its interpretation depends.

16.2 Effective Access and Retained Capacity

Current feasibility can depend on both an actor's abilities and the conditions under which those abilities can be exercised. This subsection examines why an empty current path fibre cannot by itself determine retained capacity.

A researcher can be represented as retaining a skill while lacking a resource or permission needed to exercise it. Under the current scenario, the target route can be unavailable. The same empty fibre could also arise in a model where the relevant skill is absent. Current nonavailability leaves those explanations unresolved.

A capacity inquiry can compare the actor across a declared family of environments. Such a comparison needs an account of what remains stable about the actor and which changes belong to the environment. Removing every constraint by stipulation can create a counterfactual too remote from the intended question.

History can also alter the actor. If a period of deprivation changes the skill or the ability to use it, restoring an external condition need not restore the previous capacity. The relevant comparison then requires a dynamic account of the actor, rather than a timeless subtraction of constraints.

The path profile can contribute to that inquiry by displaying feasible activity under several specified scenarios. It supplies one component of a capacity analysis. A universal scalar would require additional arguments about domains, environments, measurement, and valuation.

Sen's discussion of basic capabilities provides a relevant comparison by distinguishing resources and utility from what people are able to do [4, Sections 3–4]. The present model retains its own path and condition definitions. Formal reachability alone supplies none of the evaluative commitments of that discussion.

16.3 Route Multiplicity and Meaningful Alternatives

Several represented paths can lead to one outcome without providing several substantively independent alternatives. This subsection examines the difference between multiplicity and meaningful choice.

Duplicated labels and idle steps can multiply word witnesses. Shared prerequisites can make distinct routes fail together. Common control can permit one decision to remove several routes. The earlier constructions show that these features can vary while the target remains unchanged.

A meaningful-alternative claim therefore needs a criterion beyond raw word count. The inquiry might care about different providers, different resource profiles, different exposure to withdrawal, or different participant experiences. Each criterion requires a corresponding record.

The same route can also serve several purposes. A path selected for one target can generate information, relationships, or intermediate capacities relevant to another. A model that records only the final target will omit those consequences unless they are added to the observation.

An outcome set can nevertheless be useful for a bounded question. Its limitation concerns the information it discards, not an inherent defect in set representation. The framework encourages the researcher to select the level appropriate to the comparison.

DRAFT

The philosophical issue is thus the individuation of alternatives. A path, a mechanism effect, a requirement profile, and an actor-accessible option can have different identity criteria. Their counts should retain those criteria.

16.4 Control over Continuation and Rule Revision

The ability to use an available route differs from the ability to sustain or revise its enabling conditions. This subsection develops that distinction through the control extension and its limits.

A participant can have access through a permission controlled by another actor. The current target may be available, while its continuation depends on a revocable condition. The actor cut identifies one aspect of that dependence under a declared removal model.

The ability to propose a new use or amend an access rule is a different capability. It can change the operation catalog, the requirement family, or the interpretation of the target. Removing conditions from a fixed scenario does not represent that kind of intervention.

Two arrangements can therefore offer the same current activity while differing in the distribution of standing to alter the conditions of future activity. A substantive comparison would need to identify those procedures and their actual effects.

This distinction connects to the source discussion of cultural continuity. Preserving a visible practice can differ from preserving participants' ability to reinterpret or revise it. The connection remains a proposed research direction; the present path model supplies only a bounded representation of access and control.

Rule-revision authority also can be contested within a group. An institutional label should not hide disagreement among participants. A model can represent several actors and procedures when the research question requires that resolution.

16.5 Information, Agency, and Nonuse

An available route can remain unchosen for several reasons. This subsection examines the interpretation of nonuse and the additional structure needed for an account of agency.

The information counterexample shows that an actor can face pointwise available routes while lacking a uniformly successful choice. The limitation comes from indistinguishable scenarios, even before resource or preference questions are introduced.

A route can also be known and feasible while remaining unwanted. The actor may prefer another outcome, reject a burden, or decline the relation through which access is offered. A model that equates nonuse with incapacity would omit these possibilities.

The research target can itself change through participation. An opportunity can reveal an existing interest or contribute to the formation of a new one. Later uptake alone does not distinguish those explanations. A history of participation and interpretation would be needed to examine them.

This possibility qualifies a fixed selection law. If interests, knowledge, or commitments change during activity, the policy model needs a corresponding state or history. A static π_P can remain a useful snapshot while leaving that development open.

An account of agency therefore requires more than an available-path set. It can need information, control, reasons, and the capacity to revise a course of action. The formal layers identify where those questions enter, rather than resolving them through the existence of a witness.

16.6 Generative and Reproductive Effects

The same mechanism can produce a new arrangement and sustain an existing one at different levels. This subsection separates those roles from their evaluation.

A generative description can concern the formation of a configuration, a newly accessible route, or a changed condition of activity. A reproductive description can concern the maintenance or recurrence of a pattern. The chosen carrier, target, and time scale determine which role is visible.

DRAFT

The fixed-semigroup reachability theorem does not rule out such changes. It states that every later unrestricted composite was already included in the original action closure. A new actual configuration or a shorter remaining route can occur within that represented closure.

The theorem does constrain one possible claim: action within the fixed system cannot produce a point outside its original unrestricted reachable set. A stronger expansion claim needs changed operations, a different availability comparison, a new carrier, or another explicit input change.

Evaluation remains separate. Maintaining a relation can sustain valued cooperation or preserve an unwanted dependency. Adding a route can increase agency or enable harmful activity. The descriptive role alone supplies no fixed moral sign.

The framework can assist evaluation by retaining which participants, conditions, and alternatives are affected. It leaves the standards used to assess those effects open to argument.

16.7 Formal Measures and Evaluative Plurality

Word counts, cut cardinalities, resource thresholds, and probabilities summarize different aspects of a path system. This subsection explains why they cannot be combined into a universal value measure without further commitments.

A larger outcome image records more distinguished targets. A larger cut cardinality records resistance to a specified removal class. A lower resource threshold records feasibility from a smaller stock. A higher probability records a change under a declared distribution and selection model.

These changes can conflict. A robust route can require more resources. A fast route can depend on concentrated control. An additional option can reduce success under a changed selection law. A single scalar would need weights or an ordering that resolves those tradeoffs.

The weights would express substantive judgments. They could reflect participants' purposes, institutional standards, or a chosen analytical objective. Their source and scope should be explicit. The formal model does not derive them from set inclusion.

Some comparisons can remain partial. Two profiles can be incomparable under the retained criteria, just as resource vectors can be incomparable. Preserving that result can be more informative than forcing a ranking.

The proposed framework therefore supports plural assessment through explicit dimensions. It aims to make the consequences of a chosen criterion clear while retaining the questions that criterion leaves unanswered.

16.8 The Scope of a Generative Relational Path Account

The path account coordinates several conditions of realization without treating them as an exhaustive theory of social action. This final subsection states the resulting research position.

A target can have mechanism support, a prefix-feasible route, adequate resources, surviving enabling conditions, an implementable selection rule, and an actual execution. Each achievement requires a different kind of record or assumption.

The conditional theorems clarify the transitions between those levels. They show when a product preserves feasibility, when a requirement family preserves availability, when a cut disables a target, and when information permits a common successful choice.

The philosophical interpretation requires additional work on the meaning of the target, the identity of actors, the adequacy of conditions, and the evaluation of outcomes. These obligations remain visible within ontological suspension.

The model's success should be assessed through a bounded inquiry in which its distinctions change what can be explained, tested, or responsibly claimed. The next section specifies the evidence and revision conditions for that assessment.

DRAFT

17 Evidence, Objections, and Research Obligations

This section identifies the evidence needed to apply the path framework and examines objections to its representational choices. It organizes the inquiry by formal layer, distinguishes model failure from changed scope, and proposes a bounded application. The purpose is to make the expanded account reviewable through explicit inputs and revision conditions.

17.1 Evidence for Operations and Intermediate Conditions

A feasible-path model requires more than a list of desired outcomes. This subsection states what must support its operations and prefix requirements.

An operation needs a criterion for identifying its occurrence and a response rule appropriate to the represented state. A label can summarize an extended procedure, but its boundaries should be stated. Otherwise step counts and composition can change with an unnoticed shift in granularity.

Availability must be assessed at the intermediate state. A condition present before preparation can expire, be consumed, or become irrelevant afterward. The persistent model is appropriate only when its fixed-scenario assumption fits the selected period and activity.

A route catalog also needs a scope argument. It can represent documented procedures, theoretically proposed alternatives, or experimentally tested paths. Those statuses should remain separate. An untested path can be retained as a conjectural witness without being reported as an established practical option.

The research record should connect each map and requirement to its supporting argument or source. A failed prediction can then be traced to a particular operation, availability rule, or omitted condition. The formal path alone supplies no evidential promotion.

17.2 Evidence for Shared Conditions and Control

The cut analysis depends on the individuation of conditions and their relation to control actions. This subsection identifies the corresponding evidence requirements.

Two route labels can conceal a shared prerequisite. A study should inspect whether apparently separate permissions depend on one credential, facility, or provision arrangement. The relevant dependence belongs in the condition or scenario model.

The control map needs evidence about what an actor can actually alter. A formal role, a technical capacity, and an effective withdrawal procedure can differ. The interpretation of D_a should specify which one is represented.

Joint consent or coordinated replacement can invalidate the unilateral union rule. A condition removable only through a coalition needs a coalition-dependent control model. A condition restored by another actor needs a response trajectory and horizon.

The resulting evidence can change the actor-level cut while leaving the condition-level family intact. That is a refinement of the intervention model, rather than a contradiction of the original set-theoretic derivation.

17.3 Probability, Information, and Selection Evidence

The probability and policy layers introduce assumptions that route existence cannot supply. This subsection states their separate evidential targets.

A scenario distribution requires a justified account of joint availability. Marginal estimates leave shared dependence open. Independence should be treated as a model assumption whose consequences can be tested, rather than a default property of different labels.

The observation map requires an account of what an actor can distinguish before choosing. A record available to the investigator may be unavailable to the participant. Using the investigator's information in a policy can overstate implementable success.

DRAFT

A selection law needs an interpretation consistent with that information and the actor's control. A distribution over feasible words can describe an informed benchmark while failing to represent actual decision making. The distinction should appear in the reported result.

Actual execution requires occurrence evidence. A planned word, an available route, and a completed sequence are different records. The study should preserve their relationship without promoting a plan into an event.

17.4 Objection from Excessive Formalization

An objection is that the framework adds notation to distinctions that can be stated in ordinary language. This subsection identifies the work that would justify the formal treatment.

The equations are useful when they expose a hidden assumption or support a discriminating consequence. The prefix trap separates existential composite availability from feasibility of a chosen word. The resource bound shows why final balance can fail. The information counterexample separates pointwise existence from implementable choice.

These results can change a model's conclusion or design. A study that would reach the same judgment without inspecting any of the distinctions should explain why the formal apparatus remains necessary for its purpose.

The framework also should permit a simpler model when its assumptions fit the task. Persistent sets can be adequate for a bounded permission inquiry. A full resource and policy model need not be added merely because it is available. The required depth follows the explanatory question.

The unresolved obligation is empirical and methodological: demonstrate a case in which the coordinated analysis clarifies what a coarser account leaves ambiguous. The finite illustrations establish possibility, while a substantive application must show relevance.

17.5 Objection from Arbitrary Primitives and Targets

The modeler's choice of labels, targets, and condition units can affect the results. This subsection treats that sensitivity as an explicit object of criticism.

A duplicated label changes word count without changing effects. A composite declared primitive changes distance without necessarily changing physical effort. Splitting a condition can change a removal count. These examples show that the representation can manufacture apparent improvement under an inadequately specified metric.

The response is to retain the individuation and measurement rules with every comparison. A count should name what it counts. A before-and-after analysis should state which rules remain fixed and which changed.

The target also requires justification. A model can optimize a completion flag while omitting cost, meaning, or distribution. An enlarged observation can represent additional concerns, but the choice of those concerns remains a substantive decision.

Sensitivity analysis can compare several defensible representations. Agreement across them can strengthen a scoped conclusion; disagreement identifies the premise on which it depends. The framework supports that comparison without promising a representation-independent scalar.

17.6 Objection from Unmodeled Adaptation

The fixed catalog can omit the responses through which actors restore or revise possibility after a condition changes. This subsection states the resulting limit and a route to extension.

A removal can trigger repair, substitution, negotiation, or rule change. If those operations are absent, the cut theorem describes loss within the current catalog and horizon. It leaves adaptive recovery open.

An extended model can include the relevant responses and recompute availability. The old cut can cease to disable the enlarged system. That change reflects a new route or rule, rather than an error in the earlier fixed comparison.

DRAFT

The extension needs evidence about which adaptations are available, who can initiate them, and what they require. Treating every imagined repair as feasible would make the model difficult to challenge. Treating every unobserved repair as impossible would narrow it without warrant.

The appropriate research status can remain a set of candidate extensions. Each can state a proposed response and its evidence requirement. A later application can test which candidates belong in the operational model.

17.7 A Bounded Empirical Application

A first application can examine access to one research or cultural activity through several routes over a specified period. This subsection outlines the study's responsibilities without selecting an empirical case in advance.

The study would define the target, the actors, and the horizon. It would record the intermediate steps of available or proposed routes, distinguishing actual attempts from hypothetical paths. Conditions would be coded with their persistence and quantity status.

The requirement family would then be derived from the ordered paths. Shared dependencies would be investigated, and condition-level cuts would be compared with actual control arrangements. Resource ledgers would be added where repeated use or timing matters.

The study would examine participants' information and reasons for using or declining routes. It would distinguish a route unknown to an actor from a known route the actor rejects. A fixed target measure would remain separate from participants' evaluations.

The proposed account would be revised if a credible route lies outside the catalog, if a prerequisite changes during execution, if a controller cannot perform the modeled removal, or if the policy uses unavailable information. Each trigger points to a specific layer of the model.

17.8 Research Status and Further Development

The formal results establish conditional relations among representations. This final subsection states the program of work that remains.

The path model gives a basis for analyzing support, prefix feasibility, shared conditions, resource timing, control, and informed selection. Its finite examples and checks corroborate the declared constructions.

Empirical validation remains open. A substantive inquiry must justify the carrier, operations, requirements, observations, and comparison scope. The philosophical interpretation must explain why the selected distinctions matter to the question.

Further formal work can investigate adaptive policies, changing condition universes, stochastic production, and rule revision. Each extension should preserve the assumptions and successful results of the earlier model while stating what has changed.

The framework's development should therefore proceed through specific questions and counterexamples. A comprehensive account gains strength from locating its limits precisely and from showing how a revised construction addresses a demonstrated need.

18 Conclusion

This paper has developed a path-based account of the conditions under which a represented outcome can be realized in a generative relational model. It distinguishes mechanism support from prefix feasibility, persistent requirements from consumable resources, condition-level robustness from actor control, and pointwise availability from implementable selection. The distinctions permit exact conditional results while preserving their interpretive limits.

Product compression is legitimate for a path property when that property is constant on evaluation fibres. The prefix counterexample shows that endpoint equality can coexist with different execution

DRAFT

status. An existential product catalog remains meaningful when it records that some feasible realization exists. Actual execution still requires a suitable witness or selection procedure.

Under persistent positive requirements, ordered target paths yield a family of sufficient condition sets. Its minimal members preserve Boolean availability, and target loss under removal is exactly a hitting-set condition. The finite illustration shows how an additional route can improve removal robustness while leaving full-availability outcomes unchanged. Duplicating a label leaves the requirement structure intact.

The extensions identify further conditions of realization. A control map changes the unit of removal analysis. A resource ledger derives the initial stock needed to satisfy every prefix, retaining the timing that a net balance loses. An observation model shows why successful scenario-specific choices can remain unavailable to an actor who cannot distinguish the scenarios.

The probability results preserve the same separation. A joint scenario law determines availability probability. A supported selection law determines target endpoint probability, bounded above by availability. Shared requirements create route dependence, and the bound's equality condition can fail under information constraints. Actual occurrence requires evidence beyond any of these possibility calculations.

The philosophical interpretation remains scoped. A current empty path fibre can leave retained capacity unresolved. More routes can preserve shared control. Progress toward a target can coexist with fixed or contracting unrestricted reachability. Evaluation of those changes requires standards concerning the target, affected participants, and distribution of burdens.

The restoration extension derives the persistent and resource deficits that an additive intervention must supply under fixed route profiles. Incomparable deficits remain a family of alternatives. With uncertain scenarios, one route works throughout the retained possibilities exactly when its profile is feasible at their common conditions and coordinatewise minimum stock. A route in every scenario can still require information that the actor lacks. The distinction locates a further requirement between availability and realization.

The framework provides a research method for making these commitments inspectable. A bounded application can now test its operation rules, condition profiles, control maps, resource timing, and information assumptions against an actual case. Its further development should follow the specific failures and unresolved comparisons that such an inquiry reveals.

Two further results specify the limits of route and policy compression. Finite permission vocabularies and fixed-dimensional integer resources admit finite minimal availability bases even for unbounded route families. The proof supplies an existence result while discovery, deadlines, and historical witness retention remain explicit responsibilities. Continuous-resource examples expose the role of discreteness and attainment.

For finite deterministic models under partial observation, the information-state recursion characterizes feasible certified termination. Its proof coordinates a common available first action with separate continuations after possible observations. The diagnostic examples establish both the benefit of adaptive choice and the possibility that an informative probe removes the access needed to complete the target. These results sharpen the passage from a possible path to an implementable policy.

References

[1] Todd J. Green, Grigoris Karvounarakis, and Val Tannen. Provenance semirings. In *Proceedings of PODS 2007*. ACM, 2007. Author-hosted full text.

[2] Wanhong Huang. A metamodel for ontologically heterogeneous social dynamics, part I: The highest abstract layer. Author-hosted working paper, 2026. Accessed 12 September 2026.

[3] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006. Author-hosted book; Section 2.1.1.

DRAFT

[4] Amartya Sen. Equality of what? In *The Tanner Lectures on Human Values*, volume 1. Cambridge University Press, 1980. Lecture delivered 22 May 1979. Organizer-hosted lecture text; publication year verified against the author's bibliography.

[5] Matyas Sustik. Proof of Dickson's lemma in ACL2 via an explicit ordinal mapping. ACL2 Workshop, Boulder, CO, presentation, 2003.

[6] W. E. Vesely, F. F. Goldberg, N. H. Roberts, and D. F. Haasl. Fault tree handbook. Technical Report NUREG-0492, U.S. Nuclear Regulatory Commission, January 1981. Official publication record.

DRAFT