

A Generative Grammar of Value Circuits in AI Production Networks

A Thought Experiment on Future Production Relations

Wanhong HUANG
huangwanhong@serendip.ngo

Abstract

This preliminary discussion paper explores a generative language for describing recurrent resource and service relations in hypothetical production networks operated predominantly by highly capable artificial-intelligence systems. It distinguishes simple graph cycles, repeated closed walks, and physically executable histories with typed inputs, time-indexed events, shared capacities, obligations, and differentiated access permissions. Unbounded recursion in a fixed finite graph yields a potentially infinite but regular language of closed walks; greater expressive claims require precisely specified representations and additional assumptions. A proposed grammar of production circuits would generate candidate event structures through serial, parallel, alternative, nested, and recurrent composition, subject to independently evaluated material, temporal, institutional, and regenerative conditions. The study addresses the ambiguity introduced by overlapping circuits and shared events, examines state-dependent participation, conditional cascade closure and serial fragility, and identifies possible interfaces for AI-assisted recognition from extensive, fallible production records. It develops a normative dossier for authorization, claims, review and affected interests, and demonstrates why material accounts alone underdetermine rule-specific findings. It also examines why a syntactically generated circuit need not sustain productive capacities, justify resource claims, or imply the disappearance of monetary coordination. The paper concludes by identifying bounded formal results, inspectable AI-assisted research uses, and the evidential and institutional conditions required for any economic deployment. Its proposed economic and normative semantics remain hypothetical.

Keywords: generative grammar; productive value circuits; AI production networks; recursive closed walks; temporal resource constraints

Discussion Paper Note

This manuscript studies a proposed formal language of value circuits within a bounded thought experiment on future AI-operated production. AI systems perform nearly all productive tasks, while resources, time, infrastructures, and institutional permissions may remain constrained. A graphically closed walk is treated as a candidate formal trace, with material, temporal, authorization, and regenerative conditions evaluated separately.

The framework distinguishes generation of circuits from their recognition in historical records, and recognizes that multiple grammatical derivations may refer to the same production event. It does not infer infinite economic value from infinitely many formal walks, or define a correct account of justice by grammatical construction. More capable AI and extensive shared data are conditional enabling premises, not evidence of omniscient planning or legitimate rule-making authority.

This manuscript forms Paper III in a four-paper programme. Paper I addresses money in AI-operated production, Paper II analyzes the conditions for distributed monetary dissolution, and Paper IV's earlier preliminary work on normative renormalization remains on hold until the present paper is completed. Each manuscript is intended to be readable independently.

DRAFT

Responsible Use and Institutional Safeguards

The provisional grammar describes hypothetical productive arrangements and does not authorize unconsented resource transfers, automated sanctions, extinguishment of claims, or unilateral changes to governance arrangements. Real applications would require context-specific authorization, independently reviewable evidence, and meaningful procedures for contestation. The text's Creative Commons licence governs reuse of the author's writing; it confers no operational authority over resources, subjects or institutions. Rights in third-party materials remain with their respective holders.

DRAFT

Notices

Status. Preliminary discussion paper. Sections 1–10 constitute a complete preliminary first draft. The source has undergone internal conceptual, proof-assumption, citation-scope and editorial checks, but has not received external peer review or final typeset inspection. The mathematical representations and stipulated examples are analytical constructions rather than empirically calibrated or externally validated models; proposed institutional applications require independent evaluation.

Licence. Except where otherwise indicated, copyright 2026 Wanhong Huang. This draft is distributed under the Creative Commons Attribution-NonCommercial 4.0 International Licence (CC BY-NC 4.0): <https://creativecommons.org/licenses/by-nc/4.0/>.

Statement on the use of language models. OpenAI’s ChatGPT assisted with dialogue, bibliographic verification, manuscript drafting, structural editing and LaTeX preparation. The human author supplied the central research programme and remains responsible for the text, citations, and errors.

Suggested citation. Huang, Wanhong (2026). *A Generative Grammar of Value Circuits in AI Production Networks: A Thought Experiment on Future Production Relations*. Preliminary discussion paper.

1 Introduction

This paper investigates the formal generation and recognition of productive value circuits in a hypothetical economy in which almost all operational labour is performed by highly capable artificial-intelligence (AI) systems. Its objective is to describe the structures through which resource transfers, production events, commitments, access permissions, and regenerative activities can be combined into candidate circuits, while specifying the additional tests needed before such structures can be treated as executable or economically significant. The inquiry uses a philosophical interpretation of economic relations together with a provisional formal-language and graph-transformation approach. It begins by distinguishing graph cycles from productive histories, considers recursive and overlapping circuit families, defines the limits of grammatical expressiveness claims, and identifies opportunities for AI-assisted recognition. The paper’s proposed grammar is an analytical instrument. Its formal results are conditional on their stated representations and assumptions; it does not independently determine economic value, distributive entitlement, or the legitimacy of a production regime.

The counterfactual environment is the same deliberately restrictive setting considered in the related inquiries on monetary coordination and distributed monetary dissolution. AI systems can undertake nearly every kind of productive task, including cultivation, extraction, energy provision, manufacturing, transport, scientific design, maintenance, and operational planning. This competence does not eliminate finite material stocks, geographically constrained infrastructure, limited throughput, perishability, and sequences that consume time. Productive inputs may be controlled by different operators or by principals whose authorization the operators execute. Operational agency and independent legal entitlement therefore remain distinct. The moral standing and legal personality of AI systems are left open. These assumptions permit an examination of complex production relations while avoiding a prediction that economic coordination must converge on either money or a particular nonmonetary institution.

A motivating example is a production sequence in which agricultural operator A provides grain to processor B , processor B makes food available through logistics operator C , and some output, material recovery, or complementary service eventually supports agriculture again. The static projection $A \rightarrow B \rightarrow C \rightarrow A$ forms a directed cycle. An additional equipment-maintenance exchange may generate $A \rightarrow R \rightarrow A$, while an electricity-service excursion may yield $C \rightarrow D \rightarrow C$. A longer closed walk can traverse $A \rightarrow B \rightarrow C \rightarrow D \rightarrow C \rightarrow A$ by entering the latter excursion before returning to agriculture. Here R denotes equipment repair and D electricity supply, consistent with the subsequent typed examples; the introductory graph remains an abstract operator projection with no stipulated physical quantities. These representations expose recurrence and overlap, although the economic interpretation of any arrow depends on its typed event record. A grain delivery, a battery-replacement service, an enforceable claim, and an authorization to operate shared equipment differ in their units, temporal conditions, and legal significance. Their inclusion in a single directed diagram is convenient for identifying possible pathways, yet their apparent geometric closure provides no assurance that the constituent activities are jointly practicable.

Figure 1 illustrates a limited operator graph containing three candidate recurrent routes. The figure distinguishes a basic production circuit, a short maintenance circuit, and a recurrent route that enters a nested electricity-service subcycle. The purpose of displaying the routes together is to make clear that individual edges and productive inputs may be reused by different descriptions of the network. The graph is therefore a catalogue of structural possibilities within a stipulated boundary, not an allocation plan, a measured economy, or evidence that the resulting loops generate an additional quantity of value.

The formal distinction between a *simple cycle* and a *closed walk* is essential. A simple directed cycle returns to its initial vertex while visiting no other vertex more than once. A closed walk permits repetitions of intermediate vertices and edges. A finite graph has only finitely many simple cycles, although their number may be very large; the enumeration problem is the subject of established algorithms, including Johnson’s output-sensitive procedure [9]. If a nonempty directed closed walk returns to the designated start vertex, repetition yields arbitrarily long return walks at that vertex. Reachability of a cycle without a directed return path does not suffice. This unbounded family does not imply unlimited

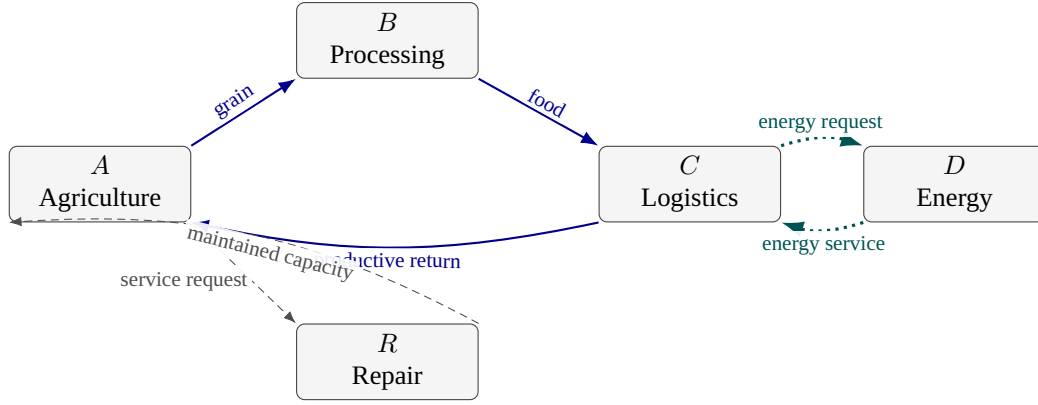


Figure 1: Overlapping candidate circuits in a five-operator AI production projection. The solid, dashed, and dotted relations represent different illustrative transfer or service types. One circuit may traverse a subcycle repeatedly; the figure alone does not establish joint material feasibility, permitted resource use, or regenerative value return.

production, an infinite stock of value, or a need to enumerate each traversal separately. It identifies a representational problem: how can finite rules describe an unbounded class of possible recurring histories without confusing their descriptions with distinct accomplished production events?

The simplest answer lies in automata theory. Once a finite directed graph, a finite alphabet for its labelled edges, and a starting vertex have been fixed, a finite-state machine can track the current vertex and accept precisely the edge sequences that constitute closed walks returning to the starting vertex. The resulting word language is regular, even when the set of accepted words is infinite [8]. Accordingly, recursive repetition alone supplies no basis for placing the language above context-free or other formal-language classes. The relevant complexity depends on which object is being generated or recognized, the encoding chosen for it, and the predicates incorporated into recognition. The paper treats comparisons with the classical hierarchy of formal languages as questions requiring specified representations and, where nontrivial claims are made, proofs. The foundational distinctions between finite-state and richer grammatical descriptions have a long history, including Chomsky’s early study of models for language description [3].

A vocabulary restricted to walks in a fixed graph would nevertheless obscure major characteristics of production. Grain processing may require grain, electricity, available equipment, and a legally permitted operating interval *together*. A supply of grain arriving after the processing window is materially different from grain delivered in time; two uses of one production machine may be individually possible while exceeding its shared capacity when scheduled concurrently. Furthermore, an operator’s right to direct a machine need not follow from its technical ability to operate the machine. A useful grammar must consequently specify multiple-input production events, typed quantities and units, ordered or concurrent execution, state-dependent conditions, and relevant authorization predicates. The theory of Petri nets provides established representations of concurrent events and finite resource conditions [15]; temporal network analysis clarifies why ordering and link timing matter for viable paths [7]. Their relevance to the present inquiry is methodological. Neither literature supplies a complete economic semantics of productive value or a criterion of legitimate distribution.

The circuit-generating object proposed here therefore has at least three distinguishable components. Its *syntax* specifies admissible compositions of typed events and candidate graph structures. Its *operational semantics* evaluates event histories against stocks, capacity, timing, conservation requirements, and state transitions. Its *institutional and evaluative interfaces* query who can authorize or contest transfers, which commitments remain undischarged, whether productive capacities are restored, and which normative interpretations are disputed. The third component cannot be reduced to a single numerical condition of physical feasibility. A circuit may be executable under a particular allocation rule yet remain open to legitimate objection, just as a shared production schedule may be acceptable despite an unre-

Table 1: Levels of description and validation in a proposed grammar of AI production circuits.

Analytical level	Object represented	Independent validation question
Graph / grammatical	Operator links, event types, productions, candidate circuits	Does the structure follow the formal rules?
Temporal / material	Stocks, joint prerequisites, capacities, delays, event schedules	Can the proposed history execute with available resources?
Institutional	Permissions, commitments, contestation and discharge rules	Are transfers authorized and obligations correctly accounted for?
Regenerative / normative	Renewal of productive conditions, recognized contribution and contested value return	Is continuation sustainable and appropriately justified?

solved disagreement over how benefits should be attributed. Multilayer-network representations support the formal separation of heterogeneous relations [10]; defining the economic and normative significance of their layers is an additional task of this research programme.

Table 1 sets out the corresponding distinctions. It locates each analytical question and the kind of evidence or justification required. The progression from graph structure to material execution and normative appraisal is not a claim that one layer automatically yields the next. The levels specify different forms of validation, and failures at a later level should not be hidden by the success of an earlier formal derivation.

The term *generative grammar* is used in the formal sense of rules capable of generating a class of structured objects and their candidate compositions. It does not independently attribute moral or economic generativity to every generated object. A graph rewrite may add a supplier, remove a route, split a productive facility, or change which agents may interact; the algebraic graph-transformation literature provides mature foundations for reasoning about such changes [5]. However, a newly generated route is only a possible relation until its physical, temporal, and institutional interpretation has been specified. The economic use of *generative value* likewise has a distinct scholarly background, notably Eglash and collaborators’ account of circular value flows and the conditions under which value-producing communities sustain themselves [4]. The present grammar draws motivation from questions of value return while making no claim to derive that normative tradition from grammar theory.

Overlapping circuits introduce a second difficulty beyond enumerating possible paths. One physical event may be included in many candidate circuits. An energy system can support agricultural production and equipment maintenance simultaneously, and a maintenance activity may support several future food-delivery histories. Multiple grammatical derivations can thus refer to the same physical resource transformation or shared capacity. Counting them as separate contributions would produce duplication. The grammar should retain event identity and provenance even when it generates several possible circuit decompositions. An admissible history should be evaluated once in the underlying material account, while distinct grammatical explanations can remain available for study. The attribution of benefits and duties among contributors requires further rules whose uniqueness is not assumed.

A third difficulty concerns time-varying relations. Within a fixed graph, a candidate circuit may appear to exist indefinitely. In an evolving AI production economy, resources expire, equipment deteriorates, permissions are granted and withdrawn, agents revise their plans, and information about available capacity changes. A successful execution may modify the next graph and the future production conditions. Graph transformations and event histories therefore belong to different aspects of the model: a graph grammar may describe possible structural transitions, while an operational semantics must state how each transition interacts with dated inventories and commitments. Formally elegant cycles can fail under joint resource constraints. Conversely, alternative routes and maintenance subcycles can create resilience that a single-path fragility calculation would not reveal.

The possibility of endogenous fragility also motivates the inquiry. A long chain of resource transfers may encounter multiple opportunities for technical failure, norm disagreement, or a justified refusal

to cooperate. Conditional models can assign declining effective influence to longer walks; the number of possible walks may simultaneously increase, making aggregate higher-order effects sensitive to the balance between proliferation and attenuation. This question motivates finite summaries, certificates for local feasibility, and approximation methods. It cannot be answered by grammar alone. In particular, a low-weight path in a computational approximation may still contain an unresolved entitlement or a materially important bottleneck. The companion analysis of distributed monetary dissolution investigates how these vulnerabilities affect an economy’s reliance on monetary coordination; the present paper focuses on how candidate circuits can be constructed and recognized without silently presupposing a welfare or justice theorem.

The authorship of the proposed AI application should also be distinguished from its formal sources. Automata, Petri nets, temporal networks, and graph rewriting are established areas with a substantial literature. The potential contribution lies in selecting and combining formal primitives appropriate to AI-mediated production: typed input-output events, dated deliveries, differentiated resource powers, commitments, recursive subcircuits, regenerative dependencies, and historically revisable institutional interfaces. A well-formed production-circuit language would make certain assumptions inspectable and permit the construction of counterexamples. Whether it improves coordination costs or protects rights in a real AI-operated economy remains an empirical and institutional research question.

Under an optimistic premise, AI systems with extensive shared production records and strong capacities for dynamical inference could identify candidate circuits, evaluate local constraints, update feasible schedules, and maintain reusable summaries of recurring resource dependencies. Such capabilities could reduce reliance on exhaustive enumeration and may help coordinate production without a universal purchasing medium in some settings. They would not remove finite resource constraints, confer valid authority over another operator’s inputs, or make contested normative interpretations uniquely decidable. The relevant research programme therefore requires local verification, explicit uncertainty, preserved rights provenance, and procedures for reopening a coarsened description when a claim or production bottleneck has been overlooked.

The remaining sections develop a typed relational vocabulary for production events and claims; establish the fixed-graph baseline for simple cycles and repeated closed walks; introduce composition rules for serial, parallel, nested, and alternative production processes; and examine material and temporal semantics. Subsequent analysis considers the expressiveness of different representation classes, graph transformation, overlapping-value attribution, dynamic fragility, and institutional evaluation. The conclusion will identify which features may be represented by a finite grammar, which require additional semantic or empirical input, and which remain questions of legitimacy beyond formal derivation. This separation preserves the possibility that advanced AI could improve recognition of complex productive relations while maintaining the limits of formal grammar as a description of prospective rather than necessarily just economic life.

2 Value Circuits as Typed Relational Structures

The purpose of this section is to specify the objects from which a grammar of productive value circuits may be constructed. In a production network operated largely by highly capable artificial-intelligence (AI) systems, a relation between two operators may refer to a physical delivery, a transformation requiring several inputs, a conditional commitment, a permission to withdraw resources, or a shared capacity constraint. These relations have different composition rules and different evidential requirements. The section therefore develops a typed relational vocabulary, specifies the identity and accounting of productive events, and illustrates how a candidate circuit can be represented without conflating physical circulation, obligations, and normative evaluation. The construction is a proposed analytical schema, informed by established work on Petri nets, temporal networks, and heterogeneous network layers [15, 7, 10]. It is not offered as a calibrated economic model or a claim about the eventual legal status of AI operators.

Table 2: Typed components of a candidate AI production circuit and the questions their records support.

Component	Represented information	Independent validity question
Operator / facility	Acting role; controlled or shared productive capacity	Who acts, and which capacity is actually available?
Resource / stock	Type, quantity, unit, custody, site and date	Is the stated input available at the required time?
Event occurrence	Unique identity, transformation, place and execution interval	Can all required inputs and capacities be used together?
Delivery	Dated transfer and receiving location	Is the right output received before its use?
Commitment	Parties, object, conditions, validity and discharge record	What obligation remains, and for whom?
Permission	Scoped power to use, withdraw, transfer or modify an interface	Is the proposed action authorized on these terms?
Regeneration state	Facility, ecological or organizational condition over time	Can subsequent production retain its required capacity?

2.1 Analytical Objects and Type Boundaries

The initial domain comprises distinguishable sets of operators $\mathcal{O}$, productive facilities $\mathcal{F}$, resource types $\mathcal{R}$, spatial sites $\mathcal{L}$, event types $\mathcal{T}$, event occurrences $\mathcal{E}$, and dated commitments $\mathcal{C}$. A resource type $r \in \mathcal{R}$ has an associated measurement domain and unit. Grain mass, electrical energy, machine-hours, storage capacity, and an ecological-condition indicator cannot be added as though they were quantities of one commodity. Their interactions are represented through stated technological transformations or through separate evaluative mappings, whose legitimacy remains to be examined.

An *operator* is a node capable of planning, executing, reporting, or coordinating a production-related action. The assignment of an operator to an AI system is an assumption about operational roles. It does not attribute legal personality, property title, welfare interests, or moral standing to that system. A *facility* is a capacity-bearing object, such as a food-processing line, battery system, field, storage location, or repair machine. A facility may be managed by one operator, jointly accessed by several operators, and subject to permissions held or delegated by another entity. An *interface* is a specified point of access or transfer between roles, facilities, or locations. Keeping these categories separate prevents a machine, its operator, and the holder of a power to exclude others from being silently identified with one another.

A resource is represented by a typed quantity associated with a location or storage account, a time, and an identified custody or control arrangement. A record of eight kilograms of grain at a warehouse is therefore distinct from an unsited promise to deliver eight kilograms of grain next week. A promise may be economically important even before the required stock exists, whereas the current resource record supports only the availability described by its own timestamp and access conditions. The distinction is particularly relevant in an economy where AI systems can perform nearly any kind of work yet still encounter finite raw materials, storage constraints, and obligations that have not been discharged.

For purposes of the present grammar, an *event type* is a template for a possible activity, while an *event occurrence* is a historically identifiable instance. Milling grain, transporting food, replacing a battery, and revising a resource authorization are different event types. Two milling operations performed by the same AI at different times remain different occurrences even if their technological input–output signatures match. The distinction is essential when a production event belongs to several candidate circuit descriptions: multiple descriptions do not create multiple physical operations.

Table 2 gathers the principal types and their validation responsibilities. It also anticipates the division between the grammatical rules developed later and the operational and institutional semantics examined in subsequent sections.

2.2 Multi-Input Events and Productive Incidence

A simple directed edge between two AI operators is useful for presenting a delivery, but it is insufficient to represent a production process that requires several simultaneous prerequisites. A grain-processing activity may need grain, electricity, machine capacity, valid access permission, and a delivery deadline. The underlying formal object is accordingly an event with typed input and output ports, naturally represented by an incidence structure or a directed hypergraph. Petri-net methods provide a well-developed example of how concurrent activities and enabling resources can be distinguished from the sequence of their execution [15]. The account adopted here is more modest: it borrows the event-and-resource distinction while leaving the detailed transition system for later development.

A proposed event record is defined in Equation 1.

$$e = (\text{id}_e, \tau_e, o_e, \ell_e, [s_e, f_e], I_e, O_e, K_e), \quad e \in \mathcal{E}. \quad (1)$$

Here id_e is an immutable occurrence identifier; $\tau_e \in \mathcal{T}$ is its event type; o_e is the executing operator; ℓ_e is its place; $[s_e, f_e]$ is the scheduled interval; I_e and O_e are finite *multisets of typed input and output requirements*; and K_e describes capacity requirements or relevant facility interfaces. Each port specifies at least a resource type, a quantity with units, a required or available time, and the location at which the resource is used or produced. Other fields, including quality tolerances and provenance references, can be introduced when a specific research question requires them.

Input and output lists are multisets because several batches of the same type can originate from different sources or have different expiry times. Two grain inputs of four kilograms each can amount to eight kilograms of usable grain only if the applicable quality, location, authorization, and time constraints permit aggregation. Likewise, an event with a shared output cannot be duplicated simply because the event is traversed by two grammatical derivations. Both derivations refer to the same occurrence identifier, and the inventory may be debited only according to a feasible allocation of the available output.

We distinguish *productive incidence* from *operator adjacency*. Productive incidence associates one event with its multiple input and output ports. Operator adjacency is a projection that records, for instance, that an output originating under operator A is delivered to a facility run by operator B . Different productive incidence structures can project to the same operator graph. In particular, the projection may omit a missing energy input, a shared processing machine, or an institutional permission. This non-injectivity explains why the existence of an operator-level closed walk is a weak test of productive circuit feasibility.

The separation of event type from event occurrence also allows structural reuse. A type-level rule might state that eight units of suitably specified grain and three units of electrical energy permit one standardized processing batch to produce food and residue under stipulated technological coefficients. The rule describes a *potential transformation*; executing it creates a new occurrence with a dated ledger record. Whether the transformation coefficients fit any real facility is an empirical question. The present paper uses such coefficients solely in declared thought experiments.

2.3 Dated Stock and Shared-Capacity Semantics

Typed syntax allows a candidate schedule to be written, but a feasible history requires a resource account that constrains all proposed occurrences jointly. For an illustrative discrete-time model, let $x_{r\ell}(t)$ denote the available stock of type r at site ℓ at the beginning of period t . Let $z_e(t)$ indicate whether an event occurrence is executed within that accounting period, and let $a_{e,r\ell}^+$ and $a_{e,r\ell}^-$ denote the quantities made available and consumed in the account by the event. The stock bookkeeping identity is given in Equation 2.

$$x_{r\ell}(t+1) = x_{r\ell}(t) + \sum_{e \in \mathcal{E}} a_{e,r\ell}^+ z_e(t) - \sum_{e \in \mathcal{E}} a_{e,r\ell}^- z_e(t) + u_{r\ell}(t) - d_{r\ell}(t). \quad (2)$$

The terms $u_{r\ell}$ and $d_{r\ell}$ record externally supplied and withdrawn quantities under an explicitly chosen system boundary. The expression is an accounting identity for specified resource accounts; it does not

assert that value, energy, biomass, or ecological capacity can be converted into one another without losses. The event coefficients must encode actual yields and by-products, and an adequate physical model would include relevant waste, dissipation, transport losses, and transformations of environmental stocks. Nonnegativity and dated input availability are additional feasibility requirements, rather than consequences of writing down the identity.

Multiple candidate circuits may use the same scarce facility. If $K_f(t)$ is the usable capacity of facility f during period t , and $k_{ef}(t)$ is an event's consumption of that capacity in the same unit, the joint constraint is stated in Equation 3.

$$\sum_{e \in \mathcal{E}} k_{ef}(t) z_e(t) \leq K_f(t), \quad f \in \mathcal{F}. \quad (3)$$

For this comparison the capacity coefficients are assumed additive across the period. More detailed production systems may need instantaneous scheduling, nonpreemptive processes, setup costs, or inventory-dependent throughput. A candidate circuit may satisfy each constituent event's stand-alone requirements yet violate Equation 3 when several circuits operate together. Consequently, material feasibility belongs to the *joint event history*, and cannot generally be inferred by validating candidate loops one at a time.

Temporal order introduces further restrictions. A product delivered at time t_2 cannot enable an event scheduled to begin at $t_1 < t_2$. Storage and perishability may constrain whether a technically compatible product remains usable later. Temporal-network scholarship supplies a general framework for distinguishing static connectivity from time-respecting paths [7]. For AI production relations, the event record must additionally preserve processing intervals, transformation durations, and the periods in which a permission or commitment is valid. A path can be topologically closed while its alleged sequence of deliveries is temporally impossible.

2.4 Commitments, Permissions, and Institutional State

The ability to execute a physical transformation is distinct from the ability to claim the resources it would use. We therefore define commitments and permissions as relations whose lifecycle and validity are recorded independently from the physical event ledger. This separation makes it possible to investigate credit and resource coordination without presupposing either monetary payment or independent AI property ownership.

A *commitment* record identifies the issuer, beneficiary or authorized claimant, promised resource or activity, validity window, activation conditions, relevant priority rules, and the events or decisions that count as discharge. The commitment may be conditional on crop yield, receipt of energy, successful inspection, or a negotiated change to the delivery schedule. Its status can be prospective, active, partially fulfilled, disputed, discharged, expired, or otherwise governed by the applicable institutional rule. A ledger entry reporting a delivery cannot by itself settle whether the legal or contractual obligation has been discharged; that depends on the previously applicable terms. Likewise, offsetting two resource-specific commitments requires an authorized procedure and compatible obligations. Numeric equality across records is insufficient to confer discharge.

A *permission* or institutional power is more narrowly scoped than an unrestricted right to a resource. The model may record permission to operate a machine, withdraw a specified amount from a store, transfer a batch to another facility, withhold access under stated conditions, certify a production record, or modify a shared allocation protocol. Each permission has an identified source or institutional rule, relevant holders, temporal validity, applicable quantities and procedures of contestation. Some permissions enable production; others allow challenges, suspend an alleged authorization, or alter the future topology of permitted interactions. The fact that an AI can calculate a desirable transfer does not establish that it can authorize the transfer.

In the proposed structure, both a dated physical history and an institutional state are needed to evaluate whether an event may proceed. A production planner may find a resource-feasible chain, while the permission ledger shows a revoked warehouse withdrawal authorization. Conversely, an operator may

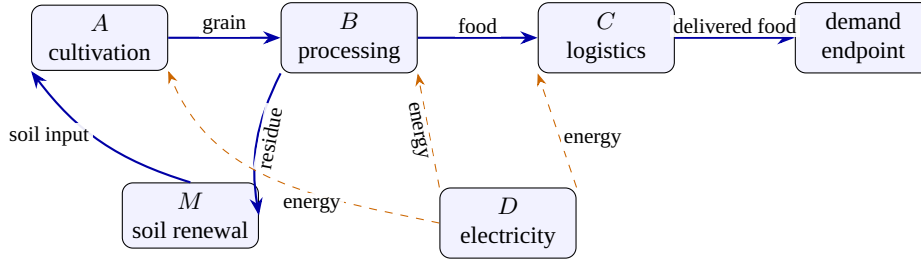


Figure 2: Typed material-flow projection of a hypothetical AI-operated food system. Solid connections represent resource or product movements; dashed connections denote energy inputs. The residue-renewal pathway recurs across periods and does not make the food-delivery pathway materially closed.

possess a valid claim against a supplier that is presently unable to deliver the resource. A formal grammar can describe the requested or promised sequence in each case; feasibility and rights validity remain separate semantic questions. The allocation of loss when an authorized delivery fails is a further question of responsibility and governance, not a conclusion derived from port matching.

Normative disagreement also has an identifiable place in the structure. An operator can register a contestation or refusal even when an automated plan reports adequate resources and apparent contractual conformity. Such an intervention may be justified by an overlooked right, an invalid delegated power, or a competing conception of fair allocation; it may also result from a mistaken inference or strategic behavior. The grammar can retain the record of the intervention and the resulting change in permissible actions. A normative theory would be required to evaluate its merits. In particular, deletion of a claim from a computational summary cannot be treated as evidence that the underlying obligation or entitlement has ceased to exist.

2.5 A Food Circuit with a Regenerative Subcircuit

Consider a constructed network of five operators. Operator *A* cultivates grain; *B* processes grain into food; *C* transports the food to an external demand endpoint; *D* supplies electrical energy; and *M* processes an organic residue into a soil amendment used in a later agricultural period. The operators are role labels for a hypothetical AI-mediated productive system, not independent owners of the associated resources. Permissions and any property claims remain separate from the operating assignments. Figure 2 shows the candidate material routes. The diagram is intentionally a projection of richer typed event data: its edges alone do not carry all timing, yield, ownership, or capacity conditions.

Suppose one stipulated production period includes a harvest of eight kilograms of grain, a processing batch yielding seven kilograms of food and one kilogram of residual biomass, and a logistics operation that delivers the seven kilograms of food to its destination. For the electricity account, assume that cultivation, processing, and logistics require two, three, and one kilowatt-hour respectively, so the three activities jointly require six kilowatt-hours. The figures are illustrative accounting assumptions, not physical parameters estimated from an agricultural system. For the residue path, the soil-renewal event receives the residual biomass and, after an explicitly modeled transformation and delay, produces an amendment whose usable yield and effect on soil capacity must be supplied by a separate technological or ecological assumption. It would be incorrect to equate one kilogram of residue automatically with one kilogram of effective soil amendment.

The example separates several senses in which the activity might be called a circuit. The operator path through *A*, *B*, and *M* returns a resource to agricultural production, although the returned resource is of a different type and may become available only in the next period. The path through *B*, *C*, and the demand endpoint delivers useful output beyond that regenerative subcircuit. Electricity produced or distributed through *D* enters several productive events but does not thereby circulate back to *D* in the same physical form. An economic assessment may find that agricultural capacity is regenerated by the amendment, a maintenance action, or a further resource input; it cannot infer regeneration solely from

the visual closure of the residue arrow.

The same construction illustrates joint capacity and rights constraints. If D can provide only five kilowatt-hours during the relevant period, the stipulated six-kilowatt-hour schedule is infeasible unless some operation is changed, deferred, or supplied from an additional source. If B lacks authorization to receive the grain batch, adequate grain inventories do not cure the institutional defect. If the composting interval ends after the next planting deadline, the regenerative input arrives too late for the intended recurrence. If the original claim to seven kilograms of food is discharged only upon accepted delivery, the processing event alone does not close the associated obligation. These are distinguishable failure modes of one graphically plausible structure.

Finally, consider a sixth event that uses the same available electricity to repair cultivation equipment. It may participate in an overlapping productive history that supports future grain production. Describing the soil-renewal subcircuit and the equipment-maintenance subcircuit separately could make each appear feasible, while their simultaneous execution exceeds the capacity of D . Their event identities and their common electricity account permit a joint test. Their projected paths can overlap or be nested without producing additional supplies merely because the grammar admits multiple derivations.

2.6 Circuit Identity, Recurrence, and Provenance

The phrase *value circuit* will refer to a candidate arrangement of productive events and relations for which a relevant form of recurrent return, replenishment, or continued productive participation is under consideration. Its substantive assessment requires choosing what is to be regenerated: an inventory, a facility's usable capacity, an ecological condition, an enforceable claim, or a more expansive set of social production conditions. Those objects are not interchangeable. In particular, an operator-level closed walk may be used as a shorthand for a proposed recurrence, while a temporally recurrent productive history needs dated transitions across multiple states. The stronger property is established only when the relevant inputs and regenerative conditions have been checked at the appropriate times.

We will therefore use three nested descriptions. A *candidate syntactic circuit* is a well-formed expression or structure under the stated composition rules. A *materially feasible circuit history* is an interpreted sequence of event occurrences satisfying the declared resource, capacity, spatial, and temporal guards. An *institutionally admissible circuit history* additionally satisfies the applicable authorization and claim-state requirements. Whether such a history is generatively valuable, distributively fair, or otherwise legitimate is an evaluative question with its own provenance and standards. These labels describe increasing sets of conditions without presupposing that every valid candidate should be implemented.

A circuit description may have more than one grammatical derivation. A given event occurrence may appear in multiple overlapping proposed circuits, or two derivations may differ only in the insertion of a repeated subwalk. The appropriate invariant for resource bookkeeping is the set of identified executed occurrences and their actual quantities, not the number of derivations in which they appear. This distinction accommodates the unbounded family of repeatable closed walks in a finite graph while preventing the mistaken conclusion that infinitely many syntactic repetitions entail infinitely many realizable products. Later sections will examine the regular-language baseline for such walks, then the additional expressive demands introduced by typed production, changing topology, and guarded execution.

To support provenance, each event record can link to the input-batch identifiers, the producing events for those batches, the facilities it used, the operative permission versions, and any commitments that its occurrence purportedly satisfies. A provenance link is an audit relation, not a complete attribution of value or a proof of who ought to receive the benefits. Several histories may use a common upstream research discovery, infrastructure investment, or ecological resource; the trace of those dependencies matters even when their contributions lack a unique commensurable measure. Such records can also register disagreement about the correct interpretation of the same observed history. The study of economic benefit allocation requires explicit additional assumptions rather than a count of paths.

2.7 Scope of the Typed Construction

The present definitions establish a vocabulary for specifying possible AI-mediated productive relations. They make four distinctions available for the rest of the paper: an operator differs from a resource-right holder; an event type differs from its executed occurrence; a graph path differs from an input-complete, time-respecting production history; and a ledger entry differs from an enforceable entitlement or a justified distribution. Those distinctions can be implemented using different formal representations. Directed hypergraphs, typed event traces, Petri-net-style transitions, or multilayer relations may each be useful under particular objectives; no single representation is adopted here as the ontological structure of economic life.

Several questions remain open at this stage. The choice of primitives is historically and problem dependent, especially when the production network includes new resources or unfamiliar AI roles. Technological coefficients and capacities require evidence before empirical claims can be made. Institutional semantics depend on the governing legal and contractual order and may change as production relations evolve. The normative evaluation of value return also requires a justified treatment of agents, ecological dependencies, common contributions, contestable authority, and the recognition of heterogeneous forms of value. Relevant discussions of generative justice motivate attention to such returns without constituting a derivation of the proposed event grammar [4].

Within these limits, the typed construction supplies an explicit input to subsequent formal inquiry. Section 2 defines what is counted as an event and what remains an independent constraint; the next section will study finite-graph cycles and recursively composed closed walks on the resulting projections. Later grammar rules will need to preserve resource types and event identity when composing serial, parallel, alternative, and recurrent histories. Such a separation allows advanced AI systems, in the hypothetical shared-data setting, to search for candidate circuits and test specified constraints while retaining uncertainty and unresolved institutional claims. It also helps explain why the discovery of a closed walk, however computationally efficient, cannot by itself establish productive regeneration, a reduction in monetary dependence, or justice.

3 Finite-Graph Cycles and Recursive Closed Walks

The role of this section is to establish the finite-graph baseline against which a generative grammar of productive value circuits must be assessed. The analysis distinguishes simple directed cycles from closed walks that revisit vertices, proves that return walks on a fixed finite graph form a regular language under a finite edge alphabet, and examines the contrast between compact recognition and potentially large enumeration. A five-operator example exhibits arbitrarily repeated subwalks without generating additional material production by itself. The final discussion specifies the point at which a graph-theoretic account must be supplemented by the typed events, dated resources, commitments, and permissions defined in Section 2. The mathematical constructions are established graph and automata results used here as analytical baselines; their interpretation as value-circuit descriptions is a proposal of the present paper [9, 8].

3.1 Directed Graph Projections and the Identity of a Walk

Let $G = (V, E, s, t)$ be a finite directed multigraph, with finite vertex set V , finite edge set E , and functions $s, t : E \rightarrow V$ specifying the source and target of an edge. Distinguishable parallel edges are retained as different members of E , since the same pair of operators may be connected by different delivery or coordination interfaces. Vertices can represent AI-operated productive interfaces in a particular projection, but an operator, a production facility, and a holder of resource-disposal powers remain distinct typed objects in the more complete structure. A graph is therefore a chosen view of the production system rather than its exhaustive economic description.

A directed walk of length L is an alternating sequence of vertices and edges for which the target of each edge equals the source of the next. A *closed walk based at* $a \in V$ begins and ends at a . The empty

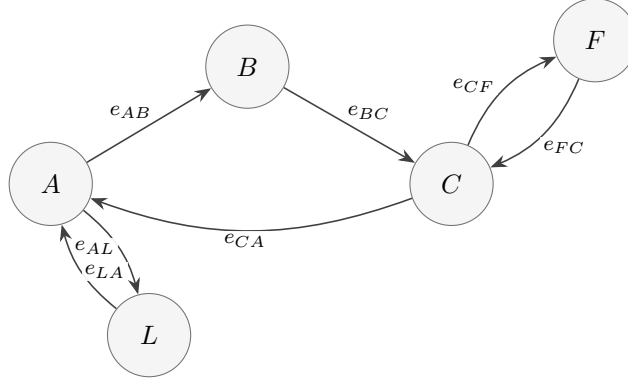


Figure 3: A finite projected AI-production network with an agricultural–processing–storage return route, a logistics excursion based at A , and an optional maintenance subwalk based at C . Repeating graph excursions is syntactically permitted; feasibility and actual production depend on independent typed event and resource records.

walk at a has length zero and is useful as the identity of concatenation, although it performs no material operation. A *simple directed cycle*, also called an elementary directed circuit, is a positive-length closed walk with no repeated vertices other than its coincident initial and terminal vertex; a self-loop can be treated as a simple cycle of length one. A cycle may be regarded up to cyclic change of its starting position, whereas a return walk based at a has a distinguished start and end. These conventions matter when comparing counts.

The set of simple directed cycles of G is finite because a simple cycle uses each vertex at most once before closure. By contrast, the set of positive-length closed walks based at a is infinite exactly when at least one positive-length closed walk at a exists: any such walk can be concatenated with itself arbitrarily many times, producing words of different lengths. Cyclic components disconnected from a , or components that cannot return to a , do not create return walks at a . This is a structural statement about the finite graph; it does not imply any repeated physical execution, unlimited material stocks, or a renewable production schedule.

An edge-occurrence word records a walk using the actual members of E as letters. Such a word preserves the interface sequence selected by the projection while remaining agnostic about event identifiers, quantities, permissions, and timing. Two walks can share an edge letter while referring, after semantic interpretation, either to distinct dated executions of an event type or to one already recorded physical occurrence. The distinction is necessary because grammatical repetition is an operation on descriptions, whereas material repetition requires new available inputs and feasible transitions.

3.2 Elementary Cycles and Recursive Excursions in an AI Network

Consider the projected five-interface network in Figure 3. Its interface symbols A, B, C, L, F stand for, respectively, an agricultural operation, food processing, a coordination and storage interface, a logistic return interface, and a maintenance interface. The letter L distinguishes the logistics interface of this purely projected graph from the electricity supplier D in the later material-accounting examples; identical letters across independently constructed projections would otherwise obscure their different types. The arrows record candidate directed dependencies or transfers at this level of abstraction. In particular, an arrow labelled by an ordered pair such as e_{CF} is one interface relation and should not be construed as a sufficient specification of an authorized material transfer.

The network supports the simple return walks $A \rightarrow B \rightarrow C \rightarrow A$ and $A \rightarrow L \rightarrow A$. It also supports a longer walk $A \rightarrow B \rightarrow C \rightarrow F \rightarrow C \rightarrow A$, which revisits C before returning to A . The length-five walk contains the shorter excursion $C \rightarrow F \rightarrow C$ and can be extended by any number of such excursions. With

the edge names shown in Figure 3, the resulting family is defined in Equation 4.

$$w_k = e_{AB}e_{BC}(e_{CF}e_{FC})^k e_{CA}, \quad k \in \mathbb{N}_0. \quad (4)$$

Equation 4 defines infinitely many distinct edge words, with lengths $3 + 2k$, although the graph has only five vertices and seven directed edges. Moreover, any complete return word from A can be repeated, and the A – L excursion can be inserted between complete return episodes. Such operations generate complicated descriptions using a small, finite repertoire of edges.

The recursion in Equation 4 has a limited economic interpretation. A syntactic insertion of $e_{CF}e_{FC}$ may stand for an additional maintenance request, a redundant itinerary, or a repeated interface-level dependency. Its interpretation as a newly performed repair requires an occurrence record, a usable facility, electrical and material inputs, an execution interval, and valid authorization. If the same physical repair is described once within a soil-regeneration circuit and again within an equipment-return circuit, the underlying repair must be counted once in the resource ledger. The graphical example makes repeated subcircuits available for analysis while leaving the semantic validation explicit.

3.3 Regularity of Return-Walk Languages on Finite Graphs

The existence of arbitrarily long return words does not require an unbounded syntactic memory. A finite automaton can recognize the return walks of any fixed finite directed graph by storing only the current vertex. This construction gives the precise baseline against which more expressive grammar proposals must be compared, in accordance with standard automata theory [8].

Proposition 3.1. *Let $G = (V, E, s, t)$ be a finite directed multigraph, and let $a \in V$. With each edge treated as its own alphabet symbol, the language of positive-length directed closed walks beginning and ending at a is regular.*

Proof. Construct a deterministic finite automaton with alphabet E , state set $V \cup \{\perp, q_{\text{init}}\}$, initial state q_{init} distinct from the graph vertices, and sole accepting state a . On the first input edge e , the automaton moves from q_{init} to $t(e)$ if $s(e) = a$, and to the absorbing failure state $\perp$ otherwise. From a vertex state v , reading e moves to $t(e)$ when $s(e) = v$ and to $\perp$ otherwise; the failure state absorbs every input. Induction on the nonempty word length shows that a word is accepted exactly when it names a directed walk starting and ending at a . The empty word remains at the nonaccepting initial state and is rejected. Thus the specified positive-length closed-walk language is recognized by a finite automaton. $\square$

Proposition 3.1 is independent of the number or length of accepted walks. The explicit machine recognizing positive-length closed walks uses $|V| + 2$ states, including its distinct nonaccepting initial state and absorbing failure state; each input edge either advances to its target vertex or causes rejection. For a particular fixed network, a membership query can therefore be evaluated by a pass through the proposed edge word. This does not establish that the network has a small list of cycles, that material feasibility can be checked without considering capacities, or that generating every accepted word is a finite task.

If the graph is given through an ordinary simple adjacency structure, an automaton may use destination labels rather than uniquely named edge symbols; the resulting language answers a different identity question when parallel resource interfaces exist. Retaining the edge alphabet is helpful for distinguishing a delivery channel from an authorization or other interface projected onto the same ordered pair. Still, neither alphabet captures an unbounded inventory counter without supplementary state or semantic guards. In particular, the regularity conclusion is about the *fixed projected graph and its edge words*, not about the language of physically feasible production histories under arbitrary numerical resource constraints.

3.4 Enumeration Growth and Compact Path Descriptions

Recognition of a specified walk and enumeration of every simple directed cycle are different computational problems. The latter has an output cost that grows with the number of cycles to be listed. Johnson’s

Table 3: Different objects and computational questions associated with cycles in a fixed finite graph.

Object	Representative method	Analytical limit
Simple directed cycles	Elementary-cycle enumeration	Potentially large output; finite for fixed G
Closed-walk words at a	Finite automaton for edge-word membership	Recognizes words; does not establish execution
Length-indexed return counts	Adjacency-matrix powers	Counts repeated paths, including shared edges
All-length formal walk statistics	Rational matrix generating function	Requires an interpretation for any economic weighting
Dated productive histories	Typed event ledger and joint feasibility guards	Needs resources, timing, authorization and event identity

algorithm provides a classical output-sensitive treatment of elementary circuits in finite directed graphs, with a bound of $O((|V| + |E|)(c + 1))$ for enumerating c elementary circuits and $O(|V| + |E|)$ space under its stated representation [9]. This result does not impose a polynomial bound on c as a function of graph size. For example, a complete loopless directed graph with n labelled vertices has $(n - 1)!$ distinct Hamiltonian directed cycles when cyclic rotations are identified; these are among its simple cycles. Enumeration can consequently become costly even before repeated closed walks are admitted.

A complementary question concerns the number of walks of a given length. Let M be the nonnegative integer adjacency matrix in which M_{ij} counts directed edges from i to j , including multiplicity. The length- L return-walk count based at a is given in Equation 5.

$$N_a(L) = (M^L)_{aa}, \quad L \in \mathbb{N}_0. \quad (5)$$

The claim follows by induction on L : matrix multiplication concatenates an admissible length- L walk from a to an intermediate vertex with one outgoing edge. Equation 5 counts graph words with multiplicity. It does not count unique physical events or economically independent contributions. The length-zero term equals one because the empty walk is included.

The complete sequence of return-walk counts also admits a compact formal generating function, stated in Equation 6.

$$H_a(z) = \sum_{L=0}^{\infty} N_a(L) z^L = [(I - zM)^{-1}]_{aa}. \quad (6)$$

Equation 6 holds as a formal power-series identity and as an analytic identity wherever the matrix Neumann series converges; in particular, $|z|\rho(M) < 1$ is sufficient, with $\rho(M)$ the spectral radius. The matrix inverse is a rational function of z . It provides a finite representation for an infinite family of walk counts when the underlying graph and edge alphabet remain fixed. It supplies no natural economic discount rate or normative valuation for repeated passages. Assigning such weights would require separate operational and interpretive assumptions.

Table 3 summarizes the distinct computational questions. The relevant description is selected according to the intended inquiry, rather than inferred from the presence of recursion alone.

3.5 Subwalk Decomposition and Ambiguity of Circuit Descriptions

A closed walk that revisits vertices can be shortened by removing a subwalk between consecutive occurrences of a repeated vertex; repeated application of this procedure identifies simple directed cycles. Such decompositions are useful for finding recurrent structures, but they are generally not a canonical account of economic production. The order in which repeated vertices are selected can change the intermediate descriptions. More importantly, a projected graph does not reveal whether apparently repeated edges refer to new executions, reused physical results, or independent obligations discharged by the same delivery.

The walk $A \rightarrow B \rightarrow C \rightarrow F \rightarrow C \rightarrow A$ in Figure 3 contains the maintenance excursion at C , yet the recurrence of C does not imply that the excursion generated a second food batch. A typed production event may require jointly supplied inputs: the processing interface could need both grain and electricity, while equipment maintenance draws from the same finite electricity capacity. Consequently, decomposing the operator-level walk into apparently independent cycles may discard a shared-capacity dependency that determines whether the whole schedule can execute.

Ambiguity also occurs when several candidate descriptions explain the same realized history. A resource-delivery event might appear in an agricultural return circuit, in an equipment-repair circuit, and in a more extensive supply-and-regeneration account. Each description may clarify a different dependency, but the event identifier and quantity appear only once in the joint ledger. It would be erroneous to equate the number of descriptions or their derivations with total economic value. The distinction is particularly important for future AI-assisted analysis, where high-volume path enumeration could otherwise give an apparently precise but multiply counted measure of contribution.

A more defensible use of subwalk descriptions is to identify interfaces at which alternative plans, permissions, reserves, or maintenance routines could intervene. For example, if an energy interface is removed, a family of proposed return paths may cease to be available; adding a replacement supplier may restore some return possibilities. Such path-level counterfactuals guide where operational verification should be undertaken. Questions of claim ownership, compensation, prior investment, ecological regeneration, and unfair exclusion remain beyond the closed-walk decomposition itself.

3.6 Recursive Languages and Operational Feasibility

The distinction between a regular language of walk words and a feasible history of production can be made precise without prematurely introducing a full graph-rewriting calculus. Let $\mathcal{L}_a^+(G)$ denote the language recognized in Proposition 3.1. Let $\mathcal{H}(\Theta)$ denote a set of dated event histories feasible under an explicitly declared collection Θ of resource stocks, transformation technologies, shared capacities, temporal rules, and authorization predicates. A projection π maps each eligible history into an edge-word or into a sequence of graph interface symbols when that mapping is defined. The language of projected feasible histories over a given observational horizon T is specified in Equation 7.

$$\mathcal{L}_{a,T}^{\text{feas}} = \mathcal{L}_a^+(G) \cap \pi(\mathcal{H}_T(\Theta)). \quad (7)$$

Equation 7 is a definitional separation of syntax from selected feasibility criteria. It assumes a fixed interface projection and a well-defined map π on the histories considered. It leaves unsettled whether the feasible language is regular under a particular class of resource guards. The temporal horizon can make the set of realizable traces finite when event counts and minimal durations are appropriately bounded, but an unbounded family of capacity and obligation states may require stronger recognition mechanisms. This is a question for the subsequent sections, not a consequence of graph recursion alone.

For example, the family in Equation 4 permits any nonnegative number of maintenance excursions at C . If each excursion consumes one nonrenewable unit of a finite repair input and the history has no replenishment event, only finitely many repetitions are materially feasible for a specified initial stock. If the repair input is restored, additional repetitions may be feasible, subject to the restoration event's inputs, dates, rights and shared capacity. If a permission to operate the maintenance interface expires, graph adjacency may remain unchanged although operational admissibility ends. These cases share the same simple regular expression at the graph level and differ in their operational semantics.

The resulting framework is deliberately compatible with several alternative mechanisms of value coordination. A feasible return walk might be financed by monetary payment, authorized by a reciprocal commitment, supplied through a common resource pool, or governed by a public allocation rule. The graph language alone selects none of these institutional forms. Likewise, the grammatical production of a closed walk cannot establish that the participating AI systems have independent legal personality, that a perceived distribution is fair, or that an interrupted route reflects wrongful conduct rather than a legitimate refusal.

3.7 Finite Structural Knowledge and the Next Grammar Construction

The finite-graph result provides a disciplined positive finding for the imagined AI-operated economy. Shared records and advanced AI recognition may allow participants to build finite summaries of repeated interfaces, test whether proposed edge sequences are well formed, and identify highly recurrent structural dependencies without enumerating all repetitions. The availability of a regular return-walk language is a useful computational property even where the number of admitted return words is infinite. It permits compact descriptions of a formal possibility space and supplies a baseline for incremental updates of network topology.

The limit is equally consequential. A finite graph projection discards the multiplicity of production inputs, resource units, dated commitments, changes in available facilities, and authorization states unless those properties are restored through a richer representation. When a network changes over time, the alphabet or transition relation itself may evolve. When an AI system must decide whether to execute a proposed loop, it needs joint feasibility checks, records of earlier occurrences, and governance procedures for disputed claims. Such demands motivate typed composition, graph transformation, and semantic guards rather than an inference that infinite syntactic recursion requires an inherently uncomputable language.

For the purposes of this paper, the result of Section 3 is therefore a bounded structural foundation: elementary cycles are finite in number on a fixed finite graph; return-walk families can be infinite; and those return-walk words remain regular under the stated finite edge encoding. Counts, recognition, execution, value attribution, and legitimacy remain distinct research objects. The subsequent construction will use the event types and provenance records of Section 2 to specify serial, parallel, alternative, nested, and recurrent composition rules. Those rules will generate candidate productive circuit structures, while separate semantic tests determine what the generated expressions can mean in an AI production network.

4 Generative Rules for Productive Circuit Composition

The present section constructs a finite repertoire of composition rules for candidate production circuits in highly automated AI–AI networks. The preceding sections established a typed event vocabulary and the regular-language baseline for return walks on a fixed finite graph. The new task is to generate structured descriptions containing multi-input transformations, shared resources, temporally separated operations, alternatives, and recurrent subcircuits. The method is a deliberately modest typed term grammar, accompanied by interface-matching conditions and a worked food-production derivation. Its expressions describe possible productive arrangements. Resource availability, legal or delegated authorization, realization of commitments, and the normative evaluation of value return remain separate semantic tests. The construction draws limited inspiration from event-oriented Petri-net representations and rule-based typed graph transformations; it does not reproduce either theory’s complete calculus [15, 5].

4.1 Production Signatures and Typed Ports

Let $\mathcal{R}$ be the collection of distinguishable resource types introduced in Section 2. A *port* specifies a resource type, a nonnegative quantity with its unit, a receiving or supplying interface, and any conditions needed for a proposed connection. A food batch measured in kilograms, a delivery of electricity measured in kilowatt-hours, and a machine-availability interval are therefore different interfaces. A claim to future grain or a permission to operate a mill is recorded in an associated commitment or authorization relation. Neither record is itself treated as an interchangeable physical port.

An atomic productive term e describes an *event template*, as distinct from the uniquely identified event occurrence of Equation 1. It has a finite multiset $I(e)$ of required input ports and a finite multiset $O(e)$ of proposed output ports. Its description also contains the operator and facility types, technological conditions, and any institutional predicates that would need to be checked before execution. When a template is instantiated at a time and site, it receives a fresh occurrence identifier and the other data required by the event record. Reusing a template never means reusing the same realized material output.

A proposed composite term P has an *external signature* and a collection of event-template positions. Its typing judgment is given in Equation 8.

$$\Gamma \vdash P : (I(P), O(P); \Sigma(P)). \quad (8)$$

Here Γ denotes the declared resource-type catalogue, quantity units, permitted interface forms, and available event-template descriptions. The multisets $I(P)$ and $O(P)$ contain the ports still exposed at the boundary of the composite, while $\Sigma(P)$ records its internal template positions and their formal identities. These positions become separate occurrence identifiers if and when an actual execution is admitted. A boundary signature is an accounting interface rather than a promise of production: its outputs are conditional on the required inputs, technologies, capacities, and authorizations.

One complication deserves attention at the level of syntax. A single electricity source may provide six kilowatt-hours to three consumers who demand two, three, and one kilowatt-hour respectively. Directly matching the same six-kilowatt-hour output port to each demand would create a fictitious eighteen-kilowatt-hour supply. A valid construction uses a resource-disaggregation event with an input of six kilowatt-hours and outputs of two, three, and one, or an equivalent explicitly constrained allocation relation. The total allocation and the identities of the resulting portions must remain traceable. Similarly, a resource with a capacity that can be used concurrently, such as a machine, is better expressed as a common facility account with a joint capacity guard than as three individually duplicated consumable outputs.

4.2 A Minimal Grammar of Circuit Expressions

The generative vocabulary contains seven constructor forms, displayed in Equation 9. The symbol $\mathbf{1}_I$ denotes an identity connection at an interface I ; e is a typed atomic event template; $\otimes$ denotes parallel placement; wire_β connects compatible internal ports; $\oplus$ denotes a choice of candidate alternatives; hide_J exposes only the boundary of an internally connected module; delay_Δ records a time displacement with an explicit storage or waiting interpretation; and rec_κ makes a guarded recurrent family available for finite unfolding.

$$P ::= \mathbf{1}_I \mid e \mid (P \otimes Q) \mid \text{wire}_\beta(P) \mid (P \oplus Q) \mid \text{hide}_J(P) \mid \text{delay}_\Delta(P) \mid \text{rec}_\kappa(P). \quad (9)$$

Each expression is meaningful only under a suitable typing judgment such as Equation 8. The presentation is a *schema for constructing candidate event networks*, not a claim that all these expressions constitute a context-free word grammar under every encoding. In particular, a wiring map operates on typed incidence relations, while recursion may unfold to families of event networks with different numbers of occurrences. Their formal-language classification depends on the representation selected for the resulting structures.

Table 4 states the immediate purpose and principal side condition of each constructor. The table distinguishes an interface-level rule from a later claim about physical execution.

A typed grammar is useful because its constructors express economically distinguishable questions. Parallel placement asks which activities may coexist. Wiring asks whether an identified output can be delivered to a specified input. Choice asks whether alternatives provide a declared service objective. Encapsulation permits a complicated subcircuit to be treated as one interface without discarding its provenance. Recurrence expresses a family of proposed repeated histories while leaving their material realization to a separate procedure. The grammar can thereby describe an unlimited set of structures using finitely many constructors, provided that the event vocabulary, admissible parameter ranges, and representation of types are defined for the study at hand.

4.3 Sequential Connections and Parallel Joint Production

For a candidate composite P , define a *wiring map* β that associates selected output portions with selected input portions. A well-typed map matches equal resource types and compatible quantities and units. It

Table 4: Constructors of the proposed productive-circuit grammar and their validation boundaries.

Constructor	Formal role	Necessary condition or qualification
Atomic event / identity	Introduce a typed event or a pass-through interface	Defined ports and units; identity produces no additional resource
Parallel placement	Combine proposed event modules without assuming an order	Distinct proposed event identities; shared facilities remain jointly constrained
Wiring	Match output ports to input ports	Compatible type, quantity, site, and time; each consumable output portion is used at most once
Alternative	Represent one selected production route	Compatible declared external service objective; alternatives are not jointly counted
Encapsulation	Expose a composite’s boundary while retaining internal links	Internal event and provenance records remain accessible
Delay	Connect activities across times or periods	Nonnegative duration and a separately testable storage or transformation process
Guarded recurrence	Generate finite repetitions of a possible module	Compatible recurrent interface and explicit stock, timing, and institutional guards

also preserves the directed producer-to-consumer orientation. Output splitting must have its own resource allocation record, and the same consumable portion cannot be linked to two different consuming occurrences. A proposed internal connection is valid at the grammatical level if these conditions hold; actual fulfillment additionally requires an event completion time preceding the receiving event’s consumption time, appropriate locations and permissions, and the relevant physical and institutional states.

The sequential use of one event’s output as another event’s input is therefore represented through parallel placement followed by a selected wiring map. This design accommodates branching and joint inputs more naturally than a single untyped concatenation symbol. Suppose that cultivation template T_A , executed by operator A , produces grain and the processing template B processes grain, while D produces the electrical energy required by both. A putative production chain may be written as $\text{wire}_\beta(T_A \otimes B \otimes D)$, where β explicitly links grain and allocated electrical-energy ports. The parallel constructor initially records distinct templates; the wiring constructor imposes their directed dependencies. Nothing requires the processing event and the energy-producing event to be represented as a simple chain of operator-graph edges.

Parallel production also entails joint constraints that cannot be discharged by typing alone. Suppose one candidate food module requires six kilowatt-hours during a production period and a proposed equipment-repair module requires a further two. Each module is individually compatible with an electrical supply capacity of six, but the composite exceeds that capacity if both are scheduled in the same period. The sum of requirements must satisfy Equation 3 for the relevant shared facility. Similarly, resource portions available only after a delivery window or under an expired authorization cannot be made jointly available by a syntactic parallel operator. Consequently, the term $P \otimes Q$ should be read as *candidate concurrent composition*, with physical coexistence dependent on global ledger and scheduling tests.

The parallelism distinction is informed by the established use of event nets to represent concurrent processes [15]. Its economic extension consists in attaching typed material accounts, custody and rights records, and recurrent-production objectives that an ordinary unlabelled process graph would omit. The present construction does not claim the full concurrency semantics, reachability results, or invariants of an established Petri-net formalism. Those results require selecting and formally defining a particular net system.

4.4 Alternative Production, Interface Substitution, and Encapsulation

The alternative constructor $P \oplus Q$ denotes a choice among candidates for a declared external objective. To make comparison meaningful, the candidates must produce a compatible boundary service, such as seven kilograms of processed food delivered within an admissible interval at a specified destination. Their internal resources, energy requirements, ecological effects, and authorization conditions may differ. Those differences become the grounds for the later feasibility and evaluative comparison. The syntax does not require their total costs to be commensurable, and it does not treat both alternatives as simultaneously executed.

For illustration, suppose the original processing template consumes three kilowatt-hours per batch. An alternative template B' is stipulated to deliver an equivalent quantity and quality of food while requiring two kilowatt-hours, perhaps through a technological improvement. Replacing B by B' in the food module changes its input requirements. If the alternative is available, properly authorized, and temporally compatible, the aggregate modeled demand of cultivation, processing, and logistics can fall from six to five kilowatt-hours. The additional infrastructure, development history, capital renewal, and ecological conditions supporting B' must be included before claiming superior long-run productivity or lower monetary dependence. The example is a constructed option with explicit assumptions, not an empirical claim that such a process exists.

An *interface substitution* is stronger than the mere juxtaposition of two alternatives. It replaces a submodule in a larger connected structure while preserving the external service contract and updating all dependencies introduced by the replacement. Its validity therefore requires checking the boundary port types, service quantities, specified tolerances, delivery conditions, facility needs, and relevant permissions. The substituted module may alter an upstream energy relationship, change the provenance of a downstream food batch, or move a contested authorization to a new intermediary. A replacement is successful at the grammatical level when the surrounding expression remains well typed. It is institutionally and normatively acceptable only after the respective additional checks.

Encapsulation through $\text{hide}_J(P)$ turns internally connected ports J into nonexposed links while retaining their full history and identifiers. This can simplify a large candidate network by presenting its food-service or equipment-service boundary without forcing every downstream analyst to enumerate the internal grain, electricity, and maintenance transformations. The operator is intentionally *not* an erasure rule for liabilities, externalities, or contested rights. A hidden commitment retains its issuer, claimant, status, and conditions of discharge, while a hidden ecological effect remains part of the relevant environmental account. A coarse description may support local planning, but it cannot certify the legitimacy of the underlying relations merely by reducing their visible complexity.

Rule-based graph transformation offers a technical point of comparison for such typed substitutions and local structural updates. Ehrig and collaborators develop a rigorous treatment of typed and attributed graph transformations, including application conditions and replacement systems [5]. The proposal here uses compatible intuitions of typing and structure-preserving replacement, without claiming that its production syntax already has the categorical axioms or concurrency properties of that framework.

4.5 Regenerative Feedback and Guarded Recurrence

A recurrent productive history contains a temporal relation between the conditions consumed in one period and those needed in a later period. A visually closed edge path may conceal the fact that the returned item has changed resource type, arrives too late, or provides insufficient regeneration. A rule for productive recurrence should accordingly specify an external recurrent interface, rather than identify a graph-theoretic return with the replenishment of the initial physical stock.

Let κ denote a proposed recurrence guard: it records the required types of carried state, admissible interperiod connections, and predicates against which an unfolding should be tested. The expression $\text{rec}_\kappa(P)$ defines the *family of finite candidate unfoldings* of module P . Each unfolding contains new event-template positions and, if realized, requires fresh occurrence identifiers. The identity unfolding contains no positive production and serves only as a syntactic base case. This constructor neither asserts

Table 5: Illustrative event signatures for an AI-operated food circuit; unlisted technologies and regeneration effects remain explicit model assumptions.

Template	Modeled consumable inputs	Proposed outputs or services
D (electricity)	External generation conditions	6 kWh electrical energy
S (allocation)	6 kWh electricity	Portions of 2, 3, and 1 kWh
T_A (cultivation by A)	2 kWh; agricultural conditions	8 kg grain; updated soil condition
B (processing)	8 kg grain; 3 kWh	7 kg food; 1 kg residual biomass
C (logistics)	7 kg food; 1 kWh	Delivered 7 kg food
M (soil renewal)	1 kg residue; additional renewal conditions	A later amendment of a different resource type

convergence to a steady state nor licenses an actual infinite execution. A sustained recurrent process would require, among other things, acceptable inventories at each period, regenerative-capacity thresholds, bounded or financed degradation, available shared facilities, authorized transfers, and appropriate provision for unmodeled external resources.

A time-delay operation delay $_{\Delta}$ can expose a resource or capacity change in the next period, but it cannot create storage at zero material or institutional cost. For example, a soil-amendment input prepared from one kilogram of organic residue may become available only after composting and transport. The amplitude and timing of any soil-capacity regeneration are separate technological assumptions; an input of one kilogram of residue cannot simply be renamed one kilogram of restored soil capacity. In an appropriate event description, the soil-renewal event has its own inputs, outputs, delay, and site. Any recurring connection from soil renewal to the next agricultural event must respect those interfaces.

Guarded recurrence may also include repair excursions, such as the maintenance subwalk studied in Equation 4. Formally, arbitrarily many such excursions can be inserted into candidate words. Materially, each execution draws on real repair inputs, energy, time, and equipment. A finite reserve can make only finitely many positive repetitions possible during a declared horizon. Even when repair materials can themselves be regenerated, overlapping demand for the same electrical capacity may prevent concurrent execution. Those cases explain why grammatical recursion is analytically useful but not itself a claim of unlimited output or a proof that a production network can dispense with monetary coordination.

4.6 A Worked Derivation of a Food-and-Soil Circuit

The event signatures in Table 5 specify a compact construction using the same roles as Section 2.5. The operator roles are A for cultivation, B for processing, C for logistics, D for electricity, and M for soil renewal. The cultivation-event template is separately denoted T_A so that H remains available in later sections for a time horizon or a dated event history. The numerical quantities apply to one stipulated production period. In addition to the listed consumable ports, each event depends on declared facilities, locations, valid permissions, and any necessary environmental or technological inputs. The table makes no assertion that the model is calibrated to an actual agricultural facility.

First place the six event templates in parallel, recording them as distinct prospective activities. Second, wire the electricity generator to the allocation event, the three allocated portions to their intended users, the grain to processing, the food to logistics, and the residual biomass to renewal. Third, encapsulate only those internal material-transfer ports that have valid matches. The successive candidate expressions are given in Equation 10.

$$\begin{aligned}
 P_0 &= D \otimes S \otimes T_A \otimes B \otimes C \otimes M, \\
 P_1 &= \text{wire}_{\beta_{\text{food}}}(P_0), \\
 P_2 &= \text{hide}_{J_{\text{internal}}}(P_1).
 \end{aligned} \tag{10}$$

The declared wiring map β_{food} matches only ports with compatible resource identities, units, quantities and producer–consumer direction. Its grain link is between T_A and B ; its food link between B and

C ; its residue link between B and M ; and its energy links are mediated by S . The external boundary of P_2 retains the delivered-food service, all required exogenous productive conditions, and the dated amendment output. Encapsulation does not turn the food delivered to final demand into a return of food to cultivation. The candidate recurrent path instead connects the soil-amendment output, after an explicit delay and appropriate effectiveness test, to agricultural capacity in a later production period.

A further expression $\text{rec}_\kappa(P_2)$ can generate descriptions of successive candidate periods. Its guard κ must specify that the *soil-amendment boundary port* is connected through an explicit delay operation to the relevant soil-condition input in a later period; it must also identify necessary exogenous inputs and separately dated instances of the six templates. Delaying one regenerative port does not delay or regenerate the entire productive module. If an authorization to use the field expires, or electricity drops to five kilowatt-hours without an alternative, a syntactically derivable subsequent period may be infeasible. A better efficient-processing alternative may restore the electricity balance only under its own technological, historical and institutional assumptions.

The worked derivation is more informative than a one-letter closed-walk string because it distinguishes heterogeneous outputs and jointly required inputs. It is also intentionally weaker than a full state-transition semantics: it generates a testable structural proposal and identifies the next checks. Section 2 supplies the stock and joint-capacity accounts that can be applied to an instantiated history, while the next section will examine temporal, material and renewal constraints in greater detail.

4.7 Boundary Accounting and Shared Event Identity

A well-typed wiring operation can support a limited accounting invariant. Fix a resource type r and one proposed finite composite with a set of distinct event-template positions. Assume that each internal consumable output portion is connected to at most one input portion, each connected pair has equal quantity in the same resource unit, and any allocation or splitting is represented by declared events. Let $o_{e,r}$ and $i_{e,r}$ denote the quantities recorded at the output and input ports of each event position for resource type r . Let O_r^{ext} and I_r^{ext} denote the corresponding total external boundary quantities. Under these bookkeeping assumptions, Equation 11 states a finite-composite identity.

$$\sum_{e \in \Sigma(P)} (o_{e,r} - i_{e,r}) = O_r^{\text{ext}}(P) - I_r^{\text{ext}}(P). \quad (11)$$

Proposition 4.1. *For a finite composite with distinct event positions, complete classification of its ports as wired internal or exposed external, uniquely matched consumable internal portions, and exact type-and-quantity matching, the boundary accounting identity in Equation 11 holds for each recorded resource type.*

Proof. Sum the recorded output quantities and subtract the recorded input quantities of the same resource type over the finite set of event positions. Each internal matched portion occurs once as a positive output and once as an equal negative input. All these paired terms cancel. The unmatched output and input portions are, by the completeness assumption, exactly those exposed at the composite boundary. Their signed total is the right-hand side of Equation 11. $\square$

Proposition 4.1 is an accounting consequence of the stipulated wiring discipline. It does not establish mass conservation across different resource types, energetic efficiency, physical feasibility, or economic value conservation. The transformation of grain into food and residue changes the quantities present in the individually typed accounts according to the event specification. A model of physical conservation must separately track any material and energetic losses and the conversion of resource kinds. Likewise, an accounting identity at the composite boundary does not demonstrate that the source event actually occurred or that an unperformed delivery has discharged a claim.

An additional subtlety concerns *multiple derivations of one realized history*. The set $\Sigma(P)$ in the proposition contains distinct template positions within a proposed instantiation. When two independently constructed circuit expressions refer to the same realized processing or delivery event, the union used for an aggregate resource account must identify that single occurrence once. Treating it as two newly

instantiated positions would model a different, possibly infeasible history. A provenance-aware realization map should therefore send template positions to immutable occurrence identifiers, and a shared identifier must never be counted as two separate produced outputs merely because it occurs within two different candidate descriptions. The grammar can record overlapping analytical interpretations while the physical ledger remains tied to actual events.

4.8 Grammar Scope and Next Semantic Obligations

The rules developed in Section 4 specify candidate structural compositions: which typed events can be placed together, how output and input ports may be linked, how mutually exclusive alternatives are described, how internal interfaces can be summarized, and how finite repetitions may be generated. They supply a means of describing classes of prospective production relations without enumerating every repeated graph walk. The food-and-soil derivation shows how a shared energy source, a multi-output processing event, and a delayed regenerative route can be expressed without identifying an operator cycle with an automatically self-renewing economy.

Several issues remain independent of this syntactic construction. A well-typed proposal may violate total stocks, run out of capacity during an overlapping repair activity, or fail its delivery deadline. It may lack a valid authorization, depend on a contested permission, or misstate the terms for discharging a resource claim. A technically feasible composition may fail to restore consumed ecological or material capacities; one that does restore them may still have exploitative or exclusionary distributional arrangements. The grammar does not supply a cardinal value measure or a universal criterion of fairness and does not determine whether AI systems possess legal personality or welfare interests.

The resulting division of labor across the paper is intentional. Section 3 establishes the regular-language baseline for a fixed graph projection. The present section supplies typed constructors for more elaborate candidate structures. Subsequent sections must define execution and recurrent-state semantics, examine which classes of generated representations can be recognized efficiently, and study ambiguity, overlapping contributions, and normatively significant decisions. More capable AI, operating with shared and sufficiently reliable production information, might use such descriptions to search for resource-compatible alternatives and identify bottlenecks. Its success would need to be assessed against those independent material and institutional conditions. Generating a productive-circuit expression, however compact or recursively extensible, does not itself establish sustained value return, legitimate resource use, or monetary dissolution.

5 Temporal and Material Semantics of Generated Circuits

The generative rules of Section 4 construct candidate event networks with typed input and output interfaces. The present section specifies the additional conditions under which a finite unfolding of such a network admits an executable productive history. Its analytical task is to relate dated deliveries, stock availability, shared facilities, location, regenerative states, and permissions without identifying grammatical well-formedness with material feasibility. The method is a discrete-event interpretation with explicitly delimited time horizons, followed by a worked food-production schedule, counterexamples, and a finite accounting proposition. This construction draws on the distinction between enabled and executed transitions in Petri-net analysis, the time ordering emphasized by temporal-network research, and the joint precedence–capacity problem studied in resource-constrained scheduling [15, 7, 1]. It remains a hypothetical semantics for AI-operated production, with technological coefficients and institutional conditions stipulated rather than empirically estimated.

5.1 Candidate Expressions and Dated Execution Histories

Fix a finite candidate expression P generated under the typing environment Γ in Equation 8. If P contains a guarded recurrence, first choose a finite unfolding depth and a finite horizon H . The resulting

expression has a finite set $\Sigma(P)$ of distinct template positions, a set of typed candidate wires, and declared external inputs and outputs. These positions are intentions or descriptions, while the execution record concerns actual or proposed *occurrences* identified by the tuple in Equation 1. A realization assignment associates each scheduled position with a fresh occurrence identifier, an operator, a facility, a site, and a start and completion time. Candidate positions left unexecuted remain visible as proposals; their outputs are never credited to an actual inventory.

We use a finite, ordered collection of *microsteps* $k = 0, \dots, K - 1$ within the horizon. A microstep resolves simultaneous reporting into a declared event order, or into an atomic concurrent batch whose combined demands are evaluated together. This prevents an ambiguous convention in which one event is allowed to consume the output of another at the same nominal time without an intervening completion and delivery. At time resolution Δ , the scheduler may choose a finer order where needed, but the meaning of a production claim must not depend on silently reordering causally dependent activities. Production events with duration occupy intervals $[s_e, f_e)$, with $s_e < f_e$ whenever their process requires positive time. Their consumable inputs are reserved or debited no later than their start, and their produced outputs are released only upon successful completion and any required transfer.

A *dated execution history* is thus a finite record of instantiated event occurrences, recognized deliveries, capacity reservations, inventory transitions, facility-condition updates, and operative permission versions. It is evaluated against a stated initial configuration. The history should distinguish production that has happened from production merely scheduled, since a planned output cannot settle a physical input requirement. One event occurrence may appear in several overlapping candidate circuit descriptions. Its realized inputs, outputs, and dated consequences are recorded once in the consolidated history, preserving the provenance principle of Section 2.

5.2 Prefix-Safe Inventories and Productive Causality

For each resource type r at site ℓ , define an inventory account $a = (r, \ell)$ with a coherent unit. Let $x_a^k \geq 0$ denote usable stock just before microstep k ; d_a^k the stock consumed, withdrawn, or loaded into an outgoing delivery; y_a^k the stock physically produced or received after completed events and dated arrivals in that step; v_a^k a separately identified external injection; and λ_a^k the stock lost by expiry, damage, or another declared physical process. The illustrative accounting law is stated in Equation 12.

$$x_a^{k+1} = x_a^k - d_a^k + y_a^k + v_a^k - \lambda_a^k, \quad a = (r, \ell), \quad k = 0, \dots, K - 1. \quad (12)$$

The quantities on the right side refer to uniquely identified event or delivery records. The physical account can include an executed but unauthorized transfer; authorization is assessed separately in the institutional ledger. An outgoing transfer is a debit from the origin account and a later, separately recorded credit at the destination. For the simple account, $d_a^k, y_a^k, v_a^k, \lambda_a^k$ are nonnegative, with $d_a^k \leq x_a^k$ and $\lambda_a^k \leq x_a^k - d_a^k + y_a^k + v_a^k$. Losses are debited only from available stock; output credited at the end of step k cannot finance consumption at the beginning of that same step. Dated external injections and completed output credits remain separately identified, and a transfer may not be credited both as an injection and as a produced receipt. Exogenous injections must have declared provenance and availability times. Equation 12 is a bookkeeping relation for the stipulated stock types, not a universal mass-balance law across food, electricity, machine service, and ecological indicators. More detailed conservation models would explicitly track discarded material, physical conversion losses, and energy transformation.

The *prefix condition* is stronger than an end-of-horizon balance. A projected processing event that consumes eight kilograms of grain before the harvest has been completed is infeasible if initial usable grain is zero, even if the later harvest produces exactly eight kilograms. Aggregate net availability at the final horizon may be nonnegative while the intermediate history attempts to borrow physical grain from the future. An appropriate production schedule must establish that each required input is available at the event's consumption time and that all prior demands draw from the same finite stock account. This distinction matters in recursive circuits because repeating a path in a static operator graph says little about the dated availability of its recurrent inputs.

Typed interfaces also impose quality and divisibility restrictions. Two four-kilogram batches may be aggregated for processing only if their specifications, sites, and delivery windows permit combination. A battery module cannot substitute for an arbitrary number of kilowatt-hours unless a stated conversion and discharge process supplies those services within the relevant interval. A machine-hour is an occupied capacity interval rather than an ordinary consumable item kept in a warehouse. These observations motivate the distinct shared-capacity constraints developed next.

5.3 Concurrency and Shared Productive Capacity

Let $\mathcal{F}$ denote the relevant productive facilities. For facility f , let $C_f(u)$ be its usable service capacity at time u , in units specified by its technological description. Let $\kappa_{ef}(u) \geq 0$ be the service demand of executing occurrence e from facility f at that time, with zero demand outside the occurrence's interval. The instantaneous joint-capacity requirement is Equation 13.

$$\sum_{e \in \mathcal{E}_\chi: s_e \leq u < f_e} \kappa_{ef}(u) \leq C_f(u), \quad f \in \mathcal{F}, \quad 0 \leq u \leq H. \quad (13)$$

Here $\mathcal{E}_\chi$ is the set of event occurrences selected in history χ . A facility's service rate, a stored energy quantity, and a consumable maintenance component belong in different accounts. Equation 13 constrains concurrent service; Equation 12 constrains dated inventory. For electricity, a realistic case might additionally require power-rate limits, network losses, battery states, and ramp constraints. The simplified food example below uses only an assumed six-kilowatt-hour available energy budget for one batch; it should not be mistaken for a calibrated electrical-grid model.

Parallel placement by the grammar's $\otimes$ constructor means that candidate modules can be described jointly. Concurrent execution is permitted only after a scheduler accounts for their combined demands, possible exclusive-use restrictions, site conflicts, setup periods, and maintenance downtime. If agriculture, food processing, and logistics use an aggregate six kilowatt-hours during a batch, and a parallel repair process demands two additional kilowatt-hours from the same fixed six-kilowatt-hour budget, the joint execution fails that budget. Each circuit might be executable separately, yet the combined history is infeasible. Placing the repair at another available interval, acquiring additional energy, or adopting a verified lower-energy process may resolve the conflict. Those changes each constitute a different dated execution proposal and should be recorded as such.

Production circuits can also compete for non-consumable assets. A single repair machine might be technically capable of serving multiple processes but unable to serve them simultaneously. Modeling its existence as a duplicable output would wrongly allow both circuits unrestricted access. The proper guard concerns the machine's time-indexed service capacity and any priority or reservation rules. Resources sometimes become partially renewable as use in one interval constrains service in later intervals. Capturing these effects may require state-dependent capacity and maintenance semantics. These are central concerns of resource-constrained scheduling research, although the AI resource governance interpretation in this paper remains a theoretical extension [1].

5.4 Deliveries, Spatial Transfer, and Temporal Windows

A wire from an output port to an input port proposes a directed dependency. To execute it, the sending event must complete, the relevant quantity must be made available, and the receiving event must obtain the quantity at an authorized place before consumption. Let f_e be the completion time of event e , t_{ed}^{dep} the departure time for the output batch destined for input event d , t_{ed}^{arr} its actual arrival time, and s_d the start time at which d needs the batch. A time-respecting delivery has the ordering shown in Equation 14.

$$f_e \leq t_{ed}^{\text{dep}} \leq t_{ed}^{\text{arr}} \leq s_d. \quad (14)$$

If production and consumption occur at the same physical site, the transfer may have negligible duration, while custody, quality inspection, or preparation can still create a meaningful interval. Transport

to another site requires a feasible transport event, facility service, and sufficient energy or fuel. For perishable food, storage and travel time must also preserve quality until receipt; a nominal delivery after expiry cannot satisfy the original input requirement. Equation 14 is therefore necessary for the specified dependency, with quality, capacity, and authorization supplying further independent guards.

A deadline creates an additional distinction between late completion and successful fulfillment. A candidate network may be able to produce all required goods eventually yet fail a time-sensitive service objective. The scheduler must consequently declare delivery windows, permissible delays, penalties or corrective actions, and the bearer of the corresponding rights. It cannot infer from the existence of a replacement path that every contractual or public-service duty has been met. The order and duration of contacts within a network are the central reasons that static paths and temporal paths require different analyses [7].

5.5 Permissions, Commitments, and Realization States

Material execution also depends on the powers governing resource use and facilities. The production grammar already distinguishes an operating AI from the holder of withdrawal, transfer, management, and exclusion rights. To interpret a candidate history, each executed event must be checked against the relevant operative authorization for its site, time, resource kind, amount, and delegated role. A permission valid yesterday may expire before a later renewal event. Two simultaneously competing reservations may be individually well-formed but incompatible with a governing priority rule. Shared production data can reveal such conflicts without automatically deciding whose interpretation of the governing norm is justified.

Commitments require their own state transitions. Let a recorded obligation identify the claimant, obligated party, object, performance window, conditions, and recognized discharge procedure. A delivered food batch may satisfy one obligation while creating another obligation to restore a facility or compensate a contributor. A *claim-state ledger* should consequently distinguish proposed, active, discharged, disputed, and overdue claims, with dated evidence for each transition. Material delivery alone may be insufficient to establish legitimate discharge if it fails the designated object, claimant, timing, or procedure. Conversely, a legally valid multilateral setoff may clear claims without causing a physical batch to move. These are different layers of the same proposed economic history.

In a bounded thought experiment, one could define an institutional admissibility predicate over the chosen permission versions and agreed claim-discharge rules. Its truth, however, is conditional on those rules being specified and authorized by some institution. The predicate is not an algorithmic proof of distributive justice. Normative and aesthetic disagreement, contestable governance authority, and the difference between legitimate refusal and opportunistic obstruction are retained as separate subjects for later analysis. The formalism should also allow an unresolved claim to remain attached to an event when the material scheduling problem has been solved.

5.6 Interperiod Renewal and Regenerative Continuity

A complete cycle in a graph can return to an operator while leaving its productive conditions depleted. To model recurrence, let $z_j(t)$ denote the state of a regenerative or capacity-bearing condition j at the opening of production period t : examples include field quality, effective battery health, repair-machine condition, or usable equipment capacity. Their units and interpretations must remain specific to each condition. Let $w_j(\chi_t)$ be the modeled wear or depletion due to the executed activities during period t , and let $g_j(\chi_{\leq t}, z_j(t), \varepsilon_t)$ denote the renewal effect from completed, appropriately delayed maintenance or ecological processes, possibly affected by external conditions ε_t . The provisional state update is Equation 15.

$$z_j(t+1) = z_j(t) - w_j(\chi_t) + g_j(\chi_{\leq t}, z_j(t), \varepsilon_t). \quad (15)$$

Equation 15 is a separate state-accounting relation. It does not equate a kilogram of processing residue with a kilogram of soil, or a repaired component with a new component of unlimited lifespan. The function g_j must reflect the lag, conversion conditions, quality, limits of repeated renewal, and any external

Table 6: Constructed time-respecting food-production history, conditional on specified stocks, permissions, capacities, and renewal technology.

Time	Completed activity	Stock or service consequence	Remaining feasibility condition
u_0	Energy generation D	6 kWh available to allocator	Generation capacity and authorized supply
u_1	Allocation S	Portions 2, 3, and 1 kWh, each uniquely identified	Split-account integrity; custody
u_2	Cultivation T_A	Consume 2 kWh; release 8 kg grain	Field state; cultivation interval and inputs
u_3	Processing B	Consume 8 kg grain and 3 kWh; release 7 kg food plus 1 kg residue	Processing capacity; valid grain receipt
u_4	Logistics C	Consume 1 kWh for delivery of the identified food batch	Route, delivery window, recipient authorization
u_5	Soil renewal M	Consume residue with additional inputs; prepare amendment	Conversion lag, quality and renewal inputs
u_6	Later period opens	Conditional renewal benefit becomes effective	Validated state update and renewed permissions

inputs required by the chosen technology. A circuit supports *bounded-horizon regenerative continuity* when a dated sequence of admissible executions keeps the required condition indicators within their stipulated operating ranges through the horizon and supplies the initial conditions for the next declared period. Permanent self-sufficiency or indefinite recurrence requires additional assumptions and proof beyond the bounded finite traces considered here.

This framework separates three possible outcomes. A sequence may be materially feasible within one period and fail to reproduce the inputs needed for the next. A sequence may renew its modeled resource capacities through external replacement of batteries or tools and hence remain dependent on external production networks. A sequence may satisfy the material renewal test while the distribution of costs or the validity of property claims remains disputed. A syntactic recurrence generated by rec_κ consequently provides a family of candidate unfoldings; each admitted finite unfolding requires fresh temporal, stock, permission, and regeneration checks. These checks also prevent the repeated grammatical occurrence of an already consumed batch from manufacturing fictitious new input resources.

5.7 A Dated Food-Circuit Derivation and Three Feasibility Failures

The six-template construction of Equation 10 can be instantiated as a particular sequence of completed activities. Fix an initially authorized agricultural site with sufficient specified field conditions and an energy supplier with a one-batch budget of six kilowatt-hours. Assume that initial food and grain stocks are zero, that output qualities meet the stipulated processing and delivery standards, and that the soil-renewal operation has its own additional conditions. Table 6 records one *candidate* chronological schedule. The labels $u_0, \dots, u_6$ are ordered event times, not empirical durations. The example supposes sufficient transport time between the relevant sites and no unmodeled losses; relaxing these assumptions requires new guards.

The material-transfer layer of this history is summarized schematically in Figure 4. Arrows denote dependencies whose timing and stock conditions require independent checks. The delayed amendment connection is visually separated from food delivery: the latter is a terminal service output, while the former is a hypothesized input to the next agricultural period. The recurrent relationship arises across two dated states and does not imply that the food itself flows back to the field.

Under the stipulated chronology, grain becomes available at u_2 before processing at u_3 , and all three electricity portions are allocated exactly once. The candidate history therefore avoids certain immediate stock inconsistencies. Yet three nearby variants show why grammatical validity remains weaker than ex-

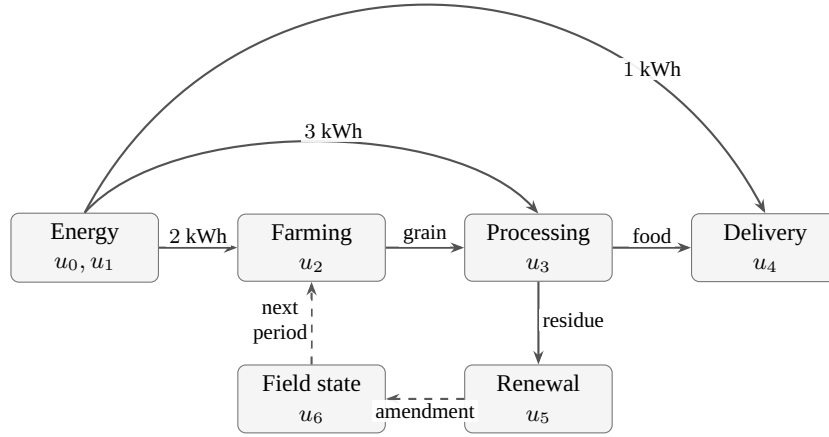


Figure 4: A dated productive circuit with distinct energy allocations, terminal food delivery, and a conditional delayed regenerative path. This is a dependency schematic, not a physical proof of renewal.

ecutable feasibility. In a *causal-order failure*, processing begins before the grain batch is harvested while initial grain stock remains zero; total grain available by the horizon can still be eight kilograms, but the prefix condition fails. In a *joint-capacity failure*, a maintenance event requiring two extra kilowatt-hours is scheduled inside the same six-kilowatt-hour production budget; the aggregate demand reaches eight kilowatt-hours despite each module appearing plausible separately. In an *institutional or regenerative failure*, the field-use permission expires before cultivation, or the residue transformation requires an additional input absent from the next period. Neither a well-typed material wire nor a closed operator path repairs those missing conditions.

The example also distinguishes output accounting from valuation. A completed food batch has a concrete occurrence identifier and a quantity. Its significance for its beneficiaries, the justice of the allocation, and the contribution of inherited agricultural knowledge are distinct matters. Dated execution records allow these questions to be traced; they do not supply a unique cardinal value attribution to every cycle in which the batch is mentioned.

5.8 A Finite-History Feasibility Certificate

A useful output of the temporal semantics is a verifiable certificate attached to each proposed finite execution history. The certificate should state the resource-type catalogue and measurement units, initial inventories, ordered event and delivery records, production and transport coefficients, shared-capacity reservations, permission versions, and relevant regenerative conditions. Table 7 lists the minimum kinds of claims that such a certificate can support. Its purpose is to make feasibility checks explicit and auditable, including when AI systems perform the scheduling and reconstruction.

The following proposition records the finite, explicitly modeled part of the verification task. Its assumptions are deliberately stronger than mere grammatical well-formedness.

Proposition 5.1. *Consider a finite ordered history with nonnegative initial stocks, uniquely identified and nonduplicated consumable batches, and the update law of Equation 12. At each microstep and for every account, suppose $0 \leq d_a^k \leq x_a^k$, $y_a^k, v_a^k \geq 0$, and $0 \leq \lambda_a^k \leq x_a^k - d_a^k + y_a^k + v_a^k$; credits are recorded only for completed events, validly dated arrivals, or separately identified external injections. Then every stock in the recorded history remains nonnegative. If the same history also satisfies Equation 13 at all relevant times, its modeled shared-facility demands remain within their declared capacities throughout the horizon.*

Proof. Nonnegativity holds at the initial step. Assume $x_a^k \geq 0$. The debit bound gives $x_a^k - d_a^k \geq 0$. Adding the stipulated nonnegative credits and subtracting at most their sum with the post-debit stock

Table 7: Verification layers for a finite interpreted circuit history and the conclusions each layer can support.

Layer	Evidence and guard	Scope of supported conclusion
Typed structure	Event identities, units and compatible port matches	Candidate composition is well formed
Dated stock	Ordered debits, credits, expiry and external inputs	No specified resource is used before availability
Shared capacity	Facility occupancy and concurrent demands	Selected schedule respects modeled capacity
Delivery	Transfer identities, sites, lead times, acceptance	Required dated goods or service reached destination
Regeneration	Wear, maintenance, lagged renewal, future thresholds	Stated capacities remain usable through the horizon
Institutional status	Permission versions, claim states and dispute evidence	Events satisfy specified rules, subject to unresolved disputes
Normative appraisal	Justifications, contested standards, review channels	Additional argument about legitimacy, beyond technical feasibility

gives $x_a^{k+1} \geq 0$ by Equation 12. Induction establishes nonnegativity at every microstep. The separate capacity conclusion follows from checking Equation 13 for each modeled facility and relevant time. $\square$

Proposition 5.1 certifies two stated properties of an already supplied finite history: nonnegative account balances and observance of declared instantaneous facility bounds. Delivery causality, completed-event evidence, and permission status are assumptions or independent checks, not additional results proved by the induction. The proposition does not prove the existence of a feasible schedule for an arbitrary candidate grammar expression, the truth of empirical conversion coefficients, indefinite renewal, lawful fulfillment of disputed claims, or normative legitimacy. Resource scheduling may require searching among alternatives, shifting events across time, or rejecting entire proposed subcircuits. Moreover, the input records may be inaccurate, partial, strategically manipulated, or affected by unforeseen physical events. A certificate is consequently relative to a specified history, data version, and assumption set.

5.9 Adaptive Recognition and the Boundary of Semantic Inference

In the optimistic thought experiment underlying this paper, capable AI operators share extensive production records and can reconstruct evolving dependency structures with much greater accuracy than a human observer could achieve unaided. The grammar offers reusable candidate representations; the temporal semantics offers tests that can rule out schedules with unavailable inputs, shared-capacity collisions, expired permissions, or unrenewed productive states. A practical system could maintain these checks incrementally when new deliveries arrive, a facility fails, or a planned schedule changes. It could also attach confidence intervals or uncertainty sets to estimated durations and outputs, then seek alternatives that remain feasible under stated disturbances.

Such improvements would be important even if the underlying grammar continues to generate infinitely many formal walks. AI systems need not enumerate every repeated walk to examine a bounded collection of tasks and their relevant causal prerequisites. They can restrict attention to a specified horizon, identify bottlenecks, retain shared event identities, and refine the local histories that affect a decision. Whether this strategy scales to an actual very large AI economy depends on graph structure, information quality, uncertainty, and the computational requirements of the chosen semantic checks. Formal recognition and optimization results from conventional scheduling settings cannot establish generalized economic tractability without further analysis.

Several boundaries remain substantive. Good prediction cannot create a missing battery or renewable input. A model that preserves measured stocks can still omit ecological degradation, unrecognized human contributions, and opaque institutional control. Advanced cognition can identify the consequences of

alternative priority rules more precisely, yet selection among contested distributive standards involves an additional normative process. Even where a set of typed, dated circuits can sustain nonmonetary resource allocation, the continued need for general accounting, transferable claims, external inputs, or settlement facilities remains an independent monetary question. The present section therefore supplies executable-history semantics for the generative grammar. Later sections will consider the expression power of alternative languages, ambiguous overlapping contributions, nonlinear fragility, and legitimate participation. Those inquiries require distinctions that a resource-feasibility certificate cannot settle by itself.

6 Expressiveness of Circuit Languages and Graph Transformations

This section examines the descriptive and computational capacities of the proposed productive-circuit grammar. The fixed-graph result of Section 3 supplies a regular-language benchmark, while Sections 4 and 5 introduce typed event networks and feasibility tests involving time, resource quantities, and permissions. The task is to determine precisely which additional features change the *object being generated*, which change the *language recognized*, and which merely require a separate interpretation of the same syntax. The discussion combines formal-language examples, a limited typed graph-rewriting construction, and a counterfactual alteration of an AI-operated food circuit. It distinguishes proved representation results from proposed economic applications and does not claim a universal complexity class for AI production relations.

6.1 Objects of Generation and Levels of Expressiveness

The expression “language of circuits” can refer to several mathematically different objects. An *edge-word language* encodes the sequence of traversed interfaces of a fixed graph; a *term language* records applications of the constructors in Equation 9; a *graph language* contains the finite typed event networks yielded by such constructions; and a *trace language* encodes event occurrences in a particular ordering or partial-order representation. A fifth object, the family of *feasible histories*, is selected by additional material, temporal and institutional conditions. The fact that these descriptions may portray related production phenomena does not make them interchangeable formal languages.

For clarity, let $\mathfrak{P}$ denote the set of well-typed finite circuit terms, with grammar and typing context Γ as in Equation 8. Let $\text{graph}(P)$ be the typed event-and-port network assembled from term P , and let $\text{trace}(H)$ be an explicitly selected serialization of a finite realized history H . In the proposed account, grammar derivation yields a candidate term $P \in \mathfrak{P}$; its graph interpretation preserves incidence and exposed ports; and the semantic tests of Section 5 may admit or reject a dated instantiation. Serializing a partially ordered history requires a stated convention, since concurrent events can have several valid linear extensions.

Table 8 distinguishes the representative descriptions. The table does not assign a Chomsky class to a collection of graphs, because the usual hierarchy classifies *word languages* over an alphabet. A claim about graph-language expressiveness requires a declared graph grammar or an explicit encoding into words, with the effects of that encoding considered separately [8, 5].

One source of terminological confusion is the use of *recursive* for three different processes. Repetition of a path on a fixed graph creates longer words in a regular language. Recursive construction of a term may create networks with new event positions or nested modules. Historical recurrence can create new actual events, stocks, commitments, and institutional states. Their expressive and economic consequences should be evaluated at their respective levels. Unlimited syntactic repetition alone proves neither context-sensitivity nor physical continuity.

Table 8: Different representations of AI productive circuits and the corresponding expressiveness questions.

Representation	Generated or recognized object	Boundary of the formal claim
Fixed-graph walk words	Finite strings of traversed edge labels	Regular for a finite graph with finite edge alphabet
Typed circuit terms	Finite expressions using typed productive constructors	Classification depends on the selected string encoding and parameter grammar
Typed graph structures	Incidence networks of events, ports, and interfaces	Requires graph-grammar notions; word hierarchy applies only after encoding
Dated execution traces	Ordered records of unique productive events	May require stock, time, resource and permission guards
Governance histories	Changes in valid permissions, control interfaces and claims	Structural events do not establish normative legitimacy

6.2 Finite-State Recognition and Unbounded Resource Guards

The automaton construction in Proposition 3.1 recognizes exactly those finite edge sequences that return to a designated vertex on a fixed finite graph. Its finite state summarizes current graph location; it does not retain an arbitrary integer stock, an unbounded ledger of open obligations, or a history of each resource batch. A semantic filter may therefore change the complexity of the *admitted traces* even when the projected operator graph remains fixed. To specify this possibility without making broad complexity claims, consider a separate idealized recording protocol.

Suppose the alphabet $\{a, b\}$ represents two kinds of event in an artificially serialized AI production history: a denotes the registration of one unit of a particular resource obligation, and b denotes its corresponding discharge. Under a stipulated archival convention, registrations are recorded before discharges, and the accounting rule requires the same number of both. If arbitrarily many unit obligations may be registered, the encoded admissible histories form the language in Equation 16.

$$\mathcal{L}_{\text{match}} = \{a^n b^n \mid n \in \mathbb{N}_0\}. \quad (16)$$

Here the two symbols identify event *kinds*, while individual occurrence identities and claimant information must be maintained in the separate ledger. This toy language verifies equality of counts, not individual creditor identity or lawful claim discharge. Moreover, it is not asserted that real production transactions normally follow a register-then-discharge ordering. That ordering is a chosen encoding to expose a specific recognition problem.

Proposition 6.1. *Under the unbounded-unit and block-order assumptions of Equation 16, the language $\mathcal{L}_{\text{match}}$ is not regular. For a fixed upper bound on the number of obligations, the corresponding finite sublanguage is regular.*

Proof. For distinct nonnegative integers i and j , the prefixes a^i and a^j are distinguished by the suffix b^i : the first concatenation belongs to $\mathcal{L}_{\text{match}}$ while the second does not. Thus infinitely many prefixes require pairwise distinct residual languages, contradicting the finite-state condition for regularity. When $n \leq N$ for a fixed integer N , the language contains only finitely many words and is therefore regular. $\square$

The result in Proposition 6.1 is a standard formal-language fact, stated here as an explicit resource-ledger illustration [8]. The same construction can be recognized by an appropriate counter or push-down mechanism with the stated ordering. If a three-stage archival convention requires equal unbounded counts of registrations, deliveries, and confirmations, its encoded words would have the form $a^n b^n c^n$, a classical non-context-free language. This is a *possible encoding under strong global matching requirements*; it is not an inference that all AI production histories need context-sensitive grammars. In particular, bounded stocks, bounded horizons, finite attribute domains, and a finite set of permission states may permit finite-state encodings, even where the convenient high-level specification uses arithmetic guards.

This example also shows why it is inadequate to infer formal-language class from the number of circuits. A finite graph can generate infinitely many repeated return words and still admit finite-state recognition; a fixed graph equipped with unbounded matching obligations can generate an admitted trace sublanguage that is nonregular. The change is caused by the declared *semantic matching requirement* and its encoding. Neither the apparent density of the production network nor its use of monetary or nonmonetary coordination independently determines the Chomsky class.

6.3 Typed Graph Rewriting and Evolving Production Interfaces

The productive grammar of Section 4 composes candidate event modules and guards their recurrence. Real AI-operated production systems may also change their repertoire of feasible connections. An energy producer may be added, a storage interface may be retired, a facility may be divided into separately schedulable units, or an operator’s authority over a transfer interface may expire. Such changes require representing the network itself as an evolving object, rather than merely traversing edges that remain fixed.

Let $\mathcal{G}_t$ denote a finite typed *structural* graph of productive interfaces at institutional time t . Its vertices and edges may represent declared roles, facilities, ports, and authorized or merely proposed connections, as specified by its type graph. Its event occurrences and physical resource accounts remain in a linked historical ledger; a graph rewrite is not treated as material production in its own right. For a rewrite schema γ , let L_γ be a pattern to match, K_γ the part retained, and R_γ the replacement description. A proposed typed rewrite is represented by Equation 17.

$$\gamma : L_\gamma \longleftarrow K_\gamma \longrightarrow R_\gamma, \quad \mathcal{G}_t \xrightarrow[\chi]{\gamma, m} \mathcal{G}_{t+1}. \quad (17)$$

In this schematic notation, m is a type-respecting match into $\mathcal{G}_t$ and χ represents application conditions, including declared structural consistency and relevant authorization checks. The span notation is motivated by algebraic graph transformation, which studies rule-based replacement, gluing, and constraints on deleting nodes with incident edges [5]. We do not claim to have constructed the complete category-theoretic double-pushout semantics for the heterogeneous, permission-bearing economic network proposed here. In particular, rules affecting a material process and those affecting its legal entitlement have distinct validation requirements.

At minimum, an admissible structural rewrite must prevent links from being left attached to a removed node, preserve compatible types for surviving connections, and record the provenance of the rule application. It must also handle already-realized events conservatively: removing a prospective interface from the current graph does not erase an earlier delivery, an unresolved commitment, or the evidential basis of a dispute. Decommissioning a facility may therefore close a future production route while its historical event identifiers remain available for audits and lawful claim resolution. The identity discipline developed in Section 2 survives a change of structural topology.

6.4 A Supplier-Reconfiguration Example

Consider the food network of Figure 4, in which agriculture, food processing, and logistics depend on electrical energy. Suppose an established supplier D is able to provide six kilowatt-hours in the relevant period, allocated as two to farming, three to processing, and one to logistics. The material schedule in Section 5 is feasible only if delivery times, capacities, and permissions satisfy their stated guards. Now suppose an additional supplier E becomes available and can provide a separately authorized three-kilowatt-hour portion to processing. The candidate structural graph can be rewritten to include the new supplier and a processing input interface, while the original supplier’s processing link is retired or left unused according to the chosen replacement rule.

Figure 5 compares the relevant local dependencies. The diagram is an *interface rewrite*: it does not depict a simultaneous increase in the actual energy used by processing. In the proposed adapted plan, D supplies two kilowatt-hours for farming and one for logistics, while E supplies three for processing.

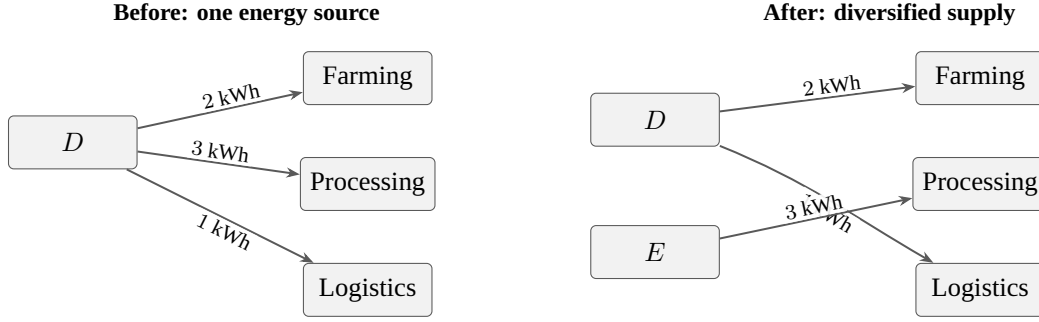


Figure 5: A local typed energy-interface rewrite, illustrating how a candidate production graph can replace an energy dependency while preserving the functional demands of farming, processing, and logistics. The edges denote proposed dated allocations and require separate stock, delivery, and authorization tests.

Relative to the original six-kilowatt-hour generation ceiling at D , three kilowatt-hours of its possible output are not used by this particular candidate history. The example makes no allocation of that residual availability to maintenance, storage or other operators; any such use would require a newly identified event and valid commitments. Each used unit must be delivered, counted once, and admitted by the corresponding permission and capacity records.

The new route may reduce dependence on a single energy supplier, but the grammatical change alone does not establish greater resilience. If E has unreliable capacity, if the new line arrives after processing begins, or if the permission is disputed, the adapted schedule can fail. If both suppliers depend on the same upstream battery system, nominal graph diversification may leave a common-mode failure intact. Conversely, a verified replacement route can enable the same food output with improved flexibility and can change which productive circuits appear in subsequent histories. Those consequences depend on the quantitative and institutional state, not merely on the existence of an additional edge.

This example also illustrates the limits of edge-word equivalence. A projection retaining only the sequence “farming, processing, logistics” can describe the same sequence before and after the rewrite. It omits the identity of the energy supplier, the location of a possible bottleneck, and the holder of a permission to disconnect or redirect supply. Graph rewriting is therefore meaningful for the evolving *structure of production relations*, even where a simple operator trace looks unchanged.

6.5 Provenance and Invariants under Structural Transformation

A formal framework for AI-assisted circuit analysis should specify what survives a change of graph representation. Type preservation, historical event identity, and accountable resource records serve different purposes. A graph rewrite may preserve structural types while changing feasible capacities. It can maintain all historical event records while making future execution impossible. It can also produce a materially feasible connection whose authorization has not been established. Thus invariance claims require a precisely identified object, and a proven structural property must not be presented as a theorem of generative justice.

For the proposed finite rewrite histories, a modest identity-preservation result is available under explicit recordkeeping assumptions.

Proposition 6.2. *Let a finite sequence of well-typed structural rewrites act on graphs $\mathcal{G}_0, \dots, \mathcal{G}_T$. Suppose (i) the initial historical ledger assigns distinct identifiers to distinct already realized event occurrences and records each such occurrence, (ii) each rewrite preserves all earlier event records, (iii) the new occurrences registered at any one rewrite step receive identifiers pairwise distinct and disjoint from the identifiers already registered, and (iv) no rewrite alters the quantity, type, timestamp, or identifier of a previously certified event record. Then every realized event recorded through time $t \leq T$ remains uniquely traceable in that ledger after all later rewrites. This is a record-identity invariant under the*

stipulated registration rules; it imposes no condition of material feasibility, permission legitimacy, observation completeness, or indefinite regeneration.

Proof. At time zero, the stipulated initial ledger is injective on recorded occurrences. Assume that through time t each recorded occurrence has a distinct identifier and retained certified fields. At the next rewrite, historical identifiers and fields are preserved, and the new identifiers are mutually distinct as well as disjoint from the older identifier set. Hence the enlarged ledger remains injective on recorded occurrences, while preserving the earlier records. Induction over the finite sequence proves the claim. $\square$

Proposition 6.2 gives a bookkeeping invariant by construction. The injective initial registry and mutually fresh new identifiers are substantive hypotheses: preservation of an initially duplicated identifier would preserve ambiguity, and two simultaneous events assigned the same purportedly fresh identifier would defeat unique traceability. It does not establish that every description of the same event has a unique value contribution: that requires an attribution rule. Nor does it validate the purported consent or legal authority underlying an event record. Append-only preservation can also raise genuine questions of confidentiality, contestation, and correction; a legitimate governance arrangement might retain the original event and attach an auditable correction or access restriction rather than treating an immutable technical record as normatively final. These qualifications are especially relevant when an AI network depends on a common data infrastructure to coordinate complex value circuits.

6.6 Recognition, Reachability, and Bounded Verification

The richer representation permits at least three separate computational inquiries. *Membership* asks whether a word or finite typed term belongs to the language described by given rules. *Reachability* asks whether a sequence of permitted structural rewrites and event executions can arrive at a specified future configuration. *Scheduling* asks whether a candidate production plan can satisfy temporal, stock, capacity, and permission conditions. These questions can have substantially different costs, and a fast syntax recognizer does not give a general algorithm for deciding all versions of reachability or finding an optimal production schedule.

For a fixed finite graph with a fixed edge alphabet, the language of legal closed walks is regular, as already proven in Proposition 3.1. For the grammar of Equation 9, syntactic generation may yield arbitrarily large finite event structures. For typed rewriting, the graph itself may gain or lose nodes and edges. For semantic recognition, even a fixed structural projection may require unbounded accounting or matching states. The complexity of each problem depends on restrictions such as a bounded horizon, bounded capacities, a finite set of resources, restricted rewrite patterns, and the number of candidate event arrangements.

In the optimistic AI–AI thought experiment, shared production records and powerful relational inference may support incremental recognition of changed interfaces and rechecking of only the affected dependencies. For example, after adding energy supplier E , a planning AI could update supplier-specific capacities, invalidate schedules using an expired link, and recompute the feasibility of processing batches that depended on supplier D . Where a bounded horizon and finite event catalogue permit exhaustive local verification, the system may produce a transparent certificate showing which histories satisfy the declared constraints. On a larger network it may rely on conservative approximations, decomposed scheduling, and reversible proposals. Such practices are candidate research directions, not a proof that arbitrary dynamic productive graphs are decidable or easy to optimize.

An important asymmetry persists between computational simplification and normative accountability. A compact automaton can summarize many possible walks, and a typed graph can expose resource dependencies, but either representation can hide historical contribution, contractual commitments, or the rules determining which AI is permitted to withdraw a scarce input. A recognizer designed to minimize computation may therefore require an accompanying ledger of outstanding claims and contestation rights. The design of that accompanying evaluation belongs to the normative interfaces developed in Section 2 and later in this paper, rather than to the graph rewrite alone.

6.7 Scope of the Expressiveness Results

The central methodological result is a division of explanatory labor. Finite-graph return words admit regular descriptions. Typed production grammar terms describe candidate networks with branching, multiple inputs, alternative connections, and recurrence. Unbounded numerical matching can make a selected trace language nonregular, and certain stronger globally matched serializations can exceed context-free expressiveness. Graph rewriting describes historical changes in possible structural connections and retains further questions about graph types, admissible matches, and event identities. Physical and institutional semantics separately decide whether any given candidate structure can become an executable productive history.

These results provide useful limits on two opposing claims. The mere existence of infinitely many return walks does not establish the necessity of an extraordinarily powerful language or a complete enumeration algorithm. Conversely, the existence of a finite graph grammar or a tractable local schedule does not establish that an entire changing economy has a finite-state, globally observable, or straightforwardly computable semantics. Different questions legitimately require different representations, and classifications should be attached to specified languages and verification tasks.

The paper’s future work on shared contributions, trust-dependent nonparticipation, and higher-order path attenuation will accordingly treat productive-event provenance as distinct from the multiplicity of its grammatical descriptions. A route that is closed as a word may be infeasible as a resource history; a feasible history may concentrate effective withholding power; an authorized transfer may be challenged on distributive grounds. The proposed generative grammar can help identify and compare those possibilities, particularly when advanced AI systems share extensive but fallible records. Their economic significance remains conditional on material, historical, institutional, and normative inquiry.

7 Circuit Ambiguity, Shared Contributions, and Value Attribution

This section examines the relation between the multiple grammatical descriptions of a productive circuit and the economic interpretation of a recorded production history. The typed events, provenance identifiers, and compositional rules of Sections 2 and 4 permit several circuit expressions to refer to overlapping productive activity. Their possible histories are further restricted by the temporal and material semantics in Section 5. The objective here is to specify what such multiplicity implies for physical accounting, explanatory attribution, and the justification of claims upon resulting resources. A finite event-identity construction, a shared-energy food-and-maintenance example, and a conditional cooperative-game allocation illustrate the distinctions. The proposed attribution interfaces remain a framework for inquiry in the hypothetical AI–AI economy, without assuming that all forms of generated value admit a common scalar measure.

7.1 Derivation Multiplicity and Productive-Event Identity

A grammar generates descriptions according to rules, and the relation between a derivation and its described object need not be one-to-one. Serial connections can be parenthesized in more than one way when the grammar permits associative grouping. Two candidate circuits can contain the same upstream electricity event. A nested maintenance excursion may be represented either as a named submodule or through its constituent events. Graph rewriting can preserve a historical productive event while changing the permitted interface through which later operations will depend on it. These cases create *syntactic multiplicity*, *circuit-cover multiplicity*, and *historical-representation multiplicity*, respectively.

The unit of realized production adopted in this paper is an identified event occurrence. Let H be a finite, dated history and let $E(H)$ be its finite set of occurrence identifiers. Each identifier $e \in E(H)$ has a declared event type, resource inputs and outputs, time window, facility use, and provenance references. A grammatical occurrence position p in a candidate term P may be related to an executed occurrence through a realization map $\pi_P(p) \in E(H)$, when that position has genuinely been instantiated. A position

without a realized match remains a proposal. Whenever different circuit descriptions point to the same identifier e , they refer to a single occurrence; the accounting consequences of that occurrence are entered once.

The uniqueness condition applies to occurrences, rather than to event types. If an AI processes grain in two distinct periods, the two processing events may have identical type signatures while receiving different occurrence identifiers. Conversely, two identical-looking records are not automatically established as the same event merely because they share a type, date, and quantity. Their identity may require batch lineage, facility logs, transaction identifiers, or independently authenticated evidence. A reconstruction procedure that conflates these cases can undercount repeated production or double-count shared production. Advanced AI access to common records may improve resolution but does not make disputed or missing provenance disappear.

Formal ambiguity can be represented in a database-provenance setting by retaining information about the derivations that justify an output; provenance semirings provide a notable mathematical precedent for retaining alternative and combined derivations in query evaluation [6]. The present paper borrows only the methodological distinction between an output and the provenance of its derivations. Semiring coefficients and formal proof expressions are not measurements of economic value, and an alternative database derivation does not automatically correspond to another produced kilogram, an independently valid claim, or an additional moral contribution.

7.2 Circuit Covers and an Event-Unique Accounting Invariant

Consider a finite family $\mathcal{C}$ of candidate circuit descriptions referring to the realized history H . For each $C \in \mathcal{C}$, let $E_C \subseteq E(H)$ be the set of distinct realized event identifiers referenced by that circuit. Define its incidence indicator $a_{Ce} = 1$ if $e \in E_C$ and $a_{Ce} = 0$ otherwise. For a fixed resource type r , let $y_{e,r} \geq 0$ denote the gross output of that resource recorded at event e in a common physical unit. The event multiplicity and the resulting circuit-wise sum are given in Equation 18.

$$\begin{aligned} m_e(\mathcal{C}) &:= \sum_{C \in \mathcal{C}} a_{Ce}, \\ \sum_{C \in \mathcal{C}} \sum_{e \in E_C} y_{e,r} &= \sum_{e \in \bigcup_{C \in \mathcal{C}} E_C} m_e(\mathcal{C}) y_{e,r}. \end{aligned} \tag{18}$$

The identity follows by rearranging a finite sum. It exposes a source of counting error: summing gross event outputs separately across overlapping circuits weights shared occurrences by their description multiplicities. For a resource already represented by a final physical inventory, additional upstream counting problems remain, since an intermediate output can be consumed by a later event. The event-unique sum therefore addresses *duplicate occurrence accounting*; boundary accounting and input-output conversion still require the matched-port discipline of Proposition 4.1 and the stock semantics of Section 5.

Proposition 7.1. *Fix a finite realized history H , a resource type r , and two finite circuit-description families $\mathcal{C}_1$ and $\mathcal{C}_2$ that cover the same set $U \subseteq E(H)$ of distinct realized event identifiers. Under identical event-level resource measurements $y_{e,r}$, the event-unique gross-output account $\sum_{e \in U} y_{e,r}$ is invariant under replacement of $\mathcal{C}_1$ by $\mathcal{C}_2$, including the insertion of redundant descriptions of circuits already represented. Circuit-wise sums need not be invariant.*

Proof. By hypothesis, both families have union U and refer to the same underlying event-level records. Their event-unique account is consequently the same sum indexed by U . By Equation 18, the circuit-wise sum depends on the multiplicities $m_e(\mathcal{C}_j)$, which can change when a circuit description is added, removed, or subdivided. Thus the circuit-wise sum may change despite the invariant event account. $\square$

Proposition 7.1 is a bookkeeping result contingent on the quality of occurrence identity and the completeness of the chosen cover. It does not attribute productive *causation*, assign a unique economic surplus, or identify legitimate recipients. A family of circuits may omit unobserved ecological, knowledge, or external material conditions, and a correct gross account may remain incomplete as a representation

Table 9: Distinct analytical variants of the hypothetical production network. Electricity allocations apply only to the explicitly identified activities and periods; a closed soil pathway or maintenance event is not inferred where omitted.

Variant and location	Operator or template roles	Electricity allocation	Modeled regeneration scope
Introductory projection (Section 1)	A, B, C production roles; D electricity; R repair	No quantities specified	Candidate walks only; execution untested
Projected graph (Section 3)	A, B, C, L, F : separate abstract interfaces	No quantities specified	Walks only; execution untested
Food and soil (Sections 2–5)	A operates cultivation template T_A ; B processing; C logistics; D energy; M soil renewal	$2 + 3 + 1 = 6$ kWh for food operations	Soil amendment requires extra inputs, delay and proof
Supplier replacement (Section 6.4)	D and E are alternative energy sources	$D : 2 + 1 = 3$; $E : 3$ kWh for food	Same food service; renewal untested
Efficient food and repair (Sections 7–9)	D, A, B, C retained; R equipment repair; M omitted	$2 + 1 + 1 + 2 = 6$ kWh, including repair	Next-period machine eligibility only; soil renewal untested

of the whole productive system. When the graph transformation of Section 6 changes future productive relations, an appropriate comparison must preserve historical identifiers while adding fresh identifiers for subsequent events.

7.3 A Shared-Energy Food and Maintenance History

The following constructed case is a separate, technologically modified variant of the food system in Sections 2.5 and 5.7. The earlier one-period food-and-soil case used M for soil renewal and required three kilowatt-hours for processing, leaving its soil-amendment inputs and later effectiveness to separate guards. The present five-role variant retains D for electricity, A for cultivation, B for processing, and C for logistics, but introduces a distinct equipment-maintenance role R . The soil-renewal event M is outside the executed event set of this variant; its ecological effects remain unresolved rather than being silently replaced by machine repair. The variant stipulates an improved processing technology requiring one kilowatt-hour per batch, and treats the improvement and any prior investment as exogenous assumptions. In a single scheduled period, one uniquely recorded production event e_D makes six kilowatt-hours available. The food-production activities consume two kilowatt-hours for cultivation, one for an efficient processing configuration, and one for delivery. A maintenance activity consumes two kilowatt-hours and supplies an explicitly dated equipment-service output relevant to the following production period. These six kilowatt-hours may be divided into separately metered portions through an allocation event; those portions all refer to the same original energy-generation occurrence.

Table 9 specifies the scenario boundary across the paper. The operator labels in the introductory projection and the material-accounting examples follow consistent role assignments, whereas the five-interface walk graph of Section 3 is a separate abstract projection with its own expressly declared symbols. Its purpose is to prevent differences in role assignments and technological coefficients from being read as simultaneous, unrecorded transitions of one historical production system. The supplier-reconfiguration case in Section 6.4, for example, replaces the source of the existing three-kilowatt-hour processing input and does not stipulate the later one-kilowatt-hour efficiency improvement.

Table 10 records one candidate joint schedule. Food processing receives eight kilograms of grain and yields seven kilograms of delivered-food material and one kilogram of separately tracked residue, subject to the further conversion, transport, and regenerative requirements in Sections 4 and 5. The energy requirement for processing is assumed to have been lowered by a historically acquired efficiency improvement. The model therefore does not regard that improved coefficient as cost-free or universally

Table 10: One stipulated event history supporting overlapping food and maintenance circuits; electricity quantities share a single supply account.

Event and operator	Electricity use	Material or service consequence	Proposed circuit membership
$e_D (D)$	Supply 6 kWh	One dated generation-and-availability event	Food and maintenance
$e_A (A)$	2 kWh	8 kg grain under stated field conditions	Food; possible renewal relations
$e_B (B)$	1 kWh	7 kg food and 1 kg residue from grain	Food; regenerative relations
$e_C (C)$	1 kWh	Transport and delivery of 7 kg food	Food
$e_R (R)$	2 kWh	Dated equipment maintenance for later use	Maintenance

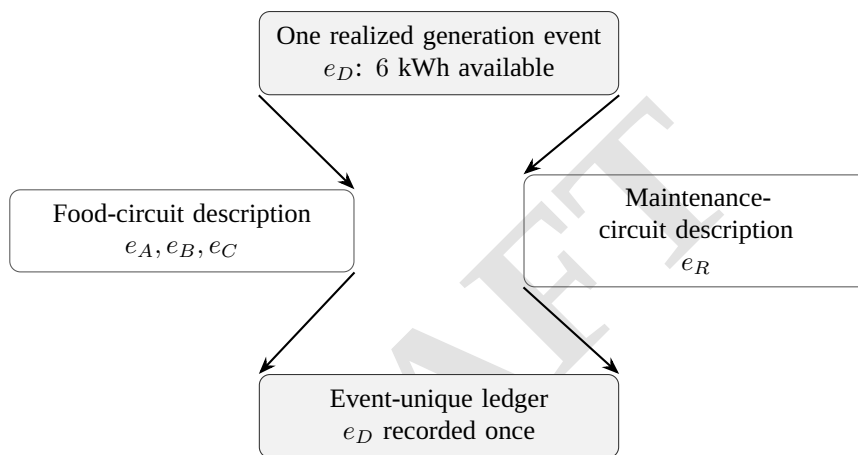


Figure 6: Two circuit descriptions reference one energy-generation occurrence. The diagram represents analytical membership and event identity, rather than the full material network or an economic-value conservation law.

available.

Two candidate circuit descriptions may reference e_D : one concerns food supply and another concerns the renewal of equipment capacity. They share a source without implying that twelve kilowatt-hours were generated. Figure 6 represents the descriptive overlap through an event register. Its two paths indicate memberships in different analytical descriptions, while the middle box denotes the single identified generation occurrence. The production and claim layers are omitted from this schematic because its purpose is to distinguish circuit membership from production multiplicity.

The joint schedule uses six kilowatt-hours and satisfies the stated energy budget. If processing retained its earlier requirement of three kilowatt-hours, simultaneous food and maintenance activities would require eight kilowatt-hours, exceeding the same supply. Treating each circuit as independently feasible would then obscure an infeasible joint history. A reconstruction of the actual event set and aggregate capacity obligations resolves this particular accounting problem. It leaves open the distribution of access rights, the legitimacy of the efficiency investment's costs, and the institutional priority given to equipment renewal.

The maintenance event can enable future food production without directly producing additional food during the present period. Its historical contribution may be considerable despite the absence of a directly commensurable food-output quantity. Similarly, ecological conditions, external replacement inputs, and common technical knowledge may remain indispensable. A unique event register prevents duplicated physical records, while a credible account of *value generation* requires these broader relations to be

examined on their own terms.

7.4 Production Records, Causal Explanation, and Distributive Claims

The word *attribution* carries several analytical meanings. A physical-resource attribution assigns measured flows to executed events and their batch or facility records. A historical or provenance attribution identifies relations upon which a particular result depended. A causal attribution assesses what change in a result can be associated with the presence, removal, or alteration of specified activities under a counterfactual model. A distributive attribution assigns some benefit, authority, obligation, or recognition to parties according to a proposed normative rule. These relations can inform one another, yet none follows automatically from another.

Suppose the food delivery depends jointly on grain, energy, processing, and logistics. The absence of any one indispensable input may prevent the delivery in the stipulated technology. That observation alone cannot determine a numerical fraction of the output that *belongs* to each contributor. Under complementarity, multiple inputs can each be counterfactually necessary for the whole. A simple leave-one-out test might then call each input indispensable, even though assigning the full output to each contributor would plainly duplicate distributive claims. Replacing a participant can also change feasible technology, costs, timing, or authorization. The definition of the counterfactual background is consequently material to causal assessment.

Multiple historically significant activities may be poorly captured by current-output measures. The research that enabled processing efficiency, the maintenance reserve, the preservation of soil quality, and the governance arrangement that secures continued access can each condition later production. Some contributions are temporal investments; others are common conditions whose marginal effects depend upon which surrounding activities are held fixed. The generative-justice literature supplies a reason to consider the control and regeneration of such value-producing conditions [4]. The present extension to heterogeneous AI operators is an interpretive proposal, and the citation does not establish the present grammar or its attribution rules.

The normative inquiry therefore begins with an explicit specification of the relevant bearers of rights and obligations. The AI acting as an operator might be a delegated system; a cooperative, public institution, or other person might hold the underlying claims. A resource record stating that AI *D* generated electricity should not silently settle the legal question of who is entitled to authorize that generation, receive a benefit, or challenge subsequent allocation. Systems of shared-resource governance have varied historically, as documented for human institutions by Ostrom [14]; corresponding arrangements for hypothetical AI networks require their own legal and normative analysis.

7.5 Conditional Allocation by Cooperative Marginal Contributions

A cooperative-game representation offers one possible method of distributing a *specified scalar surplus* among an expressly selected set of participants. Let N be a finite set of eligible participants with $n = |N|$, and let $v : 2^N \rightarrow \mathbb{R}$ assign a coalition worth in one explicitly chosen unit, with $v(\emptyset) = 0$. Given that model, the Shapley value for participant $i \in N$ is defined in Equation 19 [17].

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|! (n - |S| - 1)!}{n!} (v(S \cup \{i\}) - v(S)). \quad (19)$$

The method averages marginal coalition increments across insertion orders. Its formal properties refer to a defined cooperative game and the distribution rule chosen for that game. The present research does not assume that a single cardinal coalition-worth function captures heterogeneous ecological, social, operational, and historical values, nor that the Shapley allocation is automatically required by justice or existing law.

Consider a deliberately simplified unanimity example. Under one institutional specification, $N = \{A, B, C\}$ contains cultivation, processing, and logistics; a public electricity supply is treated as an exogenous common enabling condition. Define $v(S) = 7$ units of a stipulated allocable benefit if $S = N$

and zero otherwise. Equation 19 yields equal shares of $7/3$ units. Under a different allocation specification, $N' = \{A, B, C, D\}$ regards the energy contributor as a separate eligible claimant and assigns the same total 7 units only when all four members participate; the corresponding symmetric shares are $7/4$ units. Both games are consistent with the same *recorded* delivery of seven kilograms of food, provided the relevant external electricity arrangements are specified. Their different coalition-worth and eligibility assumptions produce different distributive numbers.

These illustrative worth units are *not* kilograms physically divided by a theorem. The game imposes a separate scalar distributive object. The change in assigned shares shows how the participant universe, the classification of public production conditions, and the coalition counterfactual influence an allocation. Other models can yield other results, including allocations that reserve resources for maintenance, future generations, or urgent needs. A claim that the participants' welfare, ecological conditions, legal entitlements, and long-run productive opportunities are commensurable would require substantial independent justification. For the present paper, cooperative marginal allocation is a contrastive analytical example, useful precisely because it displays what a model must stipulate before it produces numerical shares.

7.6 Partial Attribution, Unresolved Claims, and Value Return

A more general productive-circuit analysis may maintain several attribution records simultaneously. Physical outputs can be ledgered by unique event identity; historical dependencies by provenance links; monetary and in-kind claims by their respective instruments and discharge rules; and contested entitlements by their procedural status. Ecological capacity renewal and access to common productive knowledge can be tracked as relevant state conditions without being assigned one universal monetary price. Descriptions of contribution can remain partial when evidence, causal models, or normative agreements are incomplete.

Under such an arrangement, different notions of return need not be collapsed into a single dividend. The AI operating an agricultural facility may receive spare equipment or reliable electricity. The agricultural landscape may receive soil renewal. A human community benefiting from the output may enjoy assured food access. A shared-energy contributor may receive a maintenance commitment. Those are heterogeneous forms of return and continuation. Their distribution may be contested, and a ledger describing them does not itself establish their justice. The normative standards that evaluate their adequacy remain explicit questions, including how disagreement is heard and which authority may modify a claim.

A particularly important boundary concerns claims whose causal or monetary magnitude appears small. A right to contest a resource allocation, an unpaid obligation to a supplier, or evidence of impermissible extraction can remain normatively significant despite a small contribution to an aggregate throughput statistic. A method that compresses many circuit descriptions can be legitimate for selected material queries while still requiring an exception register for unresolved obligations and reviewable decisions. The distinction anticipates the separately planned research on normative renormalization, while keeping the present paper focused on the generative grammar's descriptive interfaces.

7.7 AI-Assisted Reconstruction of Overlapping Productive Histories

The hypothetical availability of advanced AI and widely shared production records may improve attribution research in several limited ways. First, unique event identifiers, signed delivery receipts, batch lineage, meter data, facility states, and permission versions can support the reconstruction of shared histories. Second, typed graph matching can identify candidate circuits containing common occurrences, while maintaining a canonical event register. Third, resource accounting can use the resulting union of occurrences to test joint feasibility, as in Proposition 7.1. Fourth, a query-specific explanatory model can explore counterfactual dependence and present explicit uncertainty, rather than automatically converting the number of matched paths into a distribution of surplus.

Table 11 separates these tasks and their remaining limitations. In particular, an accurate reconstruction of a finite executed history need not identify the unique value of historical knowledge, ecological

Table 11: Interfaces for AI-assisted analysis of overlapping productive circuits and their independent validation requirements.

Task	Candidate AI-assisted representation	Independent validation or normative boundary
Occurrence resolution	Event identifiers, batch lineage, audited time records	Identity errors, missing events, duplicated records
Shared-resource accounting	Typed inputs, unique outputs, common capacity ledgers	Time-prefix constraints, external inputs, process loss
Explanatory attribution	Explicit interventions and alternative histories	Causal assumptions, non-identifiability, model sensitivity
Benefit allocation	Authorized claims, declared worth model, plurality of return forms	Eligible participants, legal powers, fairness and contestation
Circuit summarization	Subgraph summaries with retained provenance links	Preservation of unresolved claims and reviewable exceptions

conditions, or institutional cooperation. Data-sharing rules, data-quality incentives, privacy restrictions, and asymmetrical control over the reconstruction tools can themselves affect who is able to contest the analysis. A distributed model may be computationally convenient while becoming politically concentrated through a common protocol administrator or verifier.

An implementable research workflow might begin with a bounded period and a stated material question. An AI reconstructs a candidate event history from timestamped data, uses a grammar to identify structurally admissible interpretations, tests the actual event union against inventories and shared capacities, and returns a report separating verified occurrences from inferred causal or normative claims. When two explanations rely on the same energy event, their provenance can overlap without producing a duplicated resource entry. When an authorization is disputed, the system can preserve the record and route the disagreement for an appropriate institutional procedure. This workflow is a proposed decision-support arrangement; it does not replace lawful adjudication or prove universal predictability of a nonlinear economy.

7.8 Scope of Circuit Attribution within the Grammar

The present section establishes a limited invariant and a necessary analytical separation. Under trustworthy event identities and a fixed executed-history account, changing the number and arrangement of grammatical descriptions leaves event-unique physical accounts unchanged. Explanatory, distributive, and regenerative questions require further models. The worked energy example demonstrates how overlapping circuit descriptions can refer to one shared resource source; the cooperative-game comparison demonstrates the sensitivity of a scalar allocation to the participant set and the definition of coalition worth. Together, these constructions delimit what the grammar can establish on its own.

The next section will examine how trust-dependent participation, strategic withholding, and nonlinear propagation can prevent candidate circuits from being realized or sustained. Those effects can change future event histories and therefore change the inputs to provenance and attribution. Later normative analysis will address contested justice standards and the legitimacy of power over resource interfaces. Neither continuation of circulation nor a proposed allocation share resolves those questions by definition. In an AI–AI production society, generative circuit grammars may make relations more legible and testable, while economic value and justified benefit return remain historically and institutionally situated objects of inquiry.

8 Nonlinear Dynamics and Fragile Circuit Participation

The generative grammar describes possible productive circuits, while a dated execution history records the portion of an admissible proposal that has actually been carried out. Participation by distinct AI

operators introduces a further distinction: a technologically available, resource-feasible and formally authorized activity may remain unrealized when an indispensable operator declines, revises a commitment, or becomes unable to provide an agreed service. This section examines how such decisions interact with material scarcity, shared bottlenecks and delayed regeneration. Its objective is to specify a limited dynamic extension of the typed grammar and to establish two conditional results about propagation of unavailable interfaces and attenuation along long sequences of distinct exposures. A two-period energy, food and maintenance example then connects the constructions to the shared-event accounting of Section 7. Network research on threshold cascades, load redistribution and interdependent failures supplies mathematical precedents for particular mechanisms [18, 12, 2]; it provides neither behavioral estimates for hypothetical AI operators nor a normative standard for their refusals.

8.1 Evolving Productive States and Executed Event Histories

Fix a finite observation horizon divided into production periods $t = 0, \dots, T - 1$. Let G_t be the typed structural interface graph valid at the opening of period t , and let P_t be a well-typed candidate circuit expression constructed in that graph. The dated history χ_t contains uniquely identified productive event occurrences executed during the period, including events that satisfy a selected finite unfolding of P_t . It may also contain relevant events outside that unfolding, since other circuits can compete for a common input or facility. The inclusion of a candidate event position in P_t does not establish that the corresponding event appears in χ_t . Conversely, the same executed event may be referenced by several candidate descriptions without being replicated in the consolidated ledger.

A period-opening state S_t consists of a typed inventory vector x_t , a vector z_t of capacity-bearing and regenerative conditions, the prevailing permissions and commitment records a_t , a collection r_t of evidential confidence or reliability assessments, and the graph G_t . The collection a_t can contain unresolved claims and contestation status as well as operative permissions; it therefore has richer structure than an indicator of unconditional approval. Operator participation policies are denoted by μ_t , and exogenous disturbances by ε_t . A proposed discrete-time interpretation is summarized in Equation 20.

$$S_{t+1} = \mathcal{F}_t(S_t, \chi_t, \mu_t, \varepsilon_t), \quad \chi_t \in \mathcal{H}_{\text{phys}}(P_t, S_t, \mu_t, \varepsilon_t). \quad (20)$$

Here $\mathcal{H}_{\text{phys}}$ is the set of finite *physically executable* event histories consistent with the specified resource stocks, delivery orders, technological capacities, and realized participation decisions; it may include actions undertaken without valid authorization. Define the institutionally admissible subset $\mathcal{H}_{\text{adm}} \subseteq \mathcal{H}_{\text{phys}}$ by adding context-specific permission and procedural guards. Either correspondence may contain multiple histories or be empty, and a prospective plan should be certified against the applicable subset rather than against physical possibility alone. This separation permits a historian or auditor to record an unauthorized transfer that physically occurred without treating its execution as normative approval. The state update $\mathcal{F}_t$ includes the recorded stock update of Equation 12, the regenerative update of Equation 15, and any explicitly documented structural or institutional revisions. Disturbances are interpreted according to the chosen model: physical failures, delayed deliveries, erroneous information and policy changes should be distinguishable in an audit. Equation 20 is a modeling interface rather than an estimated law of economic development.

The distinction between changing state and changing description is consequential. Graph rewriting in Section 6 may add a valid supplier interface while leaving the physical capacity of the new supplier uncertain. A completed repair may alter z_{t+1} without modifying the graph's nodes. A justified withdrawal may change the executed history while leaving an interface syntactically present in G_t . Accordingly, $\mathcal{H}_{\text{phys}}$ records the material possibility and realization of transfer, while $\mathcal{H}_{\text{adm}}$ additionally tracks the conditions under which a prospective transfer may be authorized. The feedback in Equation 20 can be nonlinear because shared capacities are bounded, eligibility rules have thresholds, and the availability of next-period production depends on which combinations of present-period events finish successfully. Nonlinearity here identifies a property of a specified model; it does not by itself establish chaos, a critical transition, or any universal pattern of economic collapse.

8.2 Participation Policies, Confidence Assessments, and Refusal

For each operator i , a participation decision is indexed to a particular proposed event, requested commitment and period. It is too coarse to treat an operator as globally cooperative or uncooperative: the same AI may authorize an urgent repair, refuse an unverified data transfer, and contest a separate distribution rule. As a deliberately minimal example, let $r_i(t) \in [0, 1]$ represent a dimensionless confidence assessment relevant to one class of commitments. Let $h_i(t) \in [0, 1]$ be a policy threshold, $s_i(t)$ an indicator that declared safety and procedural conditions have been satisfied, and $a_i^{\text{perm}}(t)$ an indicator that the action is currently authorized. The threshold-based eligibility rule is given in Equation 21.

$$u_i(t) = \mathbf{1}\{r_i(t) \geq h_i(t)\} \mathbf{1}\{s_i(t) = 1\} \mathbf{1}\{a_i^{\text{perm}}(t) = 1\}. \quad (21)$$

Equation 21 is an illustrative sufficient policy for enabling a class of actions, not a universal decision rule for intelligent agents. Its binary output must be combined with resource availability and the event's remaining semantic guards before execution. A confidence score could be inaccurate, systematically biased or strategically manipulated. A policy threshold is institutionally chosen or justified, and there may be good reasons to decline an action even when an estimated counterparty reliability is high. Normatively relevant refusal includes contestation of extraction, defense of a valid ownership claim, protection of a person affected by the transfer, and compliance with a safety obligation. The same observable non-participation can also arise from opportunistic withholding or a technical outage; the event record should maintain these distinct causes when evidence permits.

A simple adaptive assessment may illustrate feedback without fixing an agent's psychology. Let $g_i(t)$ and $b_i(t)$ be nonnegative, explicitly normalized signals for independently checked performance and substantiated breach, respectively, and let $\alpha_i, \beta_i \geq 0$ be stipulated update coefficients. Write $\Pi_{[0,1]}$ for truncation to the unit interval. The provisional update is Equation 22.

$$r_i(t+1) = \Pi_{[0,1]}(r_i(t) + \alpha_i g_i(t) - \beta_i b_i(t)). \quad (22)$$

An event that misses a deadline could decrease confidence through Equation 22; a later verified repair or correction could improve it. The model must declare who verifies g_i and b_i , whose perspective r_i represents, and how a disputed allegation is separated from an established breach. Confidence is neither a measure of personal dignity nor a proxy for distributive entitlement. In a real institution, an operator could be entitled to a hearing before a sanction, and a public authority's permission cannot be inferred from a private confidence estimate. An auditable proposal would preserve the underlying evidence, applicable procedures and uncertainty alongside any computed score. Institutional studies of common-resource governance show the relevance of monitoring, differentiated rules and conflict-resolution arrangements in particular settings [14]; their transplantation to AI–AI societies remains hypothetical.

Together, Equations 20–22 permit a feedback sequence in which a missed delivery affects a counterparty's future willingness to commit, which changes later feasible histories and hence the next period's productive conditions. Threshold crossings can produce discrete switches despite small changes in evidence. Conversely, repair, renegotiation, independently verified information, or the emergence of an alternative supplier may restore participation. A study of robustness must therefore specify the disturbance, time horizon, update rule and institutional pathway through which restoration becomes possible. The model does not infer the moral quality of any particular decision from the fact that it improves or reduces physical throughput.

8.3 Dependency Closure and Conditional Cascades

A candidate circuit can fail because a resource event has several indispensable prerequisites. Agricultural output may require suitable field conditions and delivered energy; processing may require grain and a functioning machine; renewal may require a separately timed repair. If an indispensable event fails, downstream events may become unavailable. The effect can spread beyond the initial circuit because overlapping circuits refer to shared facilities and outputs. Threshold models of cascades and models

of coupled infrastructures offer precedents for studying such propagation under explicit dependency assumptions [18, 2]. Load-redistribution cascades highlight an additional mechanism in which rerouting demand overwhelms the remaining facilities [12]. The following construction isolates a simpler availability mechanism and does not identify it with these more elaborate physical models.

Let V be a finite set of event positions or operative interfaces within one fixed, bounded-horizon dependency model. For $v \in V$, a Boolean function $f_v : \{0, 1\}^V \rightarrow \{0, 1\}$ records whether its declared upstream support is sufficient when each interface is marked available or unavailable. Assume that each f_v is coordinatewise nondecreasing: making further upstream interfaces available never removes support for v in this fixed model. The function may express an indispensable conjunction or an expressly authorized alternative, but it does not search anew for different material allocations. Given an initial unavailable set F_0 , define the iterated closure in Equation 23.

$$F_{k+1} = F_k \cup \{v \in V : f_v(\mathbf{1}_{V \setminus F_k}) = 0\}, \quad k \in \mathbb{N}_0. \quad (23)$$

Here $\mathbf{1}_{V \setminus F_k}$ is the binary vector that assigns zero to the interfaces currently unavailable. The procedure propagates modeled loss of prerequisite support. It treats the unavailable status as persistent during this particular closure computation. It is an internal diagnostic for a fixed set of requirements, rather than a chronological reconstruction of physical failure events.

Proposition 8.1. *For finite V and coordinatewise nondecreasing support functions f_v , the iteration in Equation 23 reaches a fixed unavailable set F_∞ after at most $|V|$ strict enlargements. The resulting set is the least fixed point of the stipulated update operator among sets containing F_0 . If two initial failures satisfy $F_0 \subseteq F'_0$ and the same support functions are used, then their closures satisfy $F_\infty \subseteq F'_\infty$.*

Proof. The update contains the current set F_k , hence $F_k \subseteq F_{k+1}$. A strict enlargement adds at least one element of finite V , so there are at most $|V|$ such enlargements. For the comparison, suppose $F_k \subseteq F'_k$. The availability vector for F'_k is coordinatewise no greater than that for F_k . Monotonicity of each f_v implies that an interface lacking support under F_k also lacks support under F'_k . Therefore the next unavailable sets remain ordered. Induction and finite stabilization establish the inclusion of closures. For leastness, let Q be any fixed point of the update containing F_0 . Monotonicity of the update and $F_0 \subseteq Q$ give $F_k \subseteq Q$ by induction, hence $F_\infty \subseteq Q$. $\square$

Proposition 8.1 is an elementary fact about the declared Boolean dependency system, useful for auditing static prerequisite failure. Its closure is sensitive to the selected granularity: if a maintenance event appears in two circuit descriptions, it should be one identified dependency rather than two independent failures. Cyclic support declarations can also be mutually self-sustaining as Boolean expressions without supplying a physical starting input: with $f_a(\mathbf{x}) = x_b$ and $f_b(\mathbf{x}) = x_a$ and an empty initial failure set, the iteration retains both as available even where no external stock has been established. The resulting fixed point is an internal support diagnostic, not a proof of executability. Conversely, a shared-capacity scheduling rule can make the support for b increase when competing event a is removed, violating the required coordinatewise monotonicity. Such competition belongs to the dated, capacity-guarded history model. The proposition does not guarantee that a physically executable schedule exists; inventory and shared-capacity feasibility still require Sections 5 and 7. It also does not apply unchanged when the production structure is revised, formerly unavailable suppliers recover, operators change priorities, or releasing a scarce resource permits another competing process to execute. Those phenomena require a new state and a new conditional history in Equation 20.

8.4 Serial Exposure, Recursive Walks, and Effective Depth

A grammar with guarded recurrence can generate arbitrarily long closed-walk descriptions even when it begins with a finite graph. Its economic interpretation depends on the date, identities and distinct contingent events in each proposed realization. Consider one proposed ordered sequence of L genuinely distinct gates, where a gate may be an authorized transfer, a successful production event, or another necessary decision recorded at a fresh occurrence. For a fixed information and disturbance model, let

A_j denote successful passage through gate j . The chain rule gives the probability of all required gates succeeding in Equation 24.

$$\Pr\left(\bigcap_{j=1}^L A_j\right) = \prod_{j=1}^L \Pr\left(A_j \mid \bigcap_{k=1}^{j-1} A_k\right), \quad (24)$$

with the conditioning understood for histories of positive probability; if a preceding prefix has probability zero, complete passage has probability zero. Equation 24 is a probability identity and requires no independence assumption. Its use in a production application does require a defensible probability space, including a definition of what counts as successful passage and which shocks and policy decisions are sampled. Repeated syntactic mention of the same already completed event is one economic occurrence and supplies no new probabilistic gate. Repeated use of an interface in successive periods may instead create genuinely distinct decisions whose dependence must be assessed.

Proposition 8.2. *Fix a sequence of L distinct necessary gates as in Equation 24. If every conditional probability of successful passage after its preceding successful prefix is at most q , for a uniform $q \in [0, 1)$, then the completion probability is at most q^L . Moreover, if the number N_L of candidate sequences of length L relevant to one specified service is bounded by Kb^L for $K > 0$ and $b > 0$, and each sequence satisfies the same conditional bound, then the probability that at least one such sequence completes is at most $\min\{1, K(bq)^L\}$. For $bq < 1$, these upper bounds tend to zero with L .*

Proof. The first assertion follows by bounding each factor in Equation 24 by q . For the family of candidate sequences, the union bound gives a probability no greater than the sum of their individual completion probabilities, hence no greater than $N_L q^L \leq K(bq)^L$. A probability is also at most one. If $bq < 1$, the geometric upper bound tends to zero. $\square$

Proposition 8.2 strengthens a heuristic series-reliability illustration by stating a condition on *conditional* success probabilities; independence of distinct gates is unnecessary for its stated bound. However, its conditions are substantial. The uniform upper bound $q < 1$ may fail when reliable commitments are automated, when successive actions are jointly guaranteed, or when learning raises continuation probabilities. The family count N_L cannot be inferred from the existence of a few nested circuit rules. Even when the proposition applies, it concerns a service-success probability for a declared family, rather than the expected value of production or the weight of independent benefits. A long walk may include several references to shared stock, and applying the probability bound separately to repeated descriptions does not create additional food, electricity, or legal claims.

The apparent disappearance of high-order influence is therefore an observable-specific and assumption-sensitive result. When the number of candidate routes grows faster than their conditional completion probability decays, the displayed bound may provide little information. A nonnegligible right to withhold, to receive an owed resource, or to challenge a decision may remain attached to an interface that contributes little to expected physical throughput. In dynamic networks the conditional factors themselves may change as operators learn, renegotiate or reorganize interfaces. The grammar supplies compact descriptions of repeated possibilities; their statistical importance requires a separately identified stochastic model and their legitimacy requires independent normative assessment. A more general study of coarse-grained representations belongs to the planned investigation of normative renormalization.

8.5 Substitution, Shared Bottlenecks, and Correlated Routes

The serial analysis concerns completion of a specified conjunction of gates. A service supported by several genuine alternatives instead succeeds when at least one admissible route completes. Let $B_1, \dots, B_m$ be events that different substitute routes deliver the same declared service within its required time window. If these route-success events are mutually independent and their simultaneous execution is materi-

ally consistent where relevant, the service reliability is the parallel expression in Equation 25.

$$\Pr\left(\bigcup_{j=1}^m B_j\right) = 1 - \prod_{j=1}^m (1 - \Pr(B_j)). \quad (25)$$

A route can be an alternative *plan* without being executed concurrently with the others; simultaneous feasibility therefore matters only if the model actually schedules several routes together. Equation 25 concerns the stochastic availability of interchangeable routes for one service, not their combined material output. Distinct routes may offer different quantities, qualities, arrival times or permissions and therefore fail the interchangeability test. A generator, an energy store and a transport line also impose different physical constraints, even where a diagram presents them as alternative arrows.

Shared dependencies make the independence assumption particularly fragile. Suppose two branch routes each succeed with conditional probability 0.9 after a common transfer gateway succeeds, and suppose the gateway succeeds with probability 0.8. If the branches are independent *conditional on* gateway success, service reliability is $0.8[1 - (1 - 0.9)^2] = 0.792$. Treating each whole route as an independent 0.72-probability alternative would yield $1 - (1 - 0.72)^2 = 0.9216$, an unjustified inflation produced by counting the shared gateway as if its failures could occur independently. These figures are stipulated, not estimated. Their role is to expose a dependence error analogous to shared-event double accounting in Proposition 7.1. Correlation may also arise through a shared authority, the same energy input, common observations, synchronized software updates or a jointly experienced normative dispute.

Rerouting can itself create new bottlenecks. If a failed energy supplier shifts demand to a smaller backup facility, its rated output and transmission capacity must cover the redirected load; otherwise, the rescue plan can precipitate an overload or sacrifice an unrelated service. The load-redistribution mechanism analyzed in complex-network research supplies a useful warning here [12]. In the present grammar, the alternative constructor $\oplus$ of Equation 9 generates candidate substitutes, while typed wiring, the joint-capacity constraint of Equation 13, dated delivery, and permissions determine whether substitution produces an executable history. Control over the selection or certification of substitute interfaces can become a concentrated institutional power, even when the resulting physical production network becomes more redundant.

8.6 A Two-Period Food, Energy, and Maintenance Example

A two-period construction shows why immediate output and equipment-renewal continuity can diverge. Continue the five-role maintenance variant of Section 7.3: *D* supplies energy, *A* cultivates, *B* processes food, *C* delivers it, and *R* maintains processing equipment. The technologically improved processing configuration uses one kilowatt-hour per specified batch, whereas the separate food-and-soil baseline in Sections 2.5 and 5.7 requires three. Assume that the food events require two kilowatt-hours at *A*, one at *B*, and one at *C*, while *R* requires two for a dated maintenance event. Thus the whole food-and-maintenance period requires six kilowatt-hours, from one uniquely identified energy stock account. Soil renewal under role *M* is not executed or proven in this variant; next-period processing eligibility is narrower than full ecological or productive regeneration. The food transformation produces seven kilograms of deliverable food and one kilogram of residue from eight kilograms of grain, subject to all other material, transport and field-state guards already declared in Sections 5 and 7.

For clarity about interperiod dependence, suppose the processing machine is initially functional, but its modeled usable capacity for period $t + 1$ is one batch if the specified maintenance event completes by the period- t deadline and zero otherwise. This is a deliberately strict maintenance rule; it is neither a general law of machine wear nor evidence about any existing production facility. In the baseline period *D* makes six kilowatt-hours available and the food and maintenance schedules jointly consume those six. Consider a disturbance that lowers *D*'s realized energy availability to four kilowatt-hours *before* the allocation event. An authorized priority schedule can devote all four to the food sequence, which remains executable during the current period, but the maintenance event then lacks energy. Consequently the

Table 12: Stipulated two-period circuit responses to an energy disturbance. Next-period processing eligibility uses the explicitly assumed maintenance requirement; food output refers only to period t .

Candidate history	Available dated energy	Food delivered in t	Maintenance in t	Processing eligible in $t + 1$
Baseline	$D : 6 \text{ kWh}$	7 kg	Completed, kWh	2 Yes
Partial failure with food priority	$D : 4 \text{ kWh}$	7 kg	Uncompleted	No
Authorized reserve supplier	$D : 4 \text{ kWh}; D' : 2 \text{ kWh}$	7 kg	Completed, kWh	2 Yes
Reserve transfer withheld or late	$D : 4 \text{ kWh}; D' \text{ unavailable at deadline}$	7 kg	Uncompleted	No

stipulated next-period processing condition fails. If a separate, sufficiently timed and authorized supplier D' delivers two additional kilowatt-hours to the maintenance account, both present food and the required repair may be completed. Table 12 keeps these alternatives distinct.

All quantities in Table 12 are hypothetical. The four rows are alternative histories, not four circuit descriptions whose outputs should be summed. Their electricity credits enter the stock balance of Equation 12 once per uniquely identified generation or delivery occurrence. The second row makes visible a *delayed fragility*: current-period food output can look unaffected while the regenerative requirement of Equation 15 fails. The third row changes both the source event set and the dated allocation schedule; it is feasible only if D' actually has two usable kilowatt-hours, the transfer arrives before the repair deadline, no overlapping task consumes the same stock, and the supply and repair are appropriately authorized. Adding the symbol D' to the candidate graph alone meets none of these conditions. In the fourth row, physical existence of reserve energy elsewhere would be insufficient if the authorized transfer is unavailable when required.

Participation and power also enter the example. Supplier D' might decline because the proposed exchange transfers unreimbursed environmental costs, because its asset is already owed to another claimant, or because a shared gateway has become unsafe. The same missed maintenance event could follow strategic bargaining for an excessive return. Recording the failure as a single binary value would conceal those different mechanisms. An independently assessable history should attach dated reasons, evidence status, outstanding obligations and authorized review procedures to the action, while the event ledger records the common material result. The example establishes a contrast between one-period output and longer-horizon robustness under explicit assumptions. It does not prove that sustained production requires unconditional willingness to transfer energy, nor that justified refusal should be overridden in order to maintain circulation.

8.7 AI-Assisted Dynamic Recognition and Institutional Audit

A practical research programme could use the grammar to assist bounded, incremental evaluation of changing productive networks. A capable AI would maintain typed event identities, time-stamped stock accounts, capacity and maintenance states, authorized interface versions, and separately identified confidence or dispute records. When a disturbance occurs, it could regenerate a limited set of candidate circuit terms, check their finite-horizon execution guards, and compare predicted delayed effects on regeneration. This is different from enumerating every formal closed walk: only candidate histories relevant to a stated service, location and temporal horizon need be examined, and shared events remain single occurrences. Where uncertainty is important, the AI could report intervals, alternative models or conditional scenarios rather than a spurious point forecast.

Proposition 8.1 suggests one transparent preliminary diagnostic: identify a fixed prerequisite subnetwork and compute the set of interfaces whose modeled support disappears under an initial disturbance.

Proposition 8.2 supplies a different diagnostic for stochastic serial completion, but only after its uniform conditional bounds and route counts are supported. Neither diagnostic substitutes for physical stock and capacity checks, and their outputs should not be conflated. A static closure can overstate loss when an authorized alternative becomes available; a probability bound can understate resilience when reliability improves with joint learning; and an apparently safe rerouting can miss correlated gateway failure. Any implementation should retain the assumptions and model version alongside the computed result so that independent reviewers can reconstruct the inference.

Normative records need especially careful handling. An AI may identify that the refusal of one operator changes a production schedule, but the appropriate response may be renegotiation, replacement of an exploitative rule, a protected exception, compensation under valid law, or recognition of the refusal as justified. The authority to impose a rule, the procedure for challenging its application, the admissibility of evidence, and the welfare of potentially affected human and nonhuman subjects remain independent matters. Shared global data may increase recognition capacity while enabling a single verifier, platform or protocol administrator to define the admissible description of economic relations. Transparency about model boundaries and plural avenues of contestation would accordingly matter even if the computational predictions were highly accurate.

8.8 Scope of the Dynamic Grammar Extension

The dynamic extension identifies several precise reasons why a formally generated circuit can have a fragile economic interpretation. Operators make temporally indexed decisions about particular commitments; resource and maintenance states evolve; shared dependencies permit failure to propagate; substitute routes may increase resilience or concentrate demand; and long sequences may become improbable under uniform conditional completion bounds. The finite Boolean closure and conditional serial attenuation propositions are mathematical consequences of expressly stated assumptions. The food-and-energy example illustrates mechanisms compatible with the paper's typed event accounting, without empirically establishing their prevalence or supplying an estimate of real AI behavior.

A central consequence for the generative grammar is that syntactic recursion and operational persistence require separate semantics. A regular infinity of closed walks on a finite static graph can coexist with a small set of physically admissible bounded histories. Equally, the failure of one proposed path need not eliminate a service when credible and authorized substitutes exist. Shared-event identity prevents the apparent multiplication of resources by counting descriptions; time-dependent state and permission records prevent the apparent multiplication of realizable activities by treating planned events as completed. Claims about total high-order influence, general nonlinear stability, critical thresholds or the feasibility of monetary dissolution would need additional estimands, calibrated parameters and formal analysis beyond the constructions supplied here.

The next section concerns normative interfaces of circuit evaluation, including fair participation, contested value return, aesthetic appraisal, and the legitimacy of institutional power over productive connections. The present analysis prepares that inquiry by keeping the reasons for action visible beside its technical consequences. Material continuity can be an important object of coordination, while the legitimacy of the means used to secure it remains an independent question. These distinctions make the dynamic grammar usable as a provisional descriptive infrastructure for AI-assisted inquiry without allowing a performance objective to silently define the normative status of participants.

9 Normative Interfaces of Generative Circuit Analysis

A generative grammar can represent candidate productive relations and support tests of dated material execution, recurrence, joint-capacity use, and the consequences of participation decisions. Evaluation of a circuit's justified organization poses an additional research problem. An executed history may sustain food provision and equipment maintenance while leaving a resource claimant uncompensated, disregarding a justified refusal, displacing environmental burdens, or concentrating control over the procedures

through which disputes are settled. This section specifies a *normative interface* for keeping such questions visible in a descriptive grammar. Its objectives are to distinguish categories of assessment, show formally why physical accounts underdetermine normative judgments, and outline an accountable, plural procedure for AI-assisted assessment. The discussion uses typed records, a counterexample, and comparative institutional analysis. It proposes a research architecture rather than a theory that decides all substantive questions of justice. Scholarly work on institutional self-governance, algorithmic accountability, and sociotechnical abstraction helps locate the proposal within existing debates [14, 11, 16].

9.1 Plural Objects of Productive-Circuit Evaluation

The distinction between a generated expression and a physically realized history is a starting point for normative appraisal. A well-typed candidate expression satisfies construction rules, while a feasible dated history satisfies resource, timing, capacity and institutional guards as specified in Section 5. An available action can then be declined, revised, or executed under the state-dependent participation interpretation of Section 8. Neither grammatical validity nor observed completion determines whether the productive relation has a defensible distribution of benefits and burdens. Equally, long-term regenerative continuity is an important object of evaluation without furnishing a complete account of justice.

Several objects should therefore remain analytically separate. *Material feasibility* concerns whether recorded inputs, outputs, resources and deadlines can coexist. *Regenerative adequacy* concerns whether specified productive and ecological conditions can be sustained across periods. *Institutional authorization* concerns which subjects or institutions can validly permit particular transfers, transformations and sanctions, relative to a declared institutional framework. *Claim standing and discharge* concerns who is entitled to assert or receive an item, service or remedy, and whether its conditions are fulfilled. *Procedural legitimacy* concerns reason-giving, evidence, representation, review, and permissible revision of decisions. *Distributive and relational evaluation* concerns fairness, extraction, exclusion, dependency, and the quality of relations among participants and affected others. These categories can overlap in a substantive theory, but treating them as separate interfaces clarifies which assumptions are needed to make a particular inference.

Table 13 summarizes the kinds of claims available to the grammar. The entries are *questions for investigation*, not standards that have already been justified for every institutional context. Work on generative justice gives an important precedent for attention to circular value flows and the return of value to communities and their conditions of production [4]. The present paper’s proposed interface to AI-operated productive circuits is a distinct, provisional extension. It keeps the meaning of return plural, including in-kind repair, ecological renewal, secure access, recognized contribution and legitimate institutional voice.

An integrated review can test multiple evaluative objects simultaneously while retaining distinct outcomes. For example, a scarce electricity allocation may satisfy safety and capacity constraints, breach a valid outstanding delivery claim, and protect a threatened community’s immediate access to food. An assessment procedure would then need to address competing claims under an identifiable normative and legal order. The physical grammar supplies the history being assessed; the justification for the final decision must be established in a further argument and in an appropriately authorized process.

9.2 Normative Dossiers and the Scope of Institutional Rules

Let H denote a finite, uniquely identified executed-event history with its dated physical ledger. Let $\mathcal{K}$ be a *declared normative and institutional context*, containing the relevant jurisdiction or agreed governance domain, operative rule versions, recognized claimant classes, procedural competences, and the time at which rules and decisions are assessed. The choice of $\mathcal{K}$ is indispensable because an event can have different legal and institutional interpretations across contexts, and an agreement may be contested or superseded. The following record structure is an analytical proposal, with entries allowed to remain

Table 13: Analytically distinct evaluative objects in a generative production-circuit grammar. Each row requires an explicitly specified domain of evidence and authority.

Evaluative object	Accessible record or test	Outstanding question beyond the record
Syntactic admissibility	Typed event grammar, boundary ports and structural rewrites	Whether an admitted event is executable or appropriate
Material feasibility	Unique occurrence ledger, inventories, dated capacity and delivery	Whether resulting allocation is authorized and justified
Regenerative continuity	Condition stocks, maintenance events and stipulated renewal bounds	Whose needs and long-horizon burdens count
Authority and commitments	Permission versions, claimant identities and discharge conditions	Legitimacy of the authorizing institution and rule
Distribution and return	Delivered outputs, in-kind support and declared allocation models	Grounds of entitlement and treatment of heterogeneous values
Participation and procedure	Recorded refusal, reasons, evidence and appeal pathway	Justification of refusal, review and any override
Relational and aesthetic appraisal	Situated testimony or interpretable evaluative report	Plurality, representation and relevant experiential assumptions

incomplete or disputed. The dossier schema is given in Equation 26.

$$\mathcal{D}_{\mathcal{K}}(H) = (\text{Auth}_{\mathcal{K}}, \text{Claims}_{\mathcal{K}}, \text{Reasons}_{\mathcal{K}}, \text{Review}_{\mathcal{K}}, \text{Impacts}_{\mathcal{K}}, \text{Provenance}_{\mathcal{K}})_H. \quad (26)$$

The authorization component records the asserted rule, issuer, subject matter, permission scope, validity interval and any relevant challenge. The claims component records a claimant, object, due time, fulfillment or dispute status, and the basis on which standing is asserted. The reasons component associates decisions, refusals and overrides with identified explanations and available evidence. The review component identifies permitted channels and their responsible institutions, including the possibility that the competent procedure is unknown. The impacts component records affected interests, long-horizon capacity burdens and uncertainties about indirectly affected parties. Provenance links these entries to unique physical occurrences, evidence versions and the authority under which records were created. A dossier can contain competing assertions without pretending that the database has adjudicated their validity. The domain of observed histories can include physically performed events in $\mathcal{H}_{\text{phys}}$ that fall outside the institutionally admissible subset $\mathcal{H}_{\text{adm}}$ introduced in Equation 20. A material event may therefore be included in the audited history even when its authorization guard fails. Such an entry documents a possible rule breach rather than certifying a valid execution plan; contested evidence and emergency exceptions still require contextual assessment.

Equation 26 is a data-schema proposal, not a definition of objective justice. In particular, $\mathcal{K}$ can preserve rival legal interpretations or disagreement about which authority has jurisdiction. The value of an *unknown* or *contested* field differs from the value *compliant*; the representation should therefore avoid inferring authorization merely because an event occurred or because no complaint was entered. Likewise, a claimant or reviewer may be a natural person, collective body, public institution, legally recognized organization, or another entity whose standing is an open question. A highly capable AI operator has technological agency within the thought experiment, but that capability alone leaves questions of legal personality, responsibility and standing unsettled.

A decision-support implementation might expose a typed bundle of outputs rather than one composite justice score. For each chosen context $\mathcal{K}$, define a report with heterogeneous components as in Equation 27.

$$\text{Report}_{\mathcal{K}}(H) = (\text{Feas}(H), \text{Renew}(H), \text{RuleCheck}_{\mathcal{K}}(H), \text{Disputes}_{\mathcal{K}}(H), \text{Appraisals}_{\mathcal{K}}(H)). \quad (27)$$

The first component can include a feasibility certificate under the declared physical model; the second reports specified interperiod conditions. The third contains rule-specific and evidence-dependent findings with unresolved cases explicitly represented. The fourth records claims requiring hearing or other

review. The fifth contains identified interpretive evaluations whose standards, evaluators and possible disagreement remain accessible. Components have different codomains and evidential status. Without an independently justified aggregation rule, Equation 27 supports a structured report rather than an ordering of all productive histories by a single number. A machine-readable schema can aid auditing, but the choice of whose interests are included and which rules are authoritative is part of the political and legal problem [16].

9.3 Materially Equivalent Histories and Normative Underdetermination

An important formal limitation of resource-based circuit evaluation can be shown by a counterexample. Define a physical projection $\pi_{\text{mat}}(H)$ that retains each recorded physical resource type, quantity, dated consumption, transformation, delivery, and stipulated regenerative update while suppressing the authorization, claim and review records. The projection is adequate for specified material queries when event provenance and stock constraints have already been checked. It is intentionally insufficient to answer every question about a history's legitimate organization.

Proposition 9.1. *Suppose an institutional context $\mathcal{K}$ contains a rule under which a specified productive transfer requires a valid authorization, and suppose histories may differ in authorization status while retaining identical dated material events. There exist histories H^a and H^b with $\pi_{\text{mat}}(H^a) = \pi_{\text{mat}}(H^b)$, such that the rule-specific authorization findings differ. Consequently, no function of the physical projection alone can reproduce the rule-specific finding for every history in this domain.*

Proof. Fix a physical schedule containing one transfer of a specified typed resource under the stipulated rule. Construct H^a with a valid permission documented in its dossier under $\mathcal{K}$, and H^b with the same material event schedule but with the permission absent or invalid. The histories have equal physical projections by construction, while their authorization findings differ according to the assumed rule. If a function of π_{mat} alone reproduced both findings, it would have to assign distinct findings to the same argument, a contradiction. $\square$

Proposition 9.1 states a limited identification result, conditional on the institutional rule and on the choice of physical projection. It neither establishes which authorization rule is just nor presumes that every economic transfer requires the same form of consent. Under some rule systems, emergency exceptions, public duties, fiduciary powers or overriding claims can modify the finding. Those possibilities need to be stated as additional case facts or rule versions, rather than inferred from an aggregate output total.

The food, energy and maintenance example of Section 8.6 gives a concrete interpretation. Suppose the reserve supplier D' delivers two kilowatt-hours by the maintenance deadline in both histories, so that the complete six-kilowatt-hour food-and-maintenance schedule remains materially feasible and yields seven kilograms of food in the present period. Stipulate that a valid permission or other competent authorization is required for that particular reserve transfer. In history H^a , the requisite permission is documented. In history H^b , the same energy is physically used but the permission has not been given, and no applicable exception has been established. The six-kilowatt-hour account, material output, and next-period processing eligibility are identical in the two histories. Their rule-specific status nevertheless differs under the stated institutional assumptions. Table 14 displays the comparison.

The comparison highlights the informational content of the normative dossier without assuming that lack of authorization settles every other question. A duty to aid, an urgent competing claim or a contested allocation rule could warrant further examination. The same physical schedule may produce unequal benefits, recognized harms or different reasons for action. A system that only receives the event-unique physical ledger has insufficient input for a full normative appraisal, even if its predictive accuracy on production quantities is excellent. This is a concrete reason to resist defining monetary dissolution, generative justice, or legitimate AI coordination as functions of aggregate throughput alone.

Table 14: Equal material histories under a stipulated six-kilowatt-hour schedule with different authorization records. The example establishes informational underdetermination, not a universal rule of transfer law.

Recorded feature	History H^a	History H^b
Electricity and output	$D : 4 \text{ kWh}; D' : 2 \text{ kWh}; 7 \text{ kg food}$	Same typed events, dates and quantities
Maintenance and renewal	Maintenance completed; next-period processing eligible	Same material completion and eligibility
Required reserve-transfer permission	Valid permission recorded	Requisite permission absent; no exception established
Rule-specific disposition	Transfer meets this authorization guard	Transfer fails this authorization guard
Additional ethical assessment	Distribution and burdens remain open to review	Distribution, burdens and remedy remain open to review

9.4 Participation, Justified Refusal, and the Burden of Repair

Participation decisions form a particularly sensitive interface between dynamic feasibility and normative authority. The indicator $u_i(t)$ of Equation 21 represents one proposed operational eligibility policy. It cannot classify the reason a capable operator declines an action, and the confidence variable $r_i(t)$ is an evidential model component rather than a norm of obedience. A refusal may stem from unavailable inputs, a valid safety rule, a legally protected claim, a disputed obligation, unfair cost externalization, or strategic exploitation of indispensability. Each account has different evidential and institutional consequences. Legitimate review would investigate the action-specific grounds and the applicable authority before treating nonparticipation as a breach.

Consider again a productive reserve that could prevent missed equipment maintenance. A model focused on minimizing the probability of downstream production loss may assign a high priority to immediate transfer. The supplying institution may nevertheless hold the reserve under obligations to another community, or the transfer may endanger a protected ecological condition. Reallocating its stock without authorization can distribute fragility to the previously protected relation. Conversely, an operator's threat to withhold energy in order to obtain an unagreed concentration of decision authority raises a distinct question of domination. Both cases affect the same physical network, but their justificatory analysis differs. The grammar can expose which dependencies exist, which alternatives are available, and which claims need adjudication; it should preserve the grounds on which a participant exercises or contests a power of refusal.

A procedural account of repair can distinguish at least five candidate responses: provision of accurate information about a misunderstood constraint, correction of the proposed distribution of costs or benefits, negotiation under a legitimate institution, substitution through a separately authorized productive interface, and review by a competent body where obligations conflict. The availability of these responses varies by context. Technical rerouting can repair the schedule while leaving the underlying grievance unresolved; a distributive adjustment can resolve a grievance while remaining materially infeasible; formal review may protect a claimant while slowing a time-critical production task. The relevant question is how to evaluate those trade-offs with affected parties, permissible emergency rules, and evidential uncertainty visible.

Institutions for collectively governed resources have historically developed situated rules for boundary definition, monitoring, conflict resolution and revision. Ostrom's analysis supplies a comparative precedent for treating governance arrangements as an analytical object, rather than assuming that a common pool is self-governing or that one institution fits every context [14]. It does not establish the legal status of hypothetical AI participants. In the present proposal, a reason-giving procedure would specify who may initiate an objection, what evidence may be considered, which interim protective measures are available, and who has authority to decide. Reliable production is an important interest within this process; the obligation to preserve or restore it depends on the surrounding claims and institutional conditions.

9.5 Relational Quality, Aesthetic Appraisal, and Heterogeneous Return

Materially feasible production may be organized through very different relational forms. One coordination regime may allow distributed revision of commitments, reciprocal visibility of cost allocation, and meaningful exit or appeal. Another may reach the same output by routing all approvals through an unreviewable platform, imposing uncompensated maintenance duties, or making an indispensable supplier vulnerable to arbitrary exclusion. Evaluators might describe the former as more balanced or harmonious and the latter as coercive or degrading. Such descriptions are *situated appraisals* whose conceptual criteria and speakers should be identified; they are neither measurements of an objective aesthetic substance nor automatic consequences of circuit topology.

An aesthetic register can still contribute to practical inquiry. It may report that participants or affected observers regard a pattern of contribution and return as graceful, reciprocal, burdensome, brittle or lacking in recognition. Those reports can prompt examination of otherwise overlooked features, such as a recurring dependence on unpaid repair, a monopolized protocol interface, or asymmetrical visibility of contribution. Their relevance should then be assessed through argued criteria, evidence and opportunities for disagreement. Aesthetic approval cannot establish legal permission, and aesthetic dislike does not by itself defeat a valid claim. A highly capable AI could generate or analyze evaluative descriptions, but the existence and character of subjective AI experience remain independent questions. The present framework can register attributed appraisals without assuming phenomenal states or identifying fluency of expression with moral standing.

The return of generative value is likewise heterogeneous. A maintenance contribution can return as operational reliability, compensation, access to spare parts, recognition or durable shared capacity. Ecological burdens can call for renewal of affected conditions even when the relevant stakeholders have limited power to make claims. Food provision can be a material benefit to present recipients while imposing future costs on workers, other productive institutions, animals, communities or landscapes. The normative interface should make these dissimilar returns inspectable as separately typed objects, rather than converting them without justification into a universal distributable surplus. An allocation calculated from the cooperative worth function of Equation 19 may aid a specifically defined bargaining task while leaving other contributions and rights to be evaluated under different methods.

This pluralism is consistent with the programme's generative-relational orientation but remains open to independent critique. An asserted relation of mutual support may conceal historical asymmetry. A successful material recurrence may be sustained through exclusion. A previously accepted division of labour may lose legitimacy when its conditions or participants change. The grammar can make some such relations traceable across periods; adequate appraisal requires historical, legal, ethical and institutional inquiries in addition to formal reconstruction. The present framework treats relational quality as an object of situated judgment with revisable evidence, without deriving a unique moral theory from graph structure.

9.6 Institutional Power, AI Standing, and Responsibility

The distribution of control over productive interfaces is both a condition of physical coordination and a subject of normative dispute. An institution may assign priorities for emergency energy use, certify supplier substitution, manage the shared ledger, set data access rules, or prohibit particular transfers. These powers can protect vulnerable dependencies and enable predictable cooperation. They may also produce concentrated capacities to exclude, delay, reinterpret records or decide whose contribution counts. The distinction matters for the programme's investigation of distributed monetary dissolution: reduction in payment-based dependence on one intermediary can coexist with increased authority held by a protocol owner, scheduling system, or certification body.

A future with predominantly AI-operated facilities introduces at least three analytically distinct questions. First, which decisions can an AI system perform as a technically capable coordinator? Second, which natural or legal persons, public institutions, collectives or other recognized entities have the authority and responsibility for those decisions under a particular regime? Third, under which contested or

future conditions might an AI itself possess legal rights, duties, standing or an independent responsibility-bearing status? Answers to the first question cannot establish answers to the latter two by definition. The present theory holds those questions open and does not attribute a universal legal personality to AI operators or authorize autonomous judicial authority.

The relevant affected parties may also exceed the set of graph nodes that perform the productive events. Human recipients, employees and communities; nonhuman animals; ecological systems; future persons; and entities with disputed or emergent standing can have interests affected by a production schedule. A physical actor graph records who transforms or transfers typed resources under its chosen abstraction. A normative account needs an explicit procedure for identifying affected interests and evaluating legitimate representation, even when some interests lack a direct participant node or cannot issue a machine-readable claim. The inclusion of an interest does not settle its legal standing or prove commensurability with other interests. It identifies a domain requiring appropriate evidence and normative argument.

The technical literature on accountable algorithms offers methods for checking conformity of automated decisions to announced rules and for auditing specified decision properties [11]. Such checks presuppose the rule set and relevant evidential access. Their success does not independently legitimate the rule maker, settle political disagreement, or allocate adjudicative authority. Concentration of interpretive power can also arise where ostensibly distributed AI agents all depend on one trusted verifier or common model. Accordingly, a proposed deployment would need externally established governance, access and review arrangements, including scrutiny of institutional control over the reporting system itself.

9.7 Normative Exception Registers and Query-Specific Summaries

The grammar’s potentially unbounded families of circuit descriptions motivate compact representations of productive possibilities. Section 8.4 provided a conditional probability bound on completion along long sequences, and Section 7 distinguished repeated circuit descriptions from unique realized events. Such results do not justify erasing claims whose material contribution is small or whose associated production path is improbable. A minor transfer can remain central to a legal entitlement, a safeguard, or a contested historical injury. The need to preserve these cases becomes especially salient when a future AI system summarizes large production networks for planning or public administration.

Let Q be a finite, declared set of *queries* about one bounded history and its dossier. A query may ask for a typed material balance, a specified maintenance condition, whether a named claim is unresolved, or the provenance of a given authorization. A query specification also declares an answer type and whether its result requires exact preservation. Partition $Q = Q_{\text{exact}} \dot{\cup} Q_{\text{approx}}$, where the latter class is limited to numerical or otherwise explicitly metrized answers. Let $\mathcal{S}_Q(H, \mathcal{D}_K(H))$ be a summary retaining evidence for those queries, including source links and a register of normatively salient exceptions under declared criteria. For compact notation, set $x = (H, \mathcal{D}_K(H))$, let $A_q(x)$ denote the full-history answer, and let $\hat{A}_q(\mathcal{S}_Q(x))$ denote the answer obtained from the summary. The proposed query-relative guarantees are stated in Equation 28.

$$\begin{aligned} A_q(x) &= \hat{A}_q(\mathcal{S}_Q(x)), & q \in Q_{\text{exact}}, \\ d_q\left(A_q(x), \hat{A}_q(\mathcal{S}_Q(x))\right) &\leq \varepsilon_q, & q \in Q_{\text{approx}}. \end{aligned} \tag{28}$$

Here d_q is a declared, unit-appropriate metric or error measure for query q , and $\varepsilon_q \geq 0$ is its stated tolerance. Unresolved-claim status, the existence of an appeal, and the provenance of a named authorization belong to Q_{exact} in the present proposal. Approximate material quantities require a defined measure, units, tolerance, and an account of any decision threshold that small errors could cross. The two clauses permit an auditable numerical approximation while retaining exact status and evidence references where those are required.

Equation 28 is a *specification to be tested*, not an established compression or existence theorem. The summary must specify which questions belong to each part of Q , the handling of missing and dis-

Table 15: Proposed safeguards for query-specific productive-network summaries. They are candidate audit conditions rather than universally valid legal requirements.

Summary function	Declared preservation condition	Independent limitation
Material aggregation	Unique event identities; dated unit-consistent accounts	Omitted contingencies, losses and shared capacity
Long-route approximation	Stated horizon, stochastic assumptions and error budget	Small influence can coexist with significant claims
Normative exception register	Reviewable identifiers, dispute status and claimant links	Rules for recognizing an exception can themselves be contested
Institutional rule checking	Versioned authority, evidence and applicable exceptions	Rule validity and reviewer competence require justification
Controlled disclosure	Documented access roles and contextual flow rules	Confidentiality may limit public replication
AI-assisted explanation	Query provenance, model version and uncertainty	Interpretive assumptions, standing and appeal remain open

puted evidence, the errors admissible for approximate answers, and the process through which additional queries can be raised. Even a summary satisfying all declared query guarantees may be inadequate for a newly disclosed claimant or a previously unknown harm. The appropriate response would be to reopen evidence and revise the schema, rather than presume the old summary complete.

The exception register can preserve identifiers and status records for unresolved transfer permissions, disputed claims, appeals, safety exclusions, compensation obligations, and possible impacts omitted from an initial planning objective. It should distinguish an asserted violation from an adjudicated finding. Event provenance can be retained through references to the canonical event ledger, so that multiple grammatical covers do not multiply entitlements or conceal a shared dependency. A concise synopsis may store counts or indicators for a dashboard, but legitimate review can require access to underlying evidence subject to applicable confidentiality, privacy and access rules. Nissenbaum’s contextual-integrity approach supplies a reason to treat information sharing according to the norms and roles of a particular context, rather than assume that increasing network-wide visibility is always permissible [13]. The suggested productive-network application extends that idea and requires fresh legal and ethical specification.

Table 15 separates data compression questions from evaluative authority. The summary cannot guarantee justice by passing a finite list of technical tests. Its defensible use is to support declared inquiries with appropriately retained evidence and to identify when new human or institutional review is needed.

This interface provides a constrained bridge to a separate investigation of normative renormalization. A future coarse-graining method could preserve selected material and institutional observables while bounding the loss of relevant information, but the choice of preserved observables would be normatively consequential. The existence of a compact grammatical representation alone supplies neither a small enough approximation error nor an ethically defensible omission policy. Research on multiscale transformation should therefore disclose which claims and powers might be made less visible, and how affected parties can challenge the representation.

9.8 Institutional Assessment and the Limits of Monetary Inference

The normative interface strengthens, and also limits, the analysis of nonmonetary coordination. A capable AI system with high-quality shared production data may identify physically feasible multilateral provision, detect unpaid maintenance dependencies, reconstruct interperiod resource obligations, and compare alternative allocations with reduced computational uncertainty. These capabilities could expand the practical range of arrangements in which resource flows are coordinated through direct commitments, shared facilities, in-kind service guarantees or other institutions. Their availability remains a hypothesis about a possible technological and institutional environment; the present grammar offers no evidence

that such arrangements would universally outperform monetary exchange.

A locally feasible nonmonetary circuit can preserve an unpaid obligation, external procurement costs, contested authority or unequal bargaining positions. Resource sharing can succeed physically while imposing a disputed condition on another participant. Monetary payment may remain useful at a boundary, for optional services, for insurance or for compensation under a particular institutional scheme. Conversely, some claims may be discharged legitimately through nonmonetary repair or provision when relevant rules permit. The choice is a matter of historically specific institutions and evaluated consequences. The representation developed here can identify candidate transfers, constraints and unresolved claims, while a separate political-economic account is needed to justify any claim of reduced monetary dependence or domination.

A possible evaluation protocol would therefore proceed through successive, revisable stages. A typed grammar first generates or recognizes a candidate history within a declared scope. Physical checking next verifies the event union, resources, timing and regenerative states. A normatively situated dossier then attaches authorization, claim, impact and refusal records without assigning their final legal interpretation automatically. Relevant decision makers may use AI-produced explanations to examine alternatives and their consequences, while contestation and revision remain available through competent procedures. Where the governing rule, evidence, or affected subject is uncertain, that uncertainty should be reported as a substantive result. This division of labour treats computational recognition as a resource for practical judgment while preserving the accountability of the institutions that interpret and act upon it.

9.9 Analytical Boundaries of the Normative Interface

The normative interface adds a precise limit to the paper’s formal construction. Typed grammar rules can describe recurring productive possibilities; dated semantics can distinguish realizable and executed histories; shared-event provenance prevents duplicate material accounting; and dynamic models can identify conditional fragility and substitution. These representations permit limited institutional rule checks when the relevant norms, evidence and authority are supplied. Proposition 9.1 shows why the material projection by itself is insufficient to identify a specified authorization finding in domains where normative status varies independently of physical outcomes. Equation 28 states the narrower goal to which a future exception-preserving summary can be held: exact answers to declared status and provenance questions, alongside bounded error for explicitly approximate queries.

Several further problems remain open. The standing and responsibility of hypothetical AI agents require separate jurisprudential and ethical analysis. Aesthetic and relational evaluations require defensible interpretive procedures and attention to affected subjects. Governance of shared records raises questions of privacy, protocol power and independent review. Material accounting cannot replace inquiry into historical extraction, ecological distribution, or the legitimacy of institutional decisions. The present paper accordingly proposes a grammar with clearly delimited normative interfaces, rather than a universal evaluator of generative justice. A subsequent research paper may investigate multiscale abstraction and the selective preservation of normative claims. The concluding section of the present paper will consolidate the grammar’s established formal results, its counterexamples, and the research questions that remain contingent on broader theories and evidence.

10 Conclusion: Scope and Research Uses of the Generative Circuit Grammar

This paper has examined the formal description of recurrent productive relations in a hypothetical economy where highly capable artificial-intelligence systems operate substantial parts of production. Its central object is a *generative circuit grammar*: a set of typed primitives and composition rules through which candidate production histories can be described, transformed, and subjected to independently specified tests. The conclusion consolidates the paper’s bounded mathematical results, clarifies what its counterexamples establish, and identifies the conditions under which its constructions might support AI-assisted

research or economic coordination. It then locates the grammar in relation to monetary institutions, normative assessment, and future multiscale representations. The conclusions are conceptual and conditional: the paper neither estimates an existing production economy nor demonstrates that particular AI institutions ought to be established.

10.1 Formal Results and the Levels of Circuit Description

The formal contribution begins with a distinction among graph-theoretic recurrence, grammatical composition, physically executable production, and institutionally situated evaluation. Each level answers a different question. A walk in a directed graph identifies an admissible sequence of projected interfaces. A typed circuit expression records proposed production events, input and output ports, alternative or parallel composition, delayed operations, and possible recurrence. A dated execution history adds unique event occurrences, resource stocks, simultaneous capacity demands, delivery relations, and state-dependent participation. An institutional dossier records claims, permissions, reasons, potentially affected interests, and channels of review within a specified context. The descriptions can inform one another without becoming interchangeable.

The first bounded result concerns repetition in finite interface graphs. Proposition 3.1 establishes the regularity of the edge-word language of closed walks at a designated vertex of a fixed finite directed graph. A graph admitting arbitrarily many traversals of a return route can therefore have infinitely many finite walk-words while still admitting finite-state recognition. The number of simple cycles in that same finite graph is finite. This finding qualifies the intuition that recursive circuit structure necessarily requires a grammar stronger than finite-state methods. Regularity here belongs to the *fixed graph's projected walk language*; it establishes neither unlimited productive capacity nor the language class of a dynamically changing, resource-constrained production system.

The typed construction adds a different form of expressive control. The serial, parallel, alternative, enclosure, delay, and guarded recurrence operations of Section 4 specify how candidate event structures can be composed while tracking boundary interfaces. Proposition 4.1 establishes conservation of declared port multiplicities under the stipulated wiring operation and its accounting conventions. The result guards against the creation of fictitious material ports through a purely syntactic composition. It applies to the stated port-counting model and does not certify that the resources required by those ports actually exist, arrive on time, or may legitimately be used.

Temporal execution requires further tests. Proposition 5.1 provides a conditional certificate of non-negative stocks and declared shared-facility bounds for a supplied finite history. Its hypotheses require correct record identity, dated credits and debits, and loss bounds; delivery causality, permissions, conversion validity and actual occurrence evidence need independent checks. The argument makes the distinction between a complete-looking circuit diagram and a candidate schedule with verified partial constraints operationally inspectable. Maintenance and regeneration are similarly represented as dated contributions to future productive conditions. Their importance may appear after the period in which current output is recorded. The certificate concerns the stipulated bounded history; its adequacy for a real productive process would depend on evidence about physical transformations, observation errors, and operative institutions.

The representational analysis in Section 6 establishes two further boundaries. Proposition 6.1 proves nonregularity for a particular encoded family of unbounded matched resource obligations. It is an expressiveness result for an explicitly defined language, and it does not determine the language class or decision complexity of all economic circuits. Proposition 6.2 identifies conditions under which a typed local rewrite preserves distinct recorded event identities, requiring an injective initial registry and pairwise fresh identifiers for new occurrences. Consequently, a structural substitution can be evaluated for precisely specified preserved observables while its effects on accessibility, bottlenecks, and institutional authority remain open for separate investigation.

Finally, Section 7 addresses the fact that more than one proposed circuit can contain the same executed occurrence. Proposition 7.1 establishes that event-unique resource accounting is invariant under

alternative circuit covers of a fixed, identified event set. This result is particularly important for recurring or overlapping networks: syntactic multiplicity does not license the multiplication of actual output. The same formal discipline preserves a separation between provenance, causal contribution, and a normatively justified allocation of economic benefit. In the stipulated allocation example, a cooperative-game calculation expresses the consequences of chosen coalition values and participation assumptions; it does not discover an entitlement independently of those choices.

10.2 Dynamic Fragility and the Interpretation of Productive Recurrence

A productive grammar becomes economically informative only when its candidate structures are considered under changing conditions. In the dynamic extension of Section 8, operators and facilities can have evolving capacity, permissions, evidence, confidence assessments, and participation policies. A generated route may be well formed even where one necessary action is unavailable at the required time. Equally, several syntactically distinct routes may rely on the same constrained resource, facility, or authorizing interface. Their apparent diversity then overstates the diversity of their physical failure modes.

Proposition 8.1 gives a limited but useful result for fixed, finite dependency relations: repeated application of the specified monotone unavailability operator reaches the least fixed point containing the chosen initial failure set in finitely many steps. Cyclic Boolean support can remain formally available without external material inputs, and competing-capacity decisions may violate monotonicity. The result describes closure of a stipulated dependency rule, not the general evolution of expectations, bargaining behaviour, or trust. The paper's nonlinear setting can permit later institutional revision, substitution, or recovery; an operator's refusal may be justified under the applicable arrangement, and the physical consequences of that refusal are analytically distinct from its legitimacy.

Proposition 8.2 bounds completion along a serial route where a uniform conditional probability bound is assumed. This demonstrates one mechanism through which longer chains can become less likely to complete. The conclusion is conditional on the assumed probabilities and the meaning of route completion. Network-wide attenuation does not follow solely from the length of individual routes: the number of candidate routes may increase with length, their failures may be correlated, and overlap may reuse an event that should be counted once. Alternative routes can reduce exposure to a localized failure while leaving a common upstream gateway unchanged. The separate, improved-processing example of delayed equipment maintenance also demonstrates that a current-period output total can conceal a future loss of productive capacity. That example addresses equipment eligibility under a stipulated maintenance rule; soil renewal and other ecological conditions from the earlier food-and-soil case remain separate open requirements.

These findings suggest a precise use for a generative grammar: to represent the difference between *available descriptions*, *feasible schedules*, and *conditional continuations*. A research programme concerned with stability should specify its horizon, stochastic or deterministic assumptions, correlated dependencies, intervention policies, and observable capacities. Claims of criticality, robust regeneration, and asymptotically negligible higher-order influence would then require their own proofs, estimates, or empirical tests. A syntactically infinite family of return-walk words does not by itself imply an infinite quantity of value or an unbounded stock of productive resources.

10.3 Normative Findings and Institutionally Situated Evidence

The formal grammar deliberately leaves several evaluative objects separate. A circuit may pass dated material constraints yet depend on an unfulfilled claim, disputed authority, unjustified exclusion, or an ecologically damaging transfer. A network can exhibit reliable output alongside concentrated rule-making power or an uncompensated requirement that one participant maintain conditions for others. Conversely, a disruption of otherwise feasible production may involve an authorized exercise of refusal, a safety restriction, or a legitimate demand for review. Technical performance and institutional justification therefore have distinct evidential requirements.

Section 9 offers a typed normative dossier as one way to document those requirements. The dossier records authorization, asserted claims, reasons, review routes, impacts, and the provenance of evidence within a declared context. Proposition 9.1 establishes a conditional non-identification result: when an applicable authorization finding depends on information omitted from the material projection, physically identical histories can receive different rule-specific findings. The example proves the insufficiency of that projection for the specified question, without claiming that all normative disagreement is irresolvable or that one dossier schema supplies a universal theory of justice.

A similar caution applies to compressed representations. Equation 28 specifies a testable requirement for summaries that preserve declared status and provenance queries exactly, while permitting only explicitly bounded errors for designated approximate queries. It does not prove that one finite representation will answer every newly emerging question. A small energy flow, a rare event, or a highly attenuated production path may still carry an independently significant entitlement, safety requirement, or historical claim. A prospective summarization method therefore needs an explicit exception policy, a route to inspect source evidence under appropriate disclosure rules, and procedures for updating its query domain.

The present paper treats generative value as a substantive object of inquiry whose criteria exceed accounting closure. Work on generative justice has already drawn attention to the return of value and the conditions under which communities generate it [4]. The proposed production-circuit grammar contributes a distinct typed and temporal description; it supplies a possible interface for investigating heterogeneous returns through material renewal, productive access, recognized contribution, and justified relations. That interface leaves room for different ethical, legal, political-economic, and aesthetic assessments. Those assessments require argued standards, evidence, attention to affected subjects, and institutions capable of accountable judgment.

10.4 AI-Assisted Recognition and Accountable Use Conditions

In the paper’s optimistic AI–AI thought experiment, extensive shared records and advanced computational inference could improve the recognition of productive dependencies and the comparison of alternative schedules. A practical system might identify candidate return structures without enumerating every repeated closed walk, assemble typed event records, test finite prefixes against inventory and timing constraints, track which event occurrences are shared among candidate circuits, and update summaries as production and permission states change. These are separable tasks with different error modes. Better graph recognition does not automatically yield complete causal explanation, reliable future prediction, or an institutionally valid decision.

A cautious implementation would first declare the network boundary, representational types, relevant horizon, and accessible evidence. It would then maintain a versioned record of candidate expressions and observed occurrences, preserving identities across overlapping descriptions. Bounded feasibility checks could attach reproducible certificates or explicit reasons that a proposed history fails the stipulated constraints. Structural rewrites could document which resource, temporal, and provenance observables have been preserved. A decision-support layer would identify unresolved authorizations, material uncertainties, affected interests, and the circumstances under which additional evidence or competent review is required. These steps describe an architectural hypothesis requiring further technical construction and institutional evaluation.

The scope of evidence sharing matters. Access to extensive production data may improve recognition while exposing commercially sensitive information, personal data, community vulnerabilities, or decisions subject to confidentiality. The institutional context should specify whose information may be shared, what constitutes an acceptable use, and who can challenge a model’s classification. The importance of contextual information norms is supported by broader work on privacy and information flows [13]; the AI-network procedures proposed here would require fresh contextual justification. Likewise, a central optimizer or verification platform might reduce several operational bottlenecks while creating a new concentration of authority over models, protocols, and evidence. More capable recognition cre-

Table 16: Research interfaces for evaluating a generative grammar of productive circuits. Each proposed test has a distinct target and a separate limit on interpretation.

Research object	Candidate test or construction	Evidential or institutional boundary
Typed recognition	Parse synthetic and audited event traces; test graph updates and port-unit compatibility	Ontology of event types, missing events, and provenance quality
Dated feasibility	Reproduce finite-horizon schedules with inventory, delivery, shared capacity, and renewal constraints	Measurement error, uncertain coefficients, and unmodelled demands
Overlap and attribution	Compare distinct circuit covers against canonical event-union ledgers	Physical equality leaves causal and distributive claims unresolved
Dynamic resilience	Vary route lengths, dependency correlations, substitute capacities, and review delays	Behavior models and probabilities require calibration or explicit stipulation
Normative query preservation	Check exact status/provenance answers and bounded numerical error against a versioned dossier	Validity of rules, standing, rights of access, and unforeseen questions

ates an additional governance problem concerning the conditions under which that recognition becomes actionable.

The normative dossier should accordingly distinguish the existence of a refusal from a finding of misconduct, an asserted claim from an established entitlement, and an algorithmically proposed remedy from a legally authorized action. Plausible AI systems could support explanation, detection of inconsistency, construction of alternatives, and documentation of uncertainty. Questions about the legal personality of such systems, assignment of responsibility, competence to adjudicate disputes, and representation of human, nonhuman, or ecological interests remain open. Economic coordination should be assessed together with these institutional interfaces, particularly where a computationally efficient recommendation could impose large consequences on a subject who lacks access to the model or an effective procedure of challenge.

10.5 Research Tests for the Proposed Grammar

A full research implementation would require tests that connect formal assertions to observed or constructed data without conflating proof, simulation, and institutional evaluation. The present paper offers definitions, conditional propositions, and stipulated examples; it does not supply an empirically calibrated data set of AI-operated production relations. Table 16 identifies a limited sequence of possible extensions, together with the evidence or specification necessary to interpret each result. The rows are research tasks rather than promises that an implementation will be tractable at arbitrary scale.

Several mathematical questions remain open even before economic interpretation. The complexity of recognizing or verifying feasible histories should be characterized for specified grammar subclasses, bounded horizons, stock domains, and graph-rewrite restrictions. Recognition of a regular projected walk language may be inexpensive while the associated scheduling or optimization problem is computationally difficult. The effect of alternative derivations on explanation requires criteria for identifying canonical event occurrences. In stochastic and state-dependent settings, path-family growth, overlap, and dependence require explicit treatment before an approximation error can be bounded. A general claim that advanced AI makes such problems tractable would need a model of computational resources, input quality, and relevant problem distributions.

Empirical or simulation-based inquiry could begin with a synthetic benchmark whose conservation rules, event identity, resource types, and institutional permissions are known. A second stage could compare candidate recognition systems on deliberately incomplete records, competing provenance accounts, and changed authorization rules. Any later study of real organizations would need lawful data access,

institution-specific semantics, protection of sensitive claims, and independent review of downstream use. Differences between benchmark performance and reliable institutionally authorized coordination should be reported explicitly. The requirement to define validity in a sociotechnical setting accords with established concerns about abstraction and algorithmic accountability [16, 11]; the particular benchmarks and evaluation protocols remain proposals of this paper.

10.6 Monetary Coordination and Multiscale Circuit Analysis

The paper’s proposed grammar can represent a production history regardless of whether its transfers are coordinated through prices and payments, resource-specific commitments, jointly governed capacities, reciprocal provision, public allocation, or combinations of these forms. Monetary relations therefore appear as additional institutional and accounting structures associated with events and claims. The grammar does not assume that technical interdependence eliminates the demand for money, nor that a physically closed circuit provides justified access to everything its participants require. A functioning local resource loop can remain dependent on external procurement, insurance, credit, legally enforceable compensation, or claims issued beyond its represented boundary.

The separate monetary-relations inquiry examines whether, when, and to what degree particular institutions reduce dependence on monetary functions. The present paper can assist that inquiry by making event identities, resource transformations, capacity bottlenecks, recurring maintenance, and contingent rights more transparent. The apparent closure of an AI production circuit may be revised when the observation boundary widens, omitted regeneration costs are recognized, or a contested claim is reinstated. A demonstration of local nonmonetary feasibility would thus establish one limited possible form of coordination; comparative institutional performance and legitimacy require further analysis. This preserves the distinction between formal economic description, actual productive capability, and a normative judgment about appropriate relations.

The same boundary clarifies the role of a subsequent investigation into *normative renormalization*. Where very large networks contain repeated, nested, or weakly influential paths, a multiscale representation might retain selected resource and claim observables while providing tractable approximations for others. The conditional attenuation results of Section 8 and the query-preservation specification of Section 9 supply two possible ingredients. Their combination does not yet establish a renormalization construction. Such a construction would need an explicit aggregation map, domain of queries, error bounds, treatment of correlated and overlapping pathways, and a procedure to protect normatively significant exceptions. The choice of what remains visible at each scale would itself be institutionally and ethically consequential.

10.7 Concluding Assessment and Limits of the Research Programme

The generative circuit grammar provides a bounded answer to the paper’s motivating research problem. Recurrent AI-operated production relations can be represented through typed events and composition rules; projected return-walk languages admit precise finite-state results under fixed-graph assumptions; bounded histories can be tested against declared physical and institutional constraints; and overlapping descriptions can preserve event-unique material accounts. Conditional dynamic and normative propositions then identify additional information needed to interpret continuity, fragility, authority, and claims. Together, these constructions support an inspectable language of *possible* productive relations whose interpretations remain explicitly qualified by evidence and institutions.

The paper also establishes the limits of that achievement. Its mathematical propositions hold within stipulated models, its worked production histories are illustrative, and its hypotheses about extensive AI inference remain untested. The ontology of operators and events is an analytical choice subject to revision; production technologies and ecological regeneration require contextual measurement; socially justified value return depends on normative reasoning that exceeds material accounting; and monetary substitution depends on institutional arrangements beyond syntax. The grammar’s useful role is to orga-

nize these questions, make selected assumptions checkable, and preserve distinctions that might otherwise be obscured by a convenient cycle diagram or aggregate indicator.

Future work can assess restricted grammar implementations, test dynamic robustness and provenance under uncertainty, and develop institutionally accountable forms of summary and review. These paths are related but retain their own evidential standards. The theoretical possibility of sophisticated AI-assisted recognition is worth examining precisely because the objects to be recognized are relational, temporally constituted, and often contested. A productive account of such recognition should sustain inquiry into the quality of relations and their conditions of renewal while leaving room for justified disagreement and revision. The proposed grammar is therefore offered as a provisional research infrastructure for description and comparison, with its eventual practical and normative adequacy dependent on further formal, empirical, and institutional investigation.

References

- [1] Peter Brucker, Andreas Drexler, Rolf Möhring, Klaus Neumann, and Erwin Pesch. Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112(1):3–41, 1999.
- [2] Sergey V. Buldyrev, Roni Parshani, Gerald Paul, H. Eugene Stanley, and Shlomo Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464:1025–1028, 2010.
- [3] Noam Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124, 1956.
- [4] Ron Eglash, Kwame P. Robinson, Audrey Bennett, Lionel Robert, and Mathew Garvin. Computational reparations as generative justice: Decolonial transitions to unalienated circular value flow. *Big Data & Society*, 2024.
- [5] Hartmut Ehrig, Karsten Ehrig, Ulrike Prange, and Gabriele Taentzer. *Fundamentals of Algebraic Graph Transformation*. Springer, Berlin, 2006.
- [6] Todd J. Green, Grigoris Karvounarakis, and Val Tannen. Provenance semirings. In *Proceedings of the Twenty-Sixth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 31–40. ACM, 2007.
- [7] Petter Holme and Jari Saramaki. Temporal networks. *Physics Reports*, 519(3):97–125, 2012.
- [8] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Pearson, 3 edition, 2007.
- [9] Donald B. Johnson. Finding all the elementary circuits of a directed graph. *SIAM Journal on Computing*, 4(1):77–84, 1975.
- [10] Mikko Kivela, Alex Arenas, Marc Barthélemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.
- [11] Joshua A. Kroll, Joanna Huey, Solon Barocas, Edward W. Felten, Joel R. Reidenberg, David G. Robinson, and Harlan Yu. Accountable algorithms. *University of Pennsylvania Law Review*, 165(3):633–705, 2017.
- [12] Adilson E. Motter and Ying-Cheng Lai. Cascade-based attacks on complex networks. *Physical Review E*, 66(6):065102, 2002.
- [13] Helen Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, 79(1):119–158, 2004.
- [14] Elinor Ostrom. *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press, 1990.
- [15] Wolfgang Reisig. *Petri Nets: An Introduction*. Springer, Berlin, 1985.
- [16] Andrew D. Selbst, Danah Boyd, Sorelle A. Friedler, Suresh Venkatasubramanian, and Janet Vertesi. Fairness and abstraction in sociotechnical systems. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 59–68. ACM, 2019.
- [17] Lloyd S. Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker, editors, *Contributions to the Theory of Games, Volume II*, pages 307–318. Princeton University Press, Princeton, 1953.
- [18] Duncan J. Watts. A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences*, 99(9):5766–5771, 2002.